PROGRAM DOCUMENT

```
.REM &

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

PRODUCT CODE:   AC-T796B-MC
PRODUCT NAME:   C2DHVB0 DHU-11 FUNC TST PART2
PRODUCT DATE:   3 MARCH 1984
MAINTAINER:     ENE - DIAGNOSTICS GROUP
AUTHOR:         ANTHONY HART
MODIFIED BY:    ANTHONY HART
```

♦♦♦♦♦ MODIFICATION HISTORY ♦♦♦♦♦

ORIGINAL RELEASE:	15 DEC-83	ANTHONY HART
VERSION B0	3-MAR-84	ANTHONY HART

THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE OLD CZDHVA:

THE HARDWARE QUESTION "TYPE OF LOOPBACK" HAS BEEN ALTERED TO INCLUDE THE STAGGERED LOOPBACK CONNECTORS ON THE DHU11 DISTRIBUTION PANEL (H3029).

THE HARDWARE QUESTION "BR LEVEL" HAS BEEN REMOVED.

THE MODEM SIGNAL TESTS THAT WERE IN THE OLD VERSION (CZDHVA0) HAVE NOW BEEN REMOVED TO PART CZDHW. THEY WERE TESTS: 16 THRU 23, IN CZDHVA0.

THE FOLLOWING NEW TESTS HAVE BEEN ADDED TO THIS PART:
TEST 13 - RXTIMER REGISTER TEST
TEST 14 - TX_ACTION FIFO TEST
TEST 15 - TX_FIFO TEST

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CZDHVBO IS PART OF THE DHU-11 FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST VERIFIES THAT THE MAJOR COMMUNICATIONS FUNCTIONS OF THE BOARD ARE FUNCTIONING CORRECTLY. THIS PROGRAM DOES NOT PERFORM EXTENSIVE DATA TRANSMISSION AND RECEPTION TESTS.

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 DHU FVT:

- O UNIBUS PROCESSOR WITH AT LEAST 32K BYTES OF MEMORY.
- O DHU BOARDS INSTALLED ON THE UNIBUS.
- O APPROPRIATE PROGRAM LOAD DEVICE SUPPORTING XXDP+ MEDIA OR A DOWN LINE LOADING SYSTEM.

1.3 RELATED DOCUMENTS AND STANDARDS

- O XXDP+ USER'S MANUAL DESCRIBES THE RUNNING OF DIAGNOSTICS UNDER THE XXDP+ MONITOR.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE PROCESSOR, THE UNIBUS, 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 PERFOMANCE 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)
/FLAGS	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.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS.SEE THE FLAGS SECTION OF THIS DOCUMENT.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

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

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

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

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
?FLAGS					
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-0>, 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 (NOT RELATED TO BELL PROMPTING).
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION (ILLEGAL FOR THIS DIAGNOSTIC).
ISR	INHIBIT STATISTICAL REPORTS.

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.
(HAS NO EFFECT IN THIS 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:'10E=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 UNITS SWITCH SHOULD NOT BE USED WITH THIS PROGRAM. 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 TES 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 -
THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

2.4.10 DISPLAY COMMAND -
****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

2.4.10.1 EFFECT OF DISPLAY COMMAND -

THE HARDWARE P TABLE FOR THE TEST STATION IS PRINTED IN THE FORMAT IN WHICH IT WAS 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 SURPRESSED 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 REQUESTES THE CSR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS ADDRESS 160460 (OCTAL).

2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER IS 310 (OCTAL).

3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHU11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 5 FOR LINE 5) THAT LINE WILL BE TESTED BY THE FVT.

4. TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277) - THIS QUESTION REQUESTS THE TYPE OF LOOPBACK TO BE USED WHEN TESTING THE DHU-11. THE FOLLOWING TYPES ARE SUPPORTED:

   - INTERNAL - ONLY INTERNAL UART LOOPBACK IS TO BE USED IN TESTING THE DHU-11.
   - H3029 OR H3277 - STAGGERED LOOPBACK CONNECTORS ARE PROVIDED ON THE DHU11 DISTRIBUTION PANEL (H3029) IF THIS DISTRIBUTION PANEL IS NOT PRESENT THEN H3277 STAGGERED BERG CONNECTOR(S) MUST BE INSTALLED ON THE BERG CONNECTOR SOCKETS OF THE DHU11.
   - H325 - SINGLE LINE, 25 PIN LOOPBACK CONNECTORS (TYPE H325) ARE INSTALLED ON THE LINES TO BE TESTED.

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 SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

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 THAT UNIT.

2. EXTENDED ERROR REPORTING - THIS QUESTION ASKS WHETHER EXTENDED ERROR INFORMATION IS REQUIRED OTHER THAN THE "TEST FAILED" MESSAGE, ON EACH ERROR REPORTED. THE DEFAULT IS "NO" I.E. ONLY A MESSAGE REPORTING THE FACT THAT THE TEST FAILED WILL BE PRINTED.

3. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS QUESTION IS ASKED ONLY IF THE PREVIOUS QUESTION WAS ANSWERED "YES". THE 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.

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 (D) ?  8<CR>

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

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

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

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

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

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

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

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

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

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

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

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

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

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

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

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

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

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

        UNIT 1
```

```
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE 0 (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/UNIBUS AND 50HZ (IF THERE IS A CLOCK) QUESTIONS. NOTE, NOT ALL VERSIONS OF XXDP+ ASK FOR THE CLOCK FREQUENCY
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 SPECIFIC ERROR MESSAGES

THIS PROGRAM IS INTENDED TO PROVIDE A GO/NOGO INDICATION OF THE FUNCTIONALITY OF THE DHU-11 BOARDS. TO EXECUTE THE PROGRAM IN THIS MODE THE OPERATOR NEED ONLY ANSWER THE "EXTENDED ERROR REPORTING" SOFTWARE QUESTION WITH "NO". THE PROGRAM WILL THEN ONLY PRINT THE NAME OF THE FAILING TEST THE TEST AND ERROR NUMBERS. FOR A LIST OF THE TEST NAMES IN THIS PROGRAM SEE THE TEST SUMMARIES SECTION OF THIS DOCUMENT. AN EXAMPLE OF SUCH A AN ERROR MESSAGE IS THE FOLLOWING:

```
CZDHV DVC FTL ERR  04106 ON UNIT 00 TST 003 SUB 000 PC: XXXXXX
DMA_ABORT BIT TEST FAILED
```

THIS ERROR INDICATES THAT A FATAL ERROR WAS ENCOUNTERED DURING THE TEST WHICH TESTS THE DMA_ABORT BIT.

IF THE OPERATOR HAD REQUESTED EXTENDED ERROR REPORTING THE SAME ERROR WOULD BE REPORTED AS FOLLOWS:

```
CZDHV DVC FTL ERR  04106 ON UNIT 00 TST 003 SUB 000 PC: XXXXXX
DMA_ABORT BIT TEST FAILED
  DMA_START BIT FOUND SET AFTER DMA ABORTED ON LINE:    8
```

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 FUTHER INFORMATION SEE THE SWITCHES SECTION OF THIS DOCUMENT.

5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDHVB:

1. DEVICE REGISTER ACCESS TEST - VERIFIES THAT THE DUT REGISTERS WILL RESPOND WITH THE CORRECT UNIBUS HANDSHAKING SIGNALS. VERIFIES THAT THE DUT IS AT THE CORRECT ADDRESS.

2. DMA.START TEST - VERIFIES THAT EACH DMA START BIT WILL INITIATE A DMA TRANSMISSION ON A LINE

3. DMA.ABORT TEST - VERIFIES THAT EACH DMA ABORT BIT WILL STOP A DMA TRANSMISSION, RETURN A TX.ACTION AND SUCCESFULY RESTART THE DMA.

4. DMA.ERROR TEST - VERIFIES THAT THE DMA ERROR BIT IN THE CSR REPORTS DMA ERRORS CORRECTLY WHEN THEY OCCUR.

5. O.AUTO INACTIVE TEST - VERIFIES THAT THE DUT WILL NOT RESPOND TO INCOMING XON AND XOFF CHARACTERS WHEN O.AUTO IS CLEAR.

6. O.AUTO ACTIVE TEST - VERIFIES THAT THE DUT RESPONDS CORRECTLY TO INCOMING FLOW CONTROL CHARACTERS WHEN ACTIVE

7. I.AUTO INACTIVE TEST - VERIFIES THAT THE DUT WILL NOT GENERATE XON AND XOFF CHARACTERS IN RESPONSE TO THE APPROPRIATE FIFO CONDITIONS WHEN I.AUTO IS INACTIVE.

8. I.AUTO ACTIVE TEST - VERIFIES THAT THE DUT WILL GENERATE XON AND XOFF CHARACTERS IN RESPONSE TO THE APPROPRIATE FIFO CONDITIONS WHEN I.AUTO IS ACTIVE.

9. FIFO DATA TEST - VERIFIES THAT THE FIFO WILL HOLD 256 CHARACTERS WITHOUT CORRUPTING DATA.

10. FIFO 3/4 LEVEL INACTIVE TEST - VERIFIES THAT THE 3/4 LEVEL ALARM DOES NOT BECOME ACTIVE BELOW THE 3/4 LEVEL.

11. FIFO 3/4 LEVEL ACTIVE TEST - VERIFIES THAT THE 3/4 LEVEL ALARM BECOMES ACTIVE WHEN THE FIFO IS 3/4 FULL.

12. FIFO 3/4 LEVEL ACTIVE/INACTIVE TEST - VERIFIES THAT THE 3/4 LEVEL ALARM, ONCE ACTIVATED, REMAINS ACTIVE UNTIL THE FIFO IS REDUCED BELOW THE 1/2 LEVEL.

13. FIFO 1/2 LEVEL TEST - VERIFIES THAT THE FIFO 1/2 LEVEL ALARM SYSTEM BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.

14. RXTIMER TEST - VERIFIES THAT THE HOLD OFF TIMER FOR RX INTERRUPTS IS OPERATING CORRECTLY, AND THAT THE 3/4 FULL LEVEL OVERIDES THE TIMER.

15. TX-ACTION FIFO TEST - VERIFIES THAT THE TX-ACTION FIFO CAN HOLD 16 UNIQUE TX-ACTIONS, AND THAT ONLY ONE TX INTERRUPT OCCURS FOR ALL 16 TX-ACTIONS.

16. TX FIFO TEST - VERIFIES THAT THE FIFO WILL 64 UNIQUE CHARACTERS AND ALSO THAT ONLY ONE INTERRUPT OCCURS FOR ALL 64 CHARACTERS.

17. BREAK GENERATION TEST - VERIFIES THAT ALL SERIAL TRANSMIT LINES CAN GENERATE A BREAK BY SETTING THE BRK BIT IN THE ASSOCIATED LNCTRL REGISTER.

18. NO OVERRUN ERROR TEST - VERIFIES THAT THE DUT WILL NOT REPORT DATA OVERRUN ERRORS WHEN THEY DO NOT OCCUR.

19. OVERRUN ERROR TEST - VERIFIES THAT THE DUT WILL REPORT DATA OVERRUN ERRORS WHEN THEY OCCUR.

20. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 CHARACTERS WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS INTERRUPTION OF THE OTHER TESTS BY THESE CODES IF THEY ARE NOT CRITICAL TO THE PERFORMANCE OF THE TESTS.

6.0 EXAMPLE ERROR FREE PASS

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

```
.R CZDHVBO
CZDHVBO.BIN
DRS
CZDHV-B-0
DHU-11 FUNC TST PART2
UNIT IS DHU-11
RESTRT ADDR:  147670
DR>STA/PAS:1

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
CSR ADDRESS:  (O)  160460  ? 12

UNIT 1
CSR ADDRESS:  (O)  160460  ? 160500
INTERRUPT VECTOR ADDRESS:  (O)  310  ? 320
ACTIVE LINE BIT MAP:  (O)  177777 ? <CR>
TYPE OF LOOPBACK (1=INTERNAL, 2=M3029 OR M3277):  (O)  2  ? 1

CHANGE SW  (L)  ? Y

REPORT UNIT NUMBER AS EACH UNIT IS TESTED:  (L) Y  ? <CR>
EXTENDED ERROR REPORTING:  (L) N  ? <CR>

 TESTING UNIT : 0

 TESTING UNIT : 1

CZDHV EOP       1
    0 TOTAL ERRS

DR>
```

```
                                .LIST SEQ,LOC,BIN,MEB
                                .NLIST CND


                        .SBTTL  PROGRAM HEADER


                                .MCALL  SVC
000000                          SVC                             ; INITIALIZE SUPERVISOR MACROS

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

        000001          SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
        000001          SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
        000001          SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
        000001          SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
        000001          SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT

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

000000                  .ENABL  ABS
                        ;.ENABL AMA
        002000          .       =       2000

002000                  BGNMOD

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

002000                  POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL


002000                  HEADER  CZDHV,B,0,22,0,PRIO7
002000                                                  L$NAME::
002000      103                                                 .ASCII /C/
002001      132                                                 .ASCII /Z/
002002      104                                                 .ASCII /D/
002003      110                                                 .ASCII /H/
002004      126                                                 .ASCII /V/
002005      000                                                 .BYTE   0
002006      000                                                 .BYTE   0
002007      000                                                 .BYTE   0
002010                                                  L$REV::
002010      102                                                 .ASCII  /B/
002011                                                  L$DEPO::
002011      060                                                 .ASCII  /0/
002012                                                  L$UNIT::
002013  000000                                                  .WORD   0
002014                                                  L$TIML::
```

```
002014  000022                          .WORD   22
002016                          L$HPCP::
002016  035156                          .WORD   L$HARD
002020                          L$SPCP::
002020  035404                          .WORD   L$SOFT
002022                          L$HPTP::
002022  002176                          .WORD   L$HW
002024                          L$SPTP::
002024  002210                          .WORD   L$SW
002026                          L$LADP::
002026  035706                          .WORD   L$LAST
002030                          L$STA::
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::
002036  000000                          .WORD   0
002036                          L$APT::
002036  000000                          .WORD   0
002040                          L$DTP::
002040  002124                          .WORD   L$DISPATCH
002042                          L$PRIO::
002042  000340                          .WORD   PRI07
002044                          L$ENVI::
002044  000000                          .WORD   0
002046                          L$EXP1::
002046  000000                          .WORD   0
002050                          L$MREV::
002050     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::
002060  004120                          .WORD   L$DVTYP
002062                          L$REPP::
002062  020030                          .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::
002070  000000                          .WORD   0
002072                          L$DUT::
002072  020704                          .WORD   L$DU
002074                          L$LUN::
002074  000000                          .WORD   0
002076                          L$DESP::
002076  004130                          .WORD   L$DESC
002100                          L$LOAD::
002100  104035                          EMT     E$LOAD
002102                          L$ETP::
002102  004050                          .WORD   L$ERRTBL
002104                          L$ICP::
002104  020044                          .WORD   L$INIT
```

N2

PROGRAM HEADER

```
        002106
        002106  020666
        002110
        002110  020664
        002112
        002112  020036
        002114
        002114  000000
        002116
        002116  000000
        002120
        002120  000000
   1114

L$CCP::
                .WORD   L$CLEAN
L$ACP::
                .WORD   L$AUTO
L$PRT::
                .WORD   L$PROT
L$TEST::
                .WORD   0
L$DLY::
                .WORD   0
L$HIME::
                .WORD   0
```

```
1126                                    .SBTTL  DISPATCH TABLE
1127
1128                                    ;**
1129                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1130                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1131                                    ;--
1132
1133 002122                                    DISPATCH 20
     002122  000024                                                    .WORD   20
     002124                                                  L$DISPATCH::
     002124  021022                                                    .WORD   T1
     002126  021304                                                    .WORD   T2
     002130  021710                                                    .WORD   T3
     002132  022346                                                    .WORD   T4
     002134  022674                                                    .WORD   T5
     002136  023474                                                    .WORD   T6
     002140  024274                                                    .WORD   T7
     002142  024716                                                    .WORD   T8
     002144  025410                                                    .WORD   T9
     002146  025716                                                    .WORD   T10
     002150  026226                                                    .WORD   T11
     002152  026726                                                    .WORD   T12
     002154  027424                                                    .WORD   T13
     002156  030102                                                    .WORD   T14
     002160  031152                                                    .WORD   T15
     002162  031740                                                    .WORD   T16
     002164  032766                                                    .WORD   T17
     002166  033412                                                    .WORD   T18
     002170  034130                                                    .WORD   T19
     002172  035074                                                    .WORD   T20
1134
```

```
1149
1150
1151                                    .SBTTL  DEFAULT HARDWARE P-TABLE
1152
1153                                    ;**
1154                                    ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1155                                    ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1156                                    ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
1157                                    ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
1158                                    ;**
1159
1160 002174                                    BGNHW   DFPTBL
      002174  000004                                                                    .WORD   L10000-L$HW/2
      002176                                                                    L$HW::
      002176                                                                    DFPTBL::
1161
1162 002176  160460                            .WORD   160460  ;DEFAULT CSR ADDRESS
1163 002200  000310                            .WORD   310     ;DEFAULT VECTOR ADDRESS
1164 002202  177777                            .WORD   177777  ;DEFAULT ACTIVE LINES BIT MAP
1165 002204     002                            .BYTE   2       ;DEFAULT LOOPBACK MODE
1166                                           .EVEN
1167 002206                                    ENDHW
      002206                                                                    L10000:
```

```
1176
1177
1178                            .SBTTL  SOFTWARE P-TABLE
1179
1180                            ;**
1181                            ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1182                            ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
1183                            ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1184                            ; AT RUN TIME.
1185                            ;--
1186
1187 002206                             BGNSW   SFPTBL
     002206  000002                                                        .WORD   L10001-L$SW/2
     002210                                                             L$SW::
     002210                                                             SFPTBL::
1188
1189 002210  000020                     OPTION::        .WORD   20      ;BIT MAP OF PROGRAM CONTROL FLAGS
1190 002212  000000                     NDERPT::        .WORD   0       ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.
1191
1192 002214                             ENDSW
     002214                                                             L10001:
```

```
1201
1202                            .SBTTL  GLOBAL EQUATES SECTION
1203
1213
1214
1215
1216                            ;++
1217                            ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1218                            ; ARE USED IN MORE THAN ONE TEST.
1219                            ;--
1220
1221    000020                          NUMLNS==20      ;NUMBER OF LINES ON DHU11 IS 8.
1222    177777                          MAPLNS==177777  ;BIT MAP OF LINES ON DHU11.
1223
1224                            ;****** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS ******
1225    000000                          CSR0==0         ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
1226    000002                          RBUF0==2        ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
1227    000002                          RXTIM0==2       ;RECIEVE TIMER REGISTER OFFSET FROM THE CSR ADDRESS
1228    000004                          LPR0==4         ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
1229    000006                          FSLS0==6        ;FIFOSIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
1230    000006                          FDAT0==6        ;FIFODATA REGISTER OFFSET FROM THE CSR ADDRESS
1231    000010                          LNCTR0==10      ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
1232    000012                          TXAD10==12      ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
1233    000014                          TXAD20==14      ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
1234    000016                          TXBFC0==16      ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS
1235
1236                            ;****** EQUATES USED WITH RESPECT TO THE RX BUFFER ******
1237    000020                          RXBETX==16.     ;LEVEL OF RX BUFFER AT WHICH TO RE-ENABLE TRANSMISSION.
1238    000030                          RXBDTX==24.     ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
1239    000100                          RXBFUL==64.     ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.
1240
1241
1256 002214                             EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
                100000          BIT15== 100000
                040000          BIT14== 40000
                020000          BIT13== 20000
                010000          BIT12== 10000
                004000          BIT11== 4000
                002000          BIT10== 2000
                001000          BIT09== 1000
                000400          BIT08== 400
                000200          BIT07== 200
                000100          BIT06== 100
                000040          BIT05== 40
                000020          BIT04== 20
                000010          BIT03== 10
                000004          BIT02== 4
                000002          BIT01== 2
                000001          BIT00== 1
                                ;
                001000          BIT9==  BIT09
                000400          BIT8==  BIT08
                000200          BIT7==  BIT07
                000100          BIT6==  BIT06
```

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

```
                                .SBTTL  GLOBAL DATA SECTION

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

                                ;*****************************************************************************
                                ;       UNIT VARIBLE AREA
                                ;*****************************************************************************

002214  000200                          RXVECA:: .WORD  200     ;RX VECTOR ADDRESS.
002216  000204                          TXVECA:: .WORD  204     ;TX VECTOR ADDRESS.
002220  177777                          ACTLNS:: .WORD  177777  ;ACTIVE LINE BIT MAP.
002222     000                          LOPBCK:: .BYTE  0       ;LOOPBACK MODE
002223     004                          BRLEVL:: .BYTE  4       ;INTERRUPT BUS REQUEST LEVEL
002224  000000                          UNITN:: .WORD   0       ;UNIT NUMBER.

                                ;*****************************************************************************
                                ;       DEVICE REGISTER ADDRESS TABLE
                                ;*****************************************************************************

002226                          DRADRT::
002226  160020                  CSRA::   .WORD   160020 ;DHU-11 CSR ADDRESS.
002230  160022          RXTMA:: RBUFA::  .WORD   160022 ;DHU-11 RECIEVE BUFFER/TIMER ADDRESS.
002232  160024                  LPRA::   .WORD   160024 ;DHU-11 LINE PARAMETER REGISTER ADDRESS.
002234  160026          FDATA:: FSLSA::  .WORD   160026 ;DHU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS,
                                                        ;AND FIFO DATA REGISTER ADDRESS.
002236  160030                  LNCTRA:: .WORD   160030 ;DHU-11 LINE CONTROL REGISTER ADDRESS.
002240  160032                  TXAD1A:: .WORD   160032 ;DHU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
002242  160034                  TXAD2A:: .WORD   160034 ;DHU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
002244  160036                  TXBFCA:: .WORD   160036 ;DHU-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS

                                ;*****************************************************************************
                                ;       ASSORTED GLOBAL VARIABLES:
                                ;*****************************************************************************

002246  000000                          BUFPTR:: .WORD  0       ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
002250  000000                          CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
002252  000000                          EXOERR:: .WORD  0       ; "EXIT ON ERROR" FLAG.
002254  000001                          TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
002256  000000                          IESTAT:: .WORD  0       ;STORAGE FOR STATES OF THE DUT INT ENABLE BITS.
002260  031463                          LGRP1M:: .WORD  31463   ;BIT MAP OF LINES IN LINE GROUP I.
002262  146314                          LGRP2M:: .WORD  146314  ;BIT MAP OF LINES IN LINE GROUP II.
002264  000000                          PASCNT:: .WORD  0       ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
002266  000000                          RXINTC:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
002270  000000                          RXINTF:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
002272  000000                          TXINTC:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
002274  000000                          TXINTF:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
002276  000000                          TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
002300  000000                          TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
002302  000000                          WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.

                                ;*****************************************************************************
                                ;       LINE TIME CLOCK VARIABLES AND STORAGE.
                                ;*****************************************************************************

002304  177546                          CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
002306  000300                          CLKBRL:: .WORD  PRI06   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
```

```
1323 002310  000100                    CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1324 002312  000074                    CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1325 002314  000000                    TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1326 002316  000000                    TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1327 002320  000170                    TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1328 002322  000170                    BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1329 002324  000021                    MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1330 002326  000062                    MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1331
1332                            ;*****************************************************************************
1333                            ;       MEMMORY MANAGEMENT VARIABLES AND FLAGS.
1334                            ;*****************************************************************************
1335 002330  177572                    MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1336 002332  000000                    MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1337 002334  000000                    MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1338
1339                            ;*****************************************************************************
1340                            ;       TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1341                            ;*****************************************************************************
1342 002336  000001            BITTBL:: .WORD  1               ;BIT 0 SET.
1343 002340  000002                     .WORD  2               ;BIT 1 SET.
1344 002342  000004                     .WORD  4               ;BIT 2 SET.
1345 002344  000010                     .WORD  10              ;BIT 3 SET.
1346 002346  000020                     .WORD  20              ;BIT 4 SET.
1347 002350  000040                     .WORD  40              ;BIT 5 SET.
1348 002352  000100                     .WORD  100             ;BIT 6 SET.
1349 002354  000200                     .WORD  200             ;BIT 7 SET.
1350 002356  000400                     .WORD  400             ;BIT 8 SET.
1351 002360  001000                     .WORD  1000            ;BIT 9 SET.
1352 002362  002000                     .WORD  2000            ;BIT 10 SET.
1353 002364  004000                     .WORD  4000            ;BIT 11 SET.
1354 002366  010000                     .WORD  10000           ;BIT 12 SET.
1355 002370  020000                     .WORD  20000           ;BIT 13 SET.
1356 002372  040000                     .WORD  40000           ;BIT 14 SET.
1357 002374  100000                     .WORD  100000          ;BIT 15 SET.
1358
1359                            ;*****************************************************************************
1360                            ;*      GPR SAVE AREAS ZERO AND ONE.
1361                            ;*****************************************************************************
1362 002376                    GPRS0B::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1363 002376  000000                     .WORD  0               ;WORD 1, STORAGE FOR R1.
1364 002400  000000                     .WORD  0               ;WORD 2, STORAGE FOR R2.
1365 002402  000000                     .WORD  0               ;WORD 3, STORAGE FOR R3.
1366 002404  000000                     .WORD  0               ;WORD 4, STORAGE FOR R4.
1367 002406  000000                     .WORD  0               ;WORD 5, STORAGE FOR R5.
1368                            ;*****************************************************************************
1369                            ;       STORAGE AREA FOR THE BMP CODE QUEUE.
1370                            ;*****************************************************************************
1371 002410  000000            BMPCQP::        .WORD   0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1372 002412                    BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1373 002612                    BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1374
1375                            ;*****************************************************************************
1376                            ;       STORAGE AREA FOR ERROR SUMMARY TABLE AND FLAGS.
1377                            ;*****************************************************************************
1378 002612  000000            ERSMRF::        .WORD   0       ;ERROR SUMMARY FLAGS.
1379 002614                    ERCNTB::        .BLKW   16      ;TABLE OF ERROR COUNTS.
```

```
                        ;*****************************************************************************
                        ;       GENERAL TABLE AND BUFFER AREA--513 WORDS.
                        ;*****************************************************************************
002650                  BUFBAS::                        ;BASE OF MEMORY BUFFER.
002650                  ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
003250                  BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
003450                  BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
003650                  BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
003650                  ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.

                        ;*****************************************************************************
                        ;               RECEPTION TABLE OF COUNTERS
                        ;*****************************************************************************
003710                  RXCNTB:: .BLKW  16.             ;RECEPTION CHARACTER COUNTERS TABLE.

                        ;*****************************************************************************
                        ;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
                        ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
                        ;*    WHEN ACCESSING A TABLE OF WORDS.
                        ;*    NOTE: DO NOT WRITE A NON-ZERO VALUE INTO THE UPPER BYTE OF ANY ENTRY.
                        ;*****************************************************************************
003750                  TXRXLB::                        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
003750  000000                  .WORD   0               ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
003752  000002                  .WORD   2.              ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
003754  000004                  .WORD   4.              ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
003756  000006                  .WORD   6.              ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
003760  000010                  .WORD   8.              ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
003762  000012                  .WORD   10.             ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
003764  000014                  .WORD   12.             ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
003766  000016                  .WORD   14.             ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
003770  000020                  .WORD   16.             ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
003772  000022                  .WORD   18.             ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
003774  000024                  .WORD   20.             ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
003776  000026                  .WORD   22.             ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
004000  000030                  .WORD   24.             ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
004002  000032                  .WORD   26.             ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
004004  000034                  .WORD   28.             ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
004006  000036                  .WORD   30.             ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
004010                  TXRXLE::                        ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
                                .EVEN                   ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
                        ;*****************************************************************************
                        ;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
                        ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBERS WHICH CAN BE USED AS SUCH OR
                        ;*    AS OFFSETS WHEN ACCESSING A TABLE OF BYTES.
                        ;*****************************************************************************
004010                  TXRLNB::                        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
004010      000                 .BYTE   0               ;TX/RX LINE FOR RX/TX LINE 0.
004011      001                 .BYTE   1.              ;TX/RX LINE FOR RX/TX LINE 1.
004012      002                 .BYTE   2               ;TX/RX LINE FOR RX/TX LINE 2.
004013      003                 .BYTE   3.              ;TX/RX LINE FOR RX/TX LINE 3.
004014      004                 .BYTE   4.              ;TX/RX LINE FOR RX/TX LINE 4.
004015      005                 .BYTE   5.              ;TX/RX LINE FOR RX/TX LINE 5.
004016      006                 .BYTE   6.              ;TX/RX LINE FOR RX/TX LINE 6.
004017      007                 .BYTE   7.              ;TX/RX LINE FOR RX/TX LINE 7.
004020      010                 .BYTE   8.              ;TX/RX LINE FOR RX/TX LINE 8.
004021      011                 .BYTE   9.              ;TX/RX LINE FOR RX/TX LINE 9.
```

```
1437 004022      012                        .BYTE   10.              ;TX/RX LINE FOR RX/TX LINE 10.
1438 004023      013                        .BYTE   11.              ;TX/RX LINE FOR RX/TX LINE 11.
1439 004024      014                        .BYTE   12.              ;TX/RX LINE FOR RX/TX LINE 12.
1440 004025      015                        .BYTE   13.              ;TX/RX LINE FOR RX/TX LINE 13.
1441 004026      016                        .BYTE   14.              ;TX/RX LINE FOR RX/TX LINE 14.
1442 004027      017                        .BYTE   15.              ;TX/RX LINE FOR RX/TX LINE 15.
1443 004030                          TXRLNE::                         ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1444                                        .EVEN                    ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1445                                ;*****************************************************************************
1446                                ;* TABLE OF TX/RX LINE NUMBER ASSOCIATIONS IN STAGGERED LOOPBACK.
1447                                ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
1448                                ;*    WHEN ACCESSING A TABLE OF WORDS.
1449                                ;* THIS IS A TABLE OF DATA FOR READING ONLY.  USE TO LOAD THE ABOVE TABLE.
1450                                ;*    NOTE: MUST CONVERT FROM BYTES TO WORDS WHEN LOADING ABOVE TABLE.
1451                                ;*****************************************************************************
1452 004030                          STGTRB::                         ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
1453 004030      004                        .BYTE   4.               ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1454 004031      006                        .BYTE   6.               ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1455 004032      000                        .BYTE   0                ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1456 004033      002                        .BYTE   2.               ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1457 004034      014                        .BYTE   12.              ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1458 004035      016                        .BYTE   14.              ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1459 004036      010                        .BYTE   8.               ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1460 004037      012                        .BYTE   10.              ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1461 004040      024                        .BYTE   20.              ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1462 004041      026                        .BYTE   22.              ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1463 004042      020                        .BYTE   16.              ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1464 004043      022                        .BYTE   18.              ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1465 004044      034                        .BYTE   28.              ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1466 004045      036                        .BYTE   30.              ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1467 004046      030                        .BYTE   24.              ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1468 004047      032                        .BYTE   26.              ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1469                                        .EVEN                    ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1482 004050                                ERRTBL                           L$ERRTBL::
     004050
     004050  000000                 ERRTYP::        .WORD   0
     004052  000000                 ERRNBR::        .WORD   0
     004054  000000                 ERRMSG::        .WORD   0
     004056  000000                 ERRBLK::        .WORD   0
1483
1484                                        .EVEN
```

GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.

```
                        .SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
                        ;*******************************************************************************
                        ;*      THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
                        ;*      GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM.  THE FOUR
                        ;*      ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
                        ;*
                        ;*      SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
                        ;*             SAVE THE GPR CONTENTS FOR LATER RESTORATION.
                        ;*      PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
                        ;*             THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
                        ;*             OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
                        ;*      PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
                        ;*             EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
                        ;*
                        ;*      DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
                        ;*      OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
                        ;*
                        ;*             SP     -> RET PC INTO PREG05 ROUTINE.
                        ;*             SP+2   -> GPR R0 CONTENTS.
                        ;*             SP+4   -> GPR R1 CONTENTS.
                        ;*             SP+6   -> GPR R2 CONTENTS.
                        ;*             SP+8   -> GPR R3 CONTENTS.
                        ;*             SP+10  -> GPR R4 CONTENTS.
                        ;*             SP+12  -> GPR R5 CONTENTS.
                        ;*             SP+14  -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
                        ;*
                        ;*      EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
                        ;*      THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
                        ;*      TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
                        ;*      ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
                        ;*      THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
                        ;*      OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
                        ;*      AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
                        ;*      SAVE CALL.  THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
                        ;*      VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
                        ;*      THE SAVE MACRO WAS CALLED.
                        ;*******************************************************************************
```

GPR FRAME ACCESS EQUATES

```
1524                                    .SBTTL GPR FRAME ACCESS EQUATES
1525                                            ;+++
1526                                            ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME.  THESE ARE THE
1527                                            ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGO5
1528                                            ;ROUTINE.
1529                                            ;---
1530
1531            000036                          LPCSLT==        36      ;OFFSET FOR LAST RETURN PC.
1532            000016                          PCSLOT==        16      ;OFFSET FOR RETURN PC.
1533            000014                          R5SLOT==        14      ;OFFSET FOR R5.
1534            000012                          R4SLOT==        12      ;OFFSET FOR R4.
1535            000010                          R3SLOT==        10      ;OFFSET FOR R3.
1536            000006                          R2SLOT==        6       ;OFFSET FOR R2.
1537            000004                          R1SLOT==        4       ;OFFSET FOR R1.
1538            000002                          R0SLOT==        2       ;OFFSET FOR R0.
```

```
1540                    .SBTTL  GLOBAL MACRO DEFINITION         - SAVE -
1541                    ;*******************************************************************************
1542                    ;*      THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1543                    ;*      CONTENTS OF THE GPRS R0 THRU R5.
1544                    ;*
1545                    ;* INPUTS:       SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1546                    ;*               R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1547                    ;*
1548                    ;* OUTPUTS:      GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1549                    ;*               TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1550                    ;*
1551                    ;* CALLING SEQUENCE:    SAVE
1552                    ;*
1553                    ;* COMMENTS:     NO ARGUMENTS ARE ALLOWED.
1554                    ;*               THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1555                    ;*
1556                    ;* SUBORDINATE ROUTINES CALLED: PREG05.
1557                    ;*******************************************************************************
1558
1559                            .MACRO  SAVE
1560                            .LIST
1561                                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBPT.
1562                            .NLIST
1563                            .ENDM   SAVE
```

```
1565                            .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
1566                            ;*************************************************************************
1567                            ;*      THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO.  IT IS
1568                            ;*      CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1569                            ;*      CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1570                            ;*      RETURNING TO PREGO5 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1571                            ;*
1572                            ;* INPUTS:       ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1573                            ;*               R0SLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1574                            ;*                       SLOT OFFSETS BEFORE CALLING THIS MACRO.
1575                            ;*
1576                            ;* OUTPUTS:      THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1577                            ;*
1578                            ;* CALLING SEQUENCE:    PASS    R0,R1,...
1579                            ;*
1580                            ;* COMMENTS:     ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1581                            ;*               FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1582                            ;*                       PASS    R1
1583                            ;*                       PASS    R4,R0,R2
1584                            ;*               THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1585                            ;*               CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1586                            ;*               THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1587                            ;*
1588                            ;*               THE MACRO CALL
1589                            ;*                       PASS    R0,R3
1590                            ;*               EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1591                            ;*                       MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
1592                            ;*                       MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
1593                            ;*                       JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1594                            ;*               IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1595                            ;*               THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1596                            ;*               WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL
1597                            ;*
1598                            ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREGO5, VALUE ON STACK.)
1599                            ;*************************************************************************
1600
1601                                    .MACRO  PASS    A,B,C,D,E,F
1602                                    .IRP    X,<A,B,C,D,E,F>
1603                                    .IF     NB,X
1604                                    .LIST
1605                                            MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
1606                                    .NLIST
1607                                    .ENDC
1608                                    .ENDM
1609                                    .LIST
1610                                            JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1611                                    .NLIST
1612                                    .ENDM   PASS
```

```
                        .SBTTL GLOBAL SUBROUTINE                - PREG05
                        ;*************************************************************************
                        ;*      PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
                        ;*
                        ;* INPUTS:      THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
                        ;*              GPR R5.  (I.E.- MACROS USE "JSR  R5,PREG05".)
                        ;*
                        ;* OUTPUTS:     REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
                        ;*
                        ;*CALLING SEQUENCE:     SAVE            ;MACRO EXPANSION CALLS ; REG05.
                        ;*                      [SUBROUTINE CODE]...
                        ;*                      PASS            ;MACRO EXPANSION RECALLS PREG05.
                        ;*
                        ;*COMMENTS:     THIS ROUTINE IS RE-ENTRANT.
                        ;*
                        ;*              PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
                        ;*              REGISTER SAVE AREA ON THE STACK.  USE THE PASS GPRN MACRO
                        ;*              TO RETURN GPR VALUES INTACT.
                        ;*              USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
                        ;*              [EXAMPLE:       MOV     VALUE,ROSLOT(SP)        ]
                        ;*              MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
                        ;*
                        ;*SUBORDINATE ROUTINES CALLED:  NONE.
                        ;*************************************************************************

004060                  PREG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
004060  010446                  MOV     R4,-(SP)        ;SAVE R4
004062  010346                  MOV     R3,-(SP)        ;SAVE R3
004064  010246                  MOV     R2,-(SP)        ;SAVE R2
004066  010146                  MOV     R1,-(SP)        ;SAVE R1
004070  010046                  MOV     R0,-(SP)        ;SAVE R0
004072  010546                  MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
004074  016605  000014          MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

004100  004736                  JSR     PC,@(SP)+       ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
                                                        ;FROM THE PREG05 CALL, PUTTING THE PRESENT
                                                        ;PC ON THE STACK AS A RETURN ADDRESS INTO
                                                        ;THIS (PREG05) ROUTINE.

                                ;***
                                ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
                                ;"RETURN" [JSR  PC,@(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
                                ;-

004102  012605          PREGRT:: MOV    (SP)+,R5        ;PUT RETURN PC IN R5.
004104  012600                  MOV     (SP)+,R0        ;RESTORE R0.
004106  012601                  MOV     (SP)+,R1        ;RESTORE R1.
004110  012602                  MOV     (SP)+,R2        ;RESTORE R2.
004112  012603                  MOV     (SP)+,R3        ;RESTORE R3.
004114  012604                  MOV     (SP)+,R4        ;RESTORE R4.

004116  000205                  RTS     R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05,
                                                          ;RESTORING R5 IN THE PROCESS.
```

```
                        .SBTTL  GLOBAL TEXT SECTION


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

                        ;
                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;
                              DEVTYP  <DHU-11>
004120                                                          L$DVTYP::
004120                                                                .ASCIZ  /DHU-11/
004120  104     110     125
004123  055     061     061
004126  000

                                                                .EVEN


                        ; TEST DESCRIPTION
                        ;
                        DESCRIPT        <DHU-11 FUNC TST PART2>
004130                                                          L$DESC::
004130                                                                .ASCIZ  /DHU-11 FUNC TST PAR
004130  104     110     125
T2/
004133  055     061     061
004136  040     106     125
004141  116     103     040
004144  124     123     124
004147  040     120     101
004152  122     124     062
004155  000

                                                                .EVEN
                        .EVEN
```

```
1715
1716                    .NLIST BIN
1717
1718
1719                    ; ********* FORMAT STATEMENTS USED IN PRINT CALLS *********
1720
1721
1722 004156  EF0503:: .ASCIZ /%T%N/
1723 004163  EF1601:: .ASCIZ /%A  %T%A, TEST ABORTED %N/
1724 004215  EF5801:: .ASCIZ /%A     RXTIMER VALUE USED WAS :%D3%A (D)%N/
1725 004270  EF5901:: .ASCIZ /%A     EXPECTED :%D3%A(D)%N/
1726 004320  EF5902:: .ASCIZ /%A     ACTUAL   :%D3%A(D)%N/
1727 004350  EF6401:: .ASCIZ /%A                                %D2%N/
1728 004417  EF7801:: .ASCIZ /%T%A ON LINE %D2%A DECIMAL.%N/
1729 004455  EF9001:: .ASCIZ /%A  UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/
1730 004537  EF9002:: .ASCIZ /%A    CODE IS ASSOCIATED WITH LINE: %D2%N/
1731 004611  EF9003:: .ASCIZ /%A      CODE IS: %O3%N/
1732 004640  EF9004:: .ASCIZ /%A    %T%A VALUE: %O3%N/
1733 004670  EF9005:: .ASCIZ /%A    %T%A VALUE: NONE%N/
1734 004721  EF9006:: .ASCIZ /%A  %T%A %D2%A(D)%N/
1735 004745  EF9010:: .ASCIZ /%A    NUMBER OF ERRORS DETECTED ON LINE %D2%A IS %D5%N/
1736 005034  EF9019:: .ASCIZ /%A  %T%A %O6%N/
1737 005053  EF9301:: .ASCIZ /%A %T%D2%A(D),  BMP CODE REPORTED :%O3%A(O)%N/
1738 005131  EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
1739 005231  MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/
1740                 .EVEN
1741                                .LIST BIN
```

```
1750
1751                    .NLIST BIN
1752
1753
1754                            ;********* GLOBAL ERROR MESSAGES ************
1755
1756 005262  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
1757 005320  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
1758 005403  EM4001:: .ASCIZ /DMA_START BIT TEST FAILED/
1759 005435  EM4002:: .ASCIZ /DMA_START BIT BAD ON LINE: /
1760 005471  EM4101:: .ASCIZ /DMA_ABORT BIT TEST FAILED/
1761 005523  EM4102:: .ASCIZ /DMA_ABORT BIT BAD ON LINE: /
1762 005557  EM4103:: .ASCIZ /DMA_START BIT FOUND SET AFTER DMA ABORTED ON LINE: /
1763 005643  EM4201:: .ASCIZ /DMA_ERROR BIT TEST FAILED/
1764 005675  EM4202:: .ASCIZ /DMA_ERROR BIT BAD/
1765 005717  EM4901:: .ASCIZ /OAUTO (INACTIVE) BIT TEST FAILED/
1766 005760  EM4902:: .ASCIZ /  OAUTO BIT BAD ON LINE: /
1767 006012  EM5001:: .ASCIZ /OAUTO (ACTIVE) BIT TEST FAILED/
1768 006051  EM5101:: .ASCIZ /IAUTO (INACTIVE) TEST FAILED/
1769 006106  EM5102:: .ASCIZ /IAUTO BIT FOUND SET ON LINE: /
1770 006144  EM5103:: .ASCIZ /IAUTO BIT BAD ON LINE: /
1771 006174  EM5201:: .ASCIZ /IAUTO (ACTIVE) TEST FAILED/
1772 006227  EM5202:: .ASCIZ /IAUTO BIT FOUND CLR ON LINE: /
1773 006265  EM5301:: .ASCIZ /FIFO VALID DATA TEST FAILED/
1774 006321  EM5302:: .ASCIZ /FIFO BAD, DATA FIELD CORRUPTED, TEST USED LINE:/
1775 006401  EM5303:: .ASCIZ /BMP CODE FOUND IN FIFO, TEST INVAILDATED/
1776 006452  EM5401:: .ASCIZ \FIFO 3/4 ALARM (INACTIVE) TEST FAILED\
1777 006520  EM5402:: .ASCIZ /FIFO BAD, ALARM SIGNAL DEFECTIVE/
1778 006561  EM5501:: .ASCIZ \FIFO 3/4 ALARM (ACTIVE) TEST FAILED\
1779 006625  EM5601:: .ASCIZ \FIFO 3/4 ALARM (ACTIVE/INACTIVE) TEST FAILED\
1780 006702  EM5701:: .ASCIZ \FIFO 1/2 LEVEL (ACTIVE/INACTIVE) TEST FAILED\
1781 006757  EM5801:: .ASCIZ /RXTIMER TEST FAILED/
1782 007003  EM5802:: .ASCIZ /RXTIMER BAD, RX-INT DELAYED BY WRONG NUMBER OF MILLISECONDS/
1783 007077  EM5803:: .ASCIZ \RXTIMER BAD, RX-INT DIDN'T OCCUR IMMEDIATELY WITH RXFIFO 3/4 FULL\
1784 007201  EM5804:: .ASCIZ /RXTIMER BAD, RX-INT OCCURED WITH RXTIMER VALUE ZERO/
1785 007265  EM5805:: .ASCIZ /RXTIMER BAD, TIME-OUT OCCURED WAITING FOR RX-INT/
1786 007346  EM5901:: .ASCIZ / TX-ACTION FIFO TEST FAILED/
1787 007402  EM5902:: .ASCIZ / TX_ACTION FIFO BAD, TX-ACTION RECIEVED FROM THE WRONG LINE/
1788 007476  EM5903:: .ASCIZ / TX_ACTION FIFO BAD, INCORRECT NUMBER OF TX-ACTIONS FOUND/
1789 007570  EM5904:: .ASCIZ / TX_ACTION FIFO BAD, TX-ACTION FIFO WOULD NOT EMPTY/
1790 007654  EM5905:: .ASCIZ / TX INTERRUPT OCCURED AFTER THE TX_ACTION FIFO HAD BEEN EMPTIED/
1791 007754  EM6001:: .ASCIZ /TX FIFO TEST FAILED/
1792 010000  EM6002:: .ASCIZ /INCORRECT VALUE IN FIFOSIZE REG/
1793 010040  EM6003:: .ASCIZ /MORE THAN ONE TX-INT OCCURED,FROM A FULL TXFIFO/
1794 010120  EM6004:: .ASCIZ /TX FIFO BAD, RECIEVED CHAR INCORRECT/
1795 010165  EM6005:: .ASCIZ /TX FIFO BAD, CHARACTER RECIEVED ON WRONG LINE/
1796 010243  EM6006:: .ASCIZ /TX FIFO BAD, TOO FEW CHARS RECIEVED/
1797 010307  EM6401:: .ASCIZ /BREAK GENERATION TEST FAILED/
1798 010344  EM6402:: .ASCIZ /  BREAK NOT RECEIVED ON LINE(S):/
1799 010405  EM6601:: .ASCIZ /NO OVERRUN ERROR TEST FAILED/
1800 010442  EM6602:: .ASCIZ /  OVERRUN ERROR REPORTED WHEN NONE FORCED/
1801 010514  EM6701:: .ASCIZ /OVERRUN ERROR TEST FAILED/
1802 010546  EM6702:: .ASCIZ /  NO OVERRURN ERROR REPORTED, OVERRUN FORCED/
1803 010623  EM9009:: .ASCIZ /EXPECTED OR CORRECT/
1804 010647  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1805 010673  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
1806 010767  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA,VALID STUCK SET),/
```

```
1807 011044                .ASCIZ / REMAINDER OF TEST SKIPPED./
1808 011100  EM9026::      .ASCIZ /     LPR CONTENTS: /
1809 011124  EM9104::      .ASCIZ /  UNEXPECTED DATA FOUND IN FIFO FROM LINE: /
1810 011200  EM9301::      .ASCIZ /BMP CODES WERE REPORTED DURING THIS DIAGNOSTIC/
1811 011257  EM9302::      .ASCIZ /BMP CODE FOUND IN TEST /
1812 011307  EM9303::      .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1813 011354  EM9304::      .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1814                       .EVEN
1815                                .LIST BIN
```

```
1824
1825
1826                            .SBTTL  GLOBAL ERROR REPORT SECTION
1827
1828                            ;++
1829                            ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1830                            ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1831                            ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1832                            ;--
```

```
1834                            .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ERC101 -
1835                            ;**************************************************************************
1836                            ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1837                            ;*      INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
1838                            ;*      ACCESS TEST). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
1839                            ;*      THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
1840                            ;*      BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP).A MESSAGE INDICATING
1841                            ;*      THAT THE DHU MAY BE AT THE WRONG UNIBUS ADDRESS IS ALSO PRINTED.
1842                            ;*
1843                            ;* INPUTS:      R5 - ERROR FLAG WORD.
1844                            ;*                   IF BIT 0 IS SET, A READ ERROR OCCURED.
1845                            ;*                   IF BIT 1 IS SET, A WRITE ERROR OCCURED.
1846                            ;*
1847                            ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
1848                            ;*
1849                            ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ERC101" AS THE MESSAGE POINTER
1850                            ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1851                            ;*
1852                            ;* COMMENTS:
1853                            ;*
1854                            ;* SUBORDINATE ROUTINES USED: NONE.
1855                            ;**************************************************************************
1856
1857 011430                             BGNMSG  ERC101                          ERC101::
1858 011430                     SAVE                    ;SAVE THE GPR CONTENTS.
      011430  004567  172424                    JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
1859
1860 011434  012700  000100      MOV     #BIT06,R0       ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
1861 011440  046700  170544      BIC     OPTION,R0       ;TRY AND CLEAR THE FLAG.
1862 011444  001036              BNE     6$              ;EXIT IF OPTION NOT SELECTED.
1863                            ;*
1864                            ; REPORT EXTENDED ERROR INFOMATION
1865                            ;*
1866
1867 011446  032705  000001      BIT     #BIT0,R5        ;TEST FOR READ ERROR.
1868 011452  001410              BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
1869 011454                      PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
      011454  012746  011546                                                    MOV     #MSG1,-(SP)
      011460  012746  000001                                                    MOV     #1,-(SP)
      011464  010600                                                            MOV     SP,R0
      011466  104414                                                            TRAP    C$PNTB
      011470  062706  000004                                                    ADD     #4,SP
1870 011474  032705  000002  2$:    BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
1871 011500  001410              BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
1872 011502                      PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
      011502  012746  011624                                                    MOV     #MSG2,-(SP)
      011506  012746  000001                                                    MOV     #1,-(SP)
      011512  010600                                                            MOV     SP,R0
      011514  104414                                                            TRAP    C$PNTB
      011516  062706  000004                                                    ADD     #4,SP
1873 011522                      4$:    PRINTX  #MSG3           ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
      011522  012746  011703                                                    MOV     #MSG3,-(SP)
      011526  012746  000001                                                    MOV     #1,-(SP)
      011532  010600                                                            MOV     SP,R0
      011534  104415                                                            TRAP    C$PNTX
      011536  062706  000004                                                    ADD     #4,SP
```

```
1874 011542                            6$:     PASS                    ;RESTORE THE GPR CONTENTS.
     011542  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1875 011544                                    ENDMSG
     011544                                                             L10002:
     011544  104423                                                             TRAP    C$MSG
1876
1877 011546      045     101     102  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
     011551      125     123     040
     011554      124     111     115
     011557      105     055     117
     011562      125     124     040
     011565      124     122     101
     011570      120     040     103
     011573      101     125     123
     011576      105     104     040
     011601      102     131     040
     011604      122     105     101
     011607      104     040     101
     011612      124     124     105
     011615      115     120     124
     011620      056     045     116
     011623      000
1878 011624      045     101     102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
     011627      125     123     040
     011632      124     111     115
     011635      105     055     117
     011640      125     124     040
     011643      124     122     101
     011646      120     040     103
     011651      101     125     123
     011654      105     104     040
     011657      102     131     040
     011662      127     122     111
     011665      124     105     040
     011670      101     124     124
     011673      105     115     120
     011676      124     056     045
     011701      116     000
1879 011703      045     101     104  MSG3::  .ASCIZ  /%ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.%N%N/
     011706      110     125     040
     011711      115     101     131
     011714      040     102     105
     011717      040     101     124
     011722      040     124     110
     011725      105     040     127
     011730      122     117     116
     011733      107     040     125
     011736      116     111     102
     011741      125     123     040
     011744      101     104     104
     011747      122     105     123
     011752      123     056     045
     011755      116     045     116
     011760      000
1880
1881                                    .EVEN
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0503 -
                        ;**********************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
                        ;*      MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER, PROVIDED
                        ;*      EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE MESSAGE IN R1.
                        ;*                      INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;**********************************************************************

011762                          BGNMSG  ER0503
011762                                                                  ER0503::
011762  012700  000100                  MOV     #BIT6,R0        ;TRY TO CLEAR THE
011766  046700  170216                  BIC     OPTION,R0       ;EXTND ERROR REPORTING FLAG
011772  001011                  BNE     2$              ;EXIT IF FLAG NOT SET.

011774                                  PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
011774  010146                                                          MOV     R1,-(SP)
011776  012746  004156                                                  MOV     #EF0503,-(SP)
012002  012746  000002                                                  MOV     #2,-(SP)
012006  010600                                                          MOV     SP,R0
012010  104414                                                          TRAP    C$PNTB
012012  062706  000006                                                  ADD     #6,SP

012016                          2$:     ENDMSG
012016                                                          L10003.
012016  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1603 -
                        ;*************************************************************************
                        ;*      THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
                        ;*      MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
                        ;*      ABOUT TO BE ABORTED, PROVIDED EXTENDED ERROR INFOMATION HAS BEEN
                        ;*      REQUESTED, OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED.
                        ;*
                        ;* INPUTS:      R1 - CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
                        ;*              ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
                        ;*              THE TEST THAT IS BEING PERFORMED, EG DMA, BREAK ETC.
                        ;*
                        ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
                        ;*              "TESTNAME TEST ABORTED"
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABLE "ER1603" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;*************************************************************************
012020                          BGNMSG  ER1603
012020                                                                  ER1603::
012020  004567  172034                  SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
                                        JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.

012024  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
012030  046700  170154          BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
012034  001024                  BNE     2$              ;EXIT IF FLAG NOT SET.


012036                          PRINTB  #EF0503,R1      ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
012036  010146                                                  MOV     R1,-(SP)
012040  012746  004156                                          MOV     #EF0503,-(SP)
012044  012746  000002                                          MOV     #2,-(SP)
012050  010600                                                  MOV     SP,R0
012052  104414                                                  TRAP    C$PNTB
012054  062706  000006                                          ADD     #6,SP

012060  016702  171770          MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
012064                          PRINTB  #EF1601,R2      ;PRINT "TEST ABORTED" MESSAGE.
012064  010246                                                  MOV     R2,-(SP)
012066  012746  004163                                          MOV     #EF1601,-(SP)
012072  012746  000002                                          MOV     #2,-(SP)
012076  010600                                                  MOV     SP,R0
012100  104414                                                  TRAP    C$PNTB
012102  062706  000006                                          ADD     #6,SP

012106                  2$:     PASS                    ;RESTORE THE CONTENTS OF THE GPRS.
012106  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
012110                          ENDMSG
012110                                                                  L10004:
012110  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER5801 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING ROUTINE WHICH PRINTS THE MESSAGE PASSED
                        ;*      AS A PARAMETER IN R1, AND THE RXTIMER VALUE IN R2, PROVIDED
                        ;*      EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
                        ;*      THIS ROUTINE IS USED BY THE RXTIMER TEST.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
                        ;*              R2 - RXTIMER VALUE.
                        ;*
                        ;* OUTPUTS:     THE MESSAGE FOLLOWED BY THE RXTIMER VALUE ARE PRINTED AT
                        ;*              THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL ER5801 AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION AND THE
                        ;*              RXTIMER VALUE IS PRINTED AS A 3 DIGIT DECIMAL NUMBER.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

012112                          BGNMSG  ER5801
012112                                                                  ER5801::
012112  032767  000100  170070    BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
012120  001422                  BEQ     2$              ;EXIT WITH "TEST FAILED" MESSAGE IF NOT.

012122                          PRINTB  #EF0503,R1
012122  010146                                                          MOV     R1,-(SP)
012124  012746  004156                                                  MOV     #EF0503,-(SP)
012130  012746  000002                                                  MOV     #2,-(SP)
012134  010600                                                          MOV     SP,R0
012136  104414                                                          TRAP    C$PNTB
012140  062706  000006                                                  ADD     #6,SP
012144                          PRINTB  #EF5801,R2
012144  010246                                                          MOV     R2,-(SP)
012146  012746  004215                                                  MOV     #EF5801,-(SP)
012152  012746  000002                                                  MOV     #2,-(SP)
012156  010600                                                          MOV     SP,R0
012160  104414                                                          TRAP    C$PNTB
012162  062706  000006                                                  ADD     #6,SP
012166                  2$:     ENDMSG
012166                                                          L10005:
012166  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER5901 -
                        ;**************************************************************************
                        ;*      THIS IS AN ERROR REPORTING ROUTINE WHICH PRINTS AN ADDITIONAL
                        ;*      MESSAGE IN ADDITION TO THE "TEST FAILED" MESSAGE AND ALSO A
                        ;*      MESSAGE SHOWING THE EXPECTED VALUE OF A PIECE OF DATA AND THE
                        ;*      ACTUAL VALUE OF THAT DATA.  THE DATA IS PRINTED AS A 3 DIGIT
                        ;*      DECIMAL NUMBER.
                        ;*
                        ;* INPUTS :     R1 - EXPECTED VALUE OF DATA.
                        ;*              R2 - ACTUAL VALUE OF DATA.
                        ;*              R3 - ADDRESS OF THE MESSAGE TO PRINT.
                        ;*
                        ;* OUTPUTS :    MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER5901" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS :
                        ;*
                        ;* SUBORDINATE ROUTINES USED : NONE.
                        ;**************************************************************************

012170                          BGNMSG  ER5901
012170                                                                  ER5901::

012170                          SAVE                    ;SAVE THE GPR CONTENTS.
012170  004567  171664                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

012174  032767  000100  170006          BIT     #BIT06,OPTION   ;EXIT THE ROUTINE IF EXTENDED
012202  001433                  BEQ     60$             ;ERROR REPORTING IS NOT REQUESTED.

                                ;*
                                ; REPORT EXTENDED ERROR INFOMATION
                                ;-

012204                          PRINTB #EF0503,R3       ;PRINT THE MESSAGE.
012204  010346                                                          MOV     R3,-(SP)
012206  012746  004156                                                  MOV     #EF0503,-(SP)
012212  012746  000002                                                  MOV     #2,-(SP)
012216  010600                                                          MOV     SP,R0
012220  104414                                                          TRAP    C$PNTB
012222  062706  000006                                                  ADD     #6,SP
012226                          PRINTX #EF5901,R1       ;PRINT THE "EXPECTED VALUE" MESSAGE.
012226  010146                                                          MOV     R1,-(SP)
012230  012746  004270                                                  MOV     #EF5901,-(SP)
012234  012746  000002                                                  MOV     #2,-(SP)
012240  010600                                                          MOV     SP,R0
012242  104415                                                          TRAP    C$PNTX
012244  062706  000006                                                  ADD     #6,SP
012250                          PRINTX #EF5902,R2       ;PRINT THE "ACTUAL VALUE" MESSAGE.
012250  010246                                                          MOV     R2,-(SP)
012252  012746  004320                                                  MOV     #EF5902,-(SP)
012256  012746  000002                                                  MOV     #2,-(SP)
012262  010600                                                          MOV     SP,R0
012264  104415                                                          TRAP    C$PNTX
012266  062706  000006                                                  ADD     #6,SP

```

```
2018 012272                        60$:    PASS                    ;RESTORE THE GPR CONTENTS.
     012272  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2019
2020 012274                                ENDMSG
     012274                                                                 L10006:
     012274  104423                                                                TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER6001 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING ROUTINE WHICH PRINTS OUT A MESSAGE
                        ;*      AT THE CONSOLE INFORMING THE OPERATOR OF AN ERROR ON A PARTICULAR
                        ;*      LINE.  THE ROUTINE ALSO PRINTS OUT A MESSAGE INFORMING THE OPERATOR
                        ;*      OF WHAT DATA WAS "EXPECTED" AND WHAT "ACTUAL" DA A WAS FOUND, IN THE
                        ;*      FORM OF A 3 DIGIT DECIMAL NUMBER.
                        ;*      IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED THEN ONLY THE
                        ;*      "TEST FAILED" MESSAGE WILL BE PRINTED.
                        ;*
                        ;* INPUTS :     R1 - EXPECTED DATA
                        ;*              R2 - ACTUAL DATA
                        ;*              R3 - ADDRESS OF THE MESSAGE TO PRINT
                        ;*              R4 - LINE NUMBER ON WHICH THE ERROR OCCURED
                        ;*
                        ;* OUTPUTS :    MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE
                        ;*              " 'MESSAGE' ON LINE @"
                        ;*                      "EXPECTED :" @@
                        ;*                      "  ACTUAL :" @@
                        ;*
                        ;* CALLING SEQUENCE :   INCLUDE THE LABLE "ER6001" AS THE ERROR ROUTINE
                        ;*                      POINTER PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED : NONE
                        ;*************************************************************************

                                BGNMSG  ER6001
012276                                                                  ER6001::

012276  032767  000100  167704          BIT     @BIT06,OPTION   ;EXIT THE ROUTINE IF EXTENDED
012304  001434                          BEQ     60$             ;ERROR REPORTING IS NOT REQUESTED.
                        ;
                        ; REPORT EXTENDED ERROR INFOMATION
                        ;
012306                                  PRINTB  @EF7801,R3,R4   ;PRINT THE MESSAGE WITH THE LINE NUMBER.
012306  010446                                                          MOV     R4,-(SP)
012310  010346                                                          MOV     R3,-(SP)
012312  012746  004417                                                  MOV     @EF7801,-(SP)
012316  012746  000003                                                  MOV     @3,-(SP)
012322  010600                                                          MOV     SP,R0
012324  104414                                                          TRAP    C$PNTB
012326  062706  000010                                                  ADD     @10,SP
012332                                  PRINTX  @EF5901,R1      ;PRINT THE "EXPECTED" DATA MESSAGE.
012332  010146                                                          MOV     R1,-(SP)
012334  012746  004270                                                  MOV     @EF5901,-(SP)
012340  012746  000002                                                  MOV     @2,-(SP)
012344  010600                                                          MOV     SP,R0
012346  104415                                                          TRAP    C$PNTX
012350  062706  000006                                                  ADD     @6,SP
012354                                  PRINTX  @EF5902,R2      ;PRINT THE "ACTUAL" DATA MESSAGE.
012354  010246                                                          MOV     R2,-(SP)
012356  012746  004320                                                  MOV     @EF5902,-(SP)
012362  012746  000002                                                  MOV     @2,-(SP)
012366  010600                                                          MOV     SP,R0
012370  104415                                                          TRAP    C$PNTX
012372  062706  000006                                                  ADD     @6,SP
```

```
2059 012376                          60$:    ENDMSG
     012376                                                  L10007:
     012376  104423                                                  TRAP    C$MSG
```

```
2061                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE           - ER6401 -
2062                    ;*****************************************************************************
2063                    ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
2064                    ;*      INFORMATION AFTER THE ERROR MESSAGE HEADER, PROVIDED EXTENDED ERROR
2065                    ;*      REPORTING HAS BEEN ENABLED.
2066                    ;*      THIS SUBROUTINE IS PASSED A CPR CONTAINING FLAGS WHICH INDICATE
2067                    ;*      THE LINE(S) FOR WHICH THE ERROR CONDITION SHOULD BE REPORTED.
2068                    ;*
2069                    ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO BE PRINTED BY THIS ROUTINE.
2070                    ;*              R5 - CONTAINS THE ERROR FLAGS, (1 FLAG PER LINE).
2071                    ;*
2072                    ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
2073                    ;*
2074                    ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE MESSAGE IN R1.
2075                    ;*                      INCLUDE THE LABEL "ER6401" AS THE MESSAGE POINTER
2076                    ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2077                    ;*
2078                    ;* COMMENTS:    THE OUTPUT FORMAT OF THIS MESSAGE IS:
2079                    ;*                      TEXT MESSAGE
2080                    ;*                              @NN
2081                    ;*                              @NN
2082                    ;*                              ...
2083                    ;*              WHERE EACH "@NN" IS THE NUMBER OF A LINE WITH THE ERROR.
2084                    ;*
2085                    ;* SUBORDINATE ROUTINES USED: NONE.
2086                    ;*****************************************************************************
2087
2088 012400                         BGNMSG  ER6401
       012400                                                                   ER6401::
2089 012400                         SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
       012400  004567  171454                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2090
2091                            ;+
2092                            ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
2093                            ;-
2094 012404  032767  000100  167576     BIT     @BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
2095 012412  001433                     BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
2096                                                             ;DURING THE SOFTWARE QUESTIONS.
2097
2098 012414  005002                     CLR     R2              ;CLEAR LINE NUMBER TO ZERO.
2099 012416  012703  000020             MOV     @NUMLNS,R3      ;SET UP MAX LINE COUNT.
2100 012422                             PRINTB  @EF0503,R1      ;PRINT MESSAGE.
       012422  010146                                                           MOV     R1,-(SP)
       012424  012746  004156                                                   MOV     @EF0503,-(SP)
       012430  012746  000002                                                   MOV     @2,-(SP)
       012434  010600                                                           MOV     SP,R0
       012436  104414                                                           TRAP    C$PNTB
       012440  062706  000006                                                   ADD     @6,SP
2101 012444  000241                     CLC                     ;CLEAR CARRY.
2102 012446  006205                 2$: ASR     R5              ;SHIFT FLAG OUT INTO CARRY BIT.
2103 012450  103011                     BCC     4$              ;SKIP ERROR REPORT IF CLEAR.
2104 012452                             PRINTB  @EF6401,R2      ;PRINT MESSAGE.
       012452  010246                                                           MOV     R2,-(SP)
       012454  012746  004350                                                   MOV     @EF6401,-(SP)
       012460  012746  000002                                                   MOV     @2,-(SP)
       012464  010600                                                           MOV     SP,R0
       012466  104414                                                           TRAP    C$PNTB
```

```
                012470  062706  000006                                                          ADD     #6,SP
     2105       012474  005202          4$:     INC     R2              ;INCREMENT LINE COUNT.
     2106       012476  020302                  CMP     R3,R2           ;CHECK IF MAX LINE COUNT EXCEEDED.
     2107       012500  001362                  BNE     2$              ;LOOP IF NOT DONE.
     2108       012502                  60$:    PASS                    ;RESTORE THE SAVED CONTENTS OF THE GPRS.
                012502  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
     2109       012504                  ENDMSG
                012504                                                          L10010:
                012504  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER7801 -
                        ;*****************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
                        ;*      MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER.  A LINE NUMBER
                        ;*      IS INCLUDED AT THE END OF THE MESSAGE. THE MESSAGE IS PRINTED ONLY IF
                        ;*      EXTENDED ERROR REPORTING IS REQUESTED.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
                        ;*              R3 - NUMBER OF LINE ON WHICH ERROR OCCURRED.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE MESSAGE IN R1.
                        ;*                      LOAD THE LINE NUMBER INTO R3.
                        ;*                      INCLUDE THE LABEL "ER7801" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*****************************************************************************

012506                          BGNMSG  ER7801
012506                                                                  ER7801::

                        ;*
                        ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                        ;-
012506  032767  000100  167474          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
012514  001412                        BEQ     2$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                ;DURING THE SOFTWARE QUESTIONS.

012516                                  PRINTB  #EF7801,R1,R3   ;PRINT THE MESSAGE.
012516  010346                                                                  MOV     R3,-(SP)
012520  010146                                                                  MOV     R1,-(SP)
012522  012746  004417                                                          MOV     #EF7801,-(SP)
012526  012746  000003                                                          MOV     #3,-(SP)
012532  010600                                                                  MOV     SP,R0
012534  104414                                                                  TRAP    C$PNTB
012536  062706  000010                                                          ADD     #10,SP

012542                          2$:     ENDMSG
012542                                                          L10011:
012542  104423                                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9001 -
                        ;**********************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS AN UNEXPECTED
                        ;*      CODE WHICH HAS BEEN FOUND IN THE DUT CSR.  THIS CODE CAN BE A BMP
                        ;*      CODE, A SELF-TEST CODE, OR A MODEM STATUS CODE.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*              R2 - SINGLE BYTE CODE WHICH HAS BEEN READ FROM THE DUT.
                        ;*              R4 - LINE NUMBER ASSOCIATED WITH THE CODE.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9001" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;**********************************************************************

012544                          BGNMSG  ER9001
012544                                                                  ER9001::

                            ;*
                        ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                        ;-
012544  032767  000100  167436      BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
012552  001433              BEQ     2$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                    ;DURING THE SOFTWARE QUESTIONS.

012554                          PRINTB  #EF9001,R1      ;REPORT TYPE OF CODE FOUND.
012554  010146                                                          MOV     R1,-(SP)
012556  012746  004455                                                  MOV     #EF9001,-(SP)
012562  012746  000002                                                  MOV     #2,-(SP)
012566  010600                                                          MOV     SP,R0
012570  104414                                                          TRAP    C$PNTB
012572  062706  000006                                                  ADD     #6,SP
012576                          PRINTX  #EF9002,R4      ;REPORT THE LINE NUMBER OF THE CODE.
012576  010446                                                          MOV     R4,-(SP)
012600  012746  004537                                                  MOV     #EF9002,-(SP)
012604  012746  000002                                                  MOV     #2,-(SP)
012610  010600                                                          MOV     SP,R0
012612  104415                                                          TRAP    C$PNTX
012614  062706  000006                                                  ADD     #6,SP
012620                          PRINTX  #EF9003,R2      ;REPORT THE CODE WHICH WAS FOUND.
012620  010246                                                          MOV     R2,-(SP)
012622  012746  004611                                                  MOV     #EF9003,-(SP)
012626  012746  000002                                                  MOV     #2,-(SP)
012632  010600                                                          MOV     SP,R0
012634  104415                                                          TRAP    C$PNTX
012636  062706  000006                                                  ADD     #6,SP

012642                  2$:     ENDMSG
012642                                                          L10012:
012642  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9002 -
                        ;**************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
                        ;*      TRANMSISSION AND RECEPTION TESTS.  IT REPORTS THE TYPE OF ERROR WHICH
                        ;*      HAS OCCURRED WHEN INCORRECT DATA IS RECEIVED FROM THE DUT.  THIS
                        ;*      ROUTINE ALSO REPORTS THE READ AND EXPECTED DATA VALUES.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*              R2 - DATA BYTE READ FROM THE DUT.
                        ;*              R3 - LINE NUMBER MULTIPLIED BY 2.
                        ;*              R4 - EXPECTED DATA BYTE, BIT 15 SET IF "NONE".
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9002" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: PRTLPR.
                        ;**************************************************************************

012644                          BGNMSG  ER9002
012644                                                                  ER9002::
                        ;*
                        ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                        ;-
012644  032767  000100  167336          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
012652  001462          BEQ     62$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                ;DURING THE SOFTWARE QUESTIONS.

012654  006203          ASR     R3              ;CALCULATE THE LINE NUMBER.
012656  042702  177400  BIC     #177400,R2      ;MASK OUT ALL BUT DATA IN READ CHAR.
012662                  PRINTB  #EF9006,R1,R3   ;PRINT THE FIRST LINE OF THE MESSAGE.
012662  010346                                                          MOV     R3,-(SP)
012664  010146                                                          MOV     R1,-(SP)
012666  012746  004721                                                  MOV     #EF9006,-(SP)
012672  012746  000003                                                  MOV     #3,-(SP)
012676  010600                                                          MOV     SP,R0
012700  104414                                                          TRAP    C$PNTB
012702  062706  000010                                                  ADD     #10,SP
012706                  PRINTX  #EF9004,#EM9010,R2      ;PRINT ACTUAL DATA.
012706  010246                                                          MOV     R2,-(SP)
012710  012746  010647                                                  MOV     #EM9010,-(SP)
012714  012746  004640                                                  MOV     #EF9004,-(SP)
012720  012746  000003                                                  MOV     #3,-(SP)
012724  010600                                                          MOV     SP,R0
012726  104415                                                          TRAP    C$PNTX
012730  062706  000010                                                  ADD     #10,SP
012734  005704          TST     R4              ;CHECK FOR "NONE" CODE SET IN EXPECTED DATA.
012736  100414          BMI     2$              ;BRANCH TO PRINT "NONE" MESSAGE IF FLAG SET.
012740                  PRINTX  #EF9004,#EM9009,R4      ;PRINT EXPECTED DATA.
012740  010446                                                          MOV     R4,-(SP)
012742  012746  010623                                                  MOV     #EM9009,-(SP)
012746  012746  004640                                                  MOV     #EF9004,-(SP)
012752  012746  000003                                                  MOV     #3,-(SP)
```

```
                012756  010600                                                          MOV     SP,R0
                012760  104415                                                          TRAP    C$PNTX
                012762  062706  000010                                                  ADD     #10,SP
     2219       012766  000412                  BR      60$             ;EXIT THIS ROUTINE.
     2220       012770                  2$:     PRINTX  #EF9005,#EM9009 ;PRINT MESSAGE INDICATING NO EXPECTED DATA.
                012770  012746  010623                                                  MOV     #EM9009,-(SP)
                012774  012746  004670                                                  MOV     #EF9005,-(SP)
                013000  012746  000002                                                  MOV     #2,-(SP)
                013004  010600                                                          MOV     SP,R0
                013006  104415                                                          TRAP    C$PNTX
                013010  062706  000006                                                  ADD     #6,SP
     2221       013014  004767  002266          60$:    JSR     PC,PRTLPR       ;PRINT CONTENTS OF THE LPR REGISTER.
     2222       013020                  62$:    ENDMSG
                013020                                                          L10013:
                013020  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9004 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS ERROR SUMMARIES
                        ;*      FOR LINES WHICH HAVE EXCEEDED THE SPECIFIED MAXIMUM NUMBER OF
                        ;*      INDIVIDUAL RECEPTION ERRORS, PROVIDED EXTENDED ERROR REPORTING HAS
                        ;*      BEEN REQUESTED BY THE OPERATOR.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*               ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
                        ;*               ERSMRF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
                        ;*
                        ;* OUTPUTS:     A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9004" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*              THE CONTENTS OF GPR'S R2, R3, R4, AND R5 ARE DESTROYED.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

013022                          BGNMSG  ER9004
013022                                                                          ER9004::
013022  012700  000100                   MOV     #BIT06,R0       ;TRY TO CLEAR THE
013026  046700  167156                   BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
013032  001040                           BNE     6$              ;EXIT IF FLAG NOT SET.
013034                                   PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
013034  012746  010673                                                   MOV     #EM9014,-(SP)
013040  012746  004156                                                   MOV     #EF0503,-(SP)
013044  012746  000002                                                   MOV     #2,-(SP)
013050  010600                                                           MOV     SP,R0
013052  104414                                                           TRAP    C$PNTB
013054  062706  000006                                                   ADD     #6,SP
013060  005002                           CLR     R2              ;CLEAR THE LINE COUNTER.
013062  016703  167524                   MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
013066  005004                           CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
013070  000241                   2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
013072  006003                           ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
013074  103013                           BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
013076                                   PRINTX  #EF9010,R2,ERCNTB(R4)
013076  016446  002614                                                   MOV     ERCNTB(R4),-(SP)
013102  010246                                                           MOV     R2,-(SP)
013104  012746  004745                                                   MOV     #EF9010,-(SP)
013110  012746  000003                                                   MOV     #3,-(SP)
013114  010600                                                           MOV     SP,R0
013116  104415                                                           TRAP    C$PNTX
013120  062706  000010                                                   ADD     #10,SP
013124  012405                   4$:     MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
013126  005202                           INC     R2              ;INCREMT THE LINE COUNTER.
013130  005703                           TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
013132  001356                           BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
013134                           6$:     ENDMSG
013134                                                                  L10014:
013134  104423                                                           TRAP    C$MSG
```

```
                                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9101 -
                                        ;**************************************************************************
                                        ;*      THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
                                        ;*      WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
                                        ;*      ASCII MESSAGE.
                                        ;*
                                        ;* INPUTS:       R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
                                        ;*               R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
                                        ;*
                                        ;* OUTPUTS:      A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                                        ;*
                                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
                                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                                        ;*
                                        ;* COMMENTS:     THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                                        ;*
                                        ;* SUBORDINATE ROUTINES USED: NONE.
                                        ;**************************************************************************

013136                                          BGNMSG  ER9101
013136                                                                                  ER9101::

013136  012700  000100                           MOV    #BIT06,R0       ;TRY TO CLEAR THE
013142  046700  167042                           BIC    OPTION,R0       ;EXT'D ERROR REPORTING FLAG
013146  001012                                  BNE    2$              ;EXIT IF FLAG NOT SET.


013150                                          PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
013150  010146                                                                                  MOV     R1,-(SP)
013152  010246                                                                                  MOV     R2,-(SP)
013154  012746  004721                                                                          MOV     #EF9006,-(SP)
013160  012746  000003                                                                          MOV     #3,-(SP)
013164  010600                                                                                  MOV     SP,R0
013166  104414                                                                                  TRAP    C$PNTB
013170  062706  000010                                                                          ADD     #10,SP

013174                                  2$:     ENDMSG
013174                                                                                  L10015:
013174  104423                                                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
                        ;*      THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
                        ;*      THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
                        ;*      PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
                        ;*
                        ;* INPUTS:      R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
                        ;*              R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
                        ;*
                        ;* OUTPUTS:     THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
                        ;*              OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;^
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

                              BGNMSG  ER9301
                                                                                ER9301::
                              SAVE                      ;SAVE THE GPRS ON THE STACK.
013176  004567  170656            JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.

                              MOV     #BIT06,R0         ;TRY TO CLEAR THE
                              BIC     OPTION,R0         ;EXT'D ERROR REPORTING FLAG
                              BNE     60$               ;EXIT IF FLAG NOT SET.

                              PRINTB  #EF0503,R1        ;REPORT UNEXPECTED BMP CODES FOUND.
013214  010146                                                          MOV     R1,-(SP)
013216  012746  004156                                                  MOV     #EF0503,-(SP)
013222  012746  000002                                                  MOV     #2,-(SP)
013226  010600                                                          MOV     SP,R0
013230  104414                                                          TRAP    C$PNTB
013232  062706  000006                                                  ADD     #6,SP

                              MOV     #BMPCQB,R3        ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
                              MOV     #EM9302,R5        ;GET THE MESSAGE TO BE REPORTED.
                      2$:     MOV     (R3)+,R1          ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
                              MOV     (R3)+,R4          ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
                              JSR     PC,50$            ;GO REPORT THE BMP CODE.
                              CMP     R3,R2             ;CHECK IF ALL CODES HAVE BEEN REPORTED.
                              BLO     2$                ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
                        ;
                        ; CHECK IF OVERFLOW HAS OCCURRED.
                        ;  THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
                        ;  LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
                        ;  CELL.
                        ;-
                              CMP     R2,#BMPCQE-4      ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
                              BNE     60$               ;EXIT IF NOT AT THE LAST LOCATION.
                              TST     2(R2)             ;CHECK FOR A BMP CODE IN THE LAST CELL
                              BEQ     60$               ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
                              MOV     (R3)+,R1          ;GET THE TEST NUMBER OFF THE QUEUE.
                              MOV     (R3),R4           ;GET THE BMP CODE OFF THE QUEUE.
                              MOV     #EM9303,R5        ;SELECT THE MESSAGE TO BE REPORTED.
```

```
2343 013306                             PRINTX  #EF9302         ;REPORT OVERFLOW CONDITION.
     013306  012746  005131                                                 MOV     #EF9302,-(SP)
     013312  012746  000001                                                 MOV     #1,-(SP)
     013316  010600                                                         MOV     SP,R0
     013320  104415                                                         TRAP    C$PNTX
     013322  062706  000004                                                 ADD     #4,SP
2344 013326  004767  000002             JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2345 013332  000414                     BR      60$             ;EXIT.
2346
2347 013334                     50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
     013334  010446                                                         MOV     R4,-(SP)
     013336  010146                                                         MOV     R1,-(SP)
     013340  010546                                                         MOV     R5,-(SP)
     013342  012746  005053                                                 MOV     #EF9301,-(SP)
     013346  012746  000004                                                 MOV     #4,-(SP)
     013352  010600                                                         MOV     SP,R0
     013354  104415                                                         TRAP    C$PNTX
     013356  062706  000012                                                 ADD     #12,SP
2348 013362  000207                     RTS     PC              ;RETURN.
2349 013364                     60$:    PASS                    ;RESTORE THE GPR CONTENTS.
     013364  004736                                     JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2350
2351 013366                             ENDMSG
     013366                                                             L10016:
     013366  104423                                                         TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION


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

```
                        .SBTTL  GLOBAL SUBROUTINE                - ALTFLD -
                        ;** ************************************************************************
                        ;*              - ALTER DEVICE REGISTER FIELDS ROUTINE -
                        ;*      THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
                        ;*      REGISTER FOR THE SPECIFIED LINES.  THIS ROUTINE CAN BE USED TO SET
                        ;*      OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
                        ;*      USE EXAMPLES:   SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
                        ;*                      CLEAR TX.DMA BITS ON ALL LINES.
                        ;*
                        ;* INPUTS:       R1 - ADDRESS OF THE REGISTERS TO ALTER.
                        ;*               R2 - BIT FIELDS SET TO DESIRED STATES.
                        ;*               R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
                        ;*               R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
                        ;*               CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
                        ;*               IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
                        ;*
                        ;* OUTPUTS:      DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
                        ;*                   CSR IND.ADR.REG FIELD - DESTROYED.
                        ;*
                        ;* CALLING SEQUENCE:     JSR     PC,ALTFLD
                        ;*
                        ;* COMMENTS:     THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
                        ;*               WITH NUMBERS LOWER THAN THE HIGHES SPECIFIED LINE.
                        ;*               THIS ROUTINE DOES NOT READ THE CSR.
                        ;*
                        ;* SUBROUTINES CALLED: NONE.
                        ;-- ************************************************************************

013370                  ALTFLD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013370  004567  170464                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                        ;*
                        ; SET UP TO LOOP FOR EACH LINE:
                        ;    PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
                        ;    SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
                        ;-
013374  010400                          MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
013376  005100                          COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
013400  040002                          BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
013402  016705  166650                  MOV     IESTAT,R5       ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
                        ;*
                        ; LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
                        ;    REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
                        ; EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
                        ;    ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
                        ;-
013406  000241                          CLC                     ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
013410  006003          2$:             ROR     R3              ;GET THE LINE SELECT BIT FOR THIS LINE.
013412  103006                        BCC     4$              ;SKIP SETUP IF LINE IS NOT SELECTED.
013414  010577  166606                  MOV     R5,@CSRA        ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
013420  011100                          MOV     (R1),R0         ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
013422  040400                          BIC     R4,R0           ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
013424  050200                          BIS     R2,R0           ;OR IN THE NEW STATES OF THE FIELDS.
013426  010011                          MOV     R0,(R1)         ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
013430  005205          4$:             INC     R5              ;SET LINE NUMBER TO THE NEXT LINE.
013432  005703                          TST     R3              ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
013434  001365                        BNE     2$              ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

C6


```
2424
2425 013436                         60$:    PASS                        ;RESTORE GPRS.
     013436  004736                                           JSR     PC,@(SP)+           ;RETURN TO PREG05 SUBRT.
2426 013440  000207                         RTS     PC                  ;RETURN TO CALLING ROUTINE.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - ASLNTL -
                        ;** ************************************************************************
                        ;*              - SETUP ASSOCIATED LINE NUMBER TABLES ROUTINE -
                        ;*      THIS ROUTINE SETS UP THE TWO TABLES WHICH ARE CONTAIN INFORMATION
                        ;*      ABOUT THE TX/RX LINE WHICH IS ASSOCIATED WITH A PARTICULAR RX/TX
                        ;*      LINE.  ONE TABLE IS A TABLE OF WORDS WHICH CONTAINS WORD OFFSET
                        ;*      VALUES AND THE OTHER TABLE IS A TABLE OF BYTES WHICH CONTAINS
                        ;*      LINE NUMBER VALUES.
                        ;*
                        ;* INPUTS:      LOPBCK - STORAGE FOR THE TYPE OF LOOPBACK ON THE DUT.
                        ;*              NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
                        ;*              STGTRB - LABEL AT BASE OF STAGGERED LINE ASSOCIATION TBL.
                        ;*              TXRLNB - LABEL AT BASE OF BYTE TX/RX LINE NUMBER TABLE.
                        ;*              TXRXLB - LABEL AT BASE OF WORD TX/RX LINE NUMBER TABLE.
                        ;*              TXRXLE - LABEL AT END OF WORD TX/RX LINE NUMBER TABLE.
                        ;*
                        ;* OUTPUTS:     TXRXL, TXRLN - TABLES INITIALIZED FOR SELECTED LOOPBACK.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,ASLNTL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

                        ASLNTL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013442  004567  170412                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013446  126727  166550  000002          CMPB    LOPBCK,#2       ;TEST FOR STAGGERED LOOPBACK.
013454  001411                        BEQ     4$              ;GO SET UP STAGGERED TABLE IF STAGGERED LPBCK.
                        ;*
                        ; SET UP THE WORD TABLE FOR NON-STAGGERED LOOPBACK.
                        ;-
013456  005005                        CLR     R5              ;CLEAR THE LINE COUNTER
013460  010565  003750  2$:           MOV     R5,TXRXLB(R5)   ;SET UP A WORD OF THE TABLE.
013464  005205                        INC     R5
013466  005205                        INC     R5              ;SET LINE COUNTER TO NEXT LINE OFFSET.
013470  020527  000040                CMP     R5,#2*NUMLNS    ;TEST FOR ALL LINES DONE.
013474  002771                        BLT     2$              ;LOOP UNTIL ALL LINES DONE.
013476  000411                        BR      8$              ;GO SET UP THE BYTE TABLE.
                        ;*
                        ; SET UP THE WORD TABLE FOR STAGGERED LOOPBACK.
                        ;-
013500  012701  004030  4$:           MOV     #STGTRB,R1      ;SET UP THE SOURCE POINTER.
013504  012702  003750                MOV     #TXRXLB,R2      ;SET UP THE DESTINATION POINTER.
013510  112122  6$:           MOVB    (R1)+,(R2)+     ;MOVE A BYTE INTO THE TABLE.
013512  105022                        CLRB    (R2)+           ;CLEAR THE UPPER BYTE OF WORD TABLE ENTRY.
013514  020227  004010                CMP     R2,#TXRXLE      ;COMPARE POINTER WITH END ADR OF TABLE.
013520  002773                        BLT     6$              ;LOOP IF NOT AT END YET.
                        ;*
                        ; SET UP THE BYTE TABLE BASED ON THE WORD ASSOCIATION TABLE.
                        ;
013522  012701  003750  8$:           MOV     #TXRXLB,R1      ;SET UP THE SOURCE POINTER.
013526  012702  004010                MOV     #TXRLNB,R2      ;SET UP THE DESTINATION POINTER.
013532  012103  10$:          MOV     (R1)+,R3        ;GET THE WORD OFFSET VALUE FROM WORD TABLE.
013534  006203                        ASR     R3              ;DIVIDE BY 2 TO GET LINE NUMBER VALUE.
013536  110322                        MOVB    R3,(R2)+        ;LOAD THE BYTE LINE NUMBER INTO TABLE.
013540  020127  004010                CMP     R1,#TXRXLE      ;COMPARE SOURCE POINTER WITH ADR OF TABLE END.
```

```
2484 013544  002772                    BLT     10$             ;LOOP IF NOT AT END OF TABLE YET.
2485
2486 013546                    60$:     PASS                    ;RESTORE GPRS.
     013546  004736                             JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
2487 013550  000207                    RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - CALMSL -
                        ;** ************************************************************************
                        ;*              - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE -
                        ;*      THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
                        ;*      ROUTINE.  THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
                        ;*      WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
                        ;*      THE MSLOOP ROUTINE.  THIS ROUTINE CALIBRATES THE COUNT BY USING THE
                        ;*      LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
                        ;*      THE DELAY COUNT MUST BE USED.
                        ;*
                        ;*
                        ;* INPUTS:      MSLCNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
                        ;*                       VALUE FROM PREVIOUS CALIBRATION.
                        ;*              MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
                        ;*              TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
                        ;*              CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
                        ;*
                        ;* OUTPUTS:     CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
                        ;*              MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
                        ;*                       UNCHANGED IF NO LTC IS AVAILABLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CALMSL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
                        ;-- ************************************************************************

013552                  CALMSL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013552  004567  170302                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013556  005067  000210          CLR     62$                     ;CLEAR THE 2ND TIME FLAG.
                        ;*
                        ; SYNCHRONIZE WITH THE LTC.
                        ;-
013562  012705  000001          2$:     MOV     #1,R5           ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
013566  005000                  CLR     R0                      ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
013570  012767  000001  166516  MOV     #1,TIMER1               ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
013576  005767  166512  4$:     TST     TIMER1                  ;CHECK FOR COUNTER HAVING GONE TO ZERO.
013602  001410                  BEQ     6$                      ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
013604  005200                  INC     R0                      ;COUNT THIS ITERATION OF THE INNER LOOP.
013606  001373                  BNE     4$                      ;LOOP IF COUNTER HAS NOT TURNED OVER.
013610  005305                  DEC     R5                      ;DECREMENT THE INNER LOOP COUNTER.
013612  003371                  BGT     4$                      ;LOOP IF OUTER LOOP COUNT NOT UP.
                                ;*
                                ; IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
                                ; LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
                                ;-
013614  005067  166472                  CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
013620  000241                  CLC                             ;INDICATE FAILURE FOR RETURN.
013622  000461                  BR      60$                     ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                        ;*
                        ; WE ARE NOW SYNCHRONIZED WITH THE LTC.
                        ; SET UP FOR THE CALIBRATION LOOP.
                        ;-
013624  012704  002314  6$:     MOV     #TIMER1,R4              ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```
2545 013630  005001                        CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2546 013632  005002                        CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2547 013634  005003                        CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2548 013636  012714  000001                MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2549
2550 013642  016705  166460        8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2551 013646  011400                10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2552 013650  010067  000120                MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2553 013654  040200                        BIC     R2,R0           ;LEAVE ALL THE BITS.
2554 013656  020003                        CMP     R0,R3           ;COMPARE AGAINST ZERO.
2555 013660  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2556 013662  001406                        BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2557 013664  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2558 013666  001367                        BNE     10$             ;LOOP IF MS NOT UP.
2559 013670  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2560 013672  001363                        BNE     8$              ;KEEP LOOPING.
2561 013674  004767  001160                JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2562                                ;+
2563                                ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2564                                ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2565                                ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2566                                ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2567                                ;-
2568 013700  005401                12$:    NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2569 013702  016702  166420                MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2570 013706  010203                        MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2571 013710  160502                        SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2572 013712  010204                        MOV     R2,R4           ;  AND ADD TO ACCUMULATOR LSWORD.
2573 013714  005005                        CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2574 013716  005301                14$:    DEC     R1              ;CHECK R1 FOR 0 CONDITION
2575 013720  100403                        BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2576 013722  060304                        ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2577 013724  005505                        ADC     R5              ; LOOPS PER OUTER LOOP BY
2578 013726  000773                        BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2579                                ;+
2580                                ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2581                                ;-
2582 013730  016701  166370        16$:    MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2583 013734  010403                        MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2584 013736  010502                        MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2585 013740  004767  003336                JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2586 013744  103402                        BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2587 013746  004767  001106                JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2588 013752  010167  166350        18$:    MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2589 013756  005167  000010                COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2590 013762  001277                        BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2591 013764  000261                        SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2592
2593 013766                        60$:    PASS                    ;RESTORE GPRS.
     013766  004736                                JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
2594 013770  000207                        RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2595
2596 013772  000000                62$:    .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2597 013774  000000                64$:    .WORD   0               ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - CHKBMP -
                                ;++ ************************************************************************
                                ;*                        - CHECK IF CHARACTER IS A BMP CODE -
                                ;*      THIS SUBROUTINE IS USED TO CHECK FOR BMP CODES.
                                ;*      IF A BMP CODE IS DETECTED, IT WILL BE SAVED ON THE QUEUE TO BE REPORTED
                                ;*      LATER.  THE CARRY IS USED AS A FLAG TO INDICATE A CODE HAS BEEN FOUND.
                                ;*
                                ;* INPUTS:      R2 - CONTAINS THE DATA TO BE CHECKED.
                                ;*
                                ;* OUTPUTS:     R1 - CONTIANS THE MESSAGE TO BE REPORTED.
                                ;*              ERRBLK - CONTAINS THE ERROR REPORTING ROUTINE.
                                ;*              CARRY BIT IS USED TO INDICATE A BMP CODE FOUND, CARRY SET.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CHKBMP
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
                                ;-- ************************************************************************

013776                          CHKBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013776  004567  170056                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014002  012700  170301                 MOV     #170301,R0              ;SET UP THE FLAGS OF A BMP CODE.
014006  040200                         BIC     R2,R0                   ;TRY TO CLEAR THE BMP CODE FLAGS.
014010  001011                        BNE     2$                      ;IF NOT A BMP CODE, EXIT WITH FAILURE.
014012  004767  002462                JSR     PC,SAVBMP               ;SAVE THE BMP CODE ON THE QUEUE.
014016  012701  006401                 MOV     #EM5303,R1              ;PASS THE MESSAGE TO BE REPORTED.
014022  012767  012020  170026         MOV     #ER1603,ERRBLK          ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
014030  000261                         SEC                             ;PASS FLAG TO INDICATE SUCCESS, BMP CODE FOUND.
014032  000401                        BR      60$                     ;EXIT.
014034  000241                2$:      CLC                             ;PASS FLAG TO INDICATE FAILURE.
014036                          60$:    PASS    R1                      ;RESTORE GPRS, EXCEPT
014036  010166  000004                                  MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
014042  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                        ;R1 - CONTAINS THE ADDRESS OF ERROR MESSAGE.
                                                                        ;CARRY BIT - SET INDICATES SUCCESS.
014044  000207                         RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - CKRXTM -
                        ;++ ****************************************************************************
                        ;*                        - CHECK RX-INT DELAY TIME -
                        ;*      THIS SUBROUTINE IS USED IN THE RXTIMER TEST AND IT CHECKS THAT THE
                        ;*      RX-INT WAS DELAYED BY  +/- 20%  OF THE VALUE SET IN THE RXTIMER REG.
                        ;*      IF THE ACTUAL DELAY TIME IS NOT WITHIN THIS MARGIN THEN AN ERROR
                        ;*      IS REPORTED.
                        ;*
                        ;* INPUTS:      R1 - REMAINING NUMBER OF MILLI SECS OF THE TIME-OUT VALUE
                        ;*                   THE TIME-OUT VALUE BEING  4*(RXTIMER VALUE + 1).
                        ;*              R2 - RXTIMER VALUE.
                        ;*              ERRNBR - SET TO ERROR NUMBER OF FIRST ERROR IN THIS ROUTINE.
                        ;*              ERRBLK - SET UP BY THIS ROUTINE.
                        ;*              EXOERR - "EXIT ON ERROR" FLAG.
                        ;*
                        ;* OUTPUTS:     ERRBLK - MAY BE ALTERED.
                        ;*              EXOERR - "EXIT ON ERROR" FLAG SET IF AN ERROR DETECTED AND
                        ;*                       EXTENDED ERROR REPORTING NOT REQUESTED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CKRXTM
                        ;*
                        ;* COMMENTS:    THIS ROUTINE REPORTS ONE ERROR WITH THE NUMBER IN ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: ER5901,UNSDIV.
                        ;****************************************************************************
014046                  CKRXTM:: SAVE
014046  004567  170006                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014052  010204                  MOV     R2,R4           ;COPY THE RXTIMER VALUE.
014054  005204                  INC     R4              ;CALCULATE THE TIME-OUT VALUE,
014056  006304                  ASL     R4              ; AS 4*(RXTIMER VALUE+1)
014060  006304                  ASL     R4
014062  160104                  SUB     R1,R4           ;CALCULATE THE NUMBER OF MS THE RX-INT WAS DELAYED.
                        ;+
                        ; CALCULATE 20% OF THE RXTIMER VALUE.
                        ;-
014064  012701  000005          MOV     #5,R1           ;PASS THE DIVISOR TO THE SUBR.
014070  010203                  MOV     R2,R3           ;LOAD THE RXTIMER VALUE INTO THE LSW OF THE DIVIDEND.
014072  005002                  CLR     R2              ;CLEAR THE MSW OF THE DIVIDEND.
014074  004767  003202          JSR     PC,UNSDIV       ;DIVIDE THE RXTIMER VALUE BY 5.
014100  010302                  MOV     R3,R2           ;RESTORE THE RXTIMER VALUE.
014102  010305                  MOV     R3,R5           ;COPY THE RXTIMER VALUE.
014104  160103                  SUB     R1,R3           ;REDUCE IT BY 20% TO OBTAIN THE LOLIMIT OF THE
                                                        ;PERMISSIBLE TIME DELAY.
014106  060105                  ADD     R1,R5           ;INCREASE THE RXTIMER VALUE BY 20% AND ADD A
014110  062705  000002          ADD     #2,R5           ;FURTHER 2 MS TO OBTAIN THE HILIMIT OF THE
                                                        ;PERMISSIBLE TIME DELAY.
                        ;+
                        ; CHECK IF THE RX-INT OCCURED WITHIN THE CALCULATED PERMISSIBLE LIMITS.
                        ;-
014114  020504                  CMP     R5,R4           ;COMPARE HILIM WITH ACTUAL RX-INT DELAY.
014116  002402                  BLT     2$              ;REPORT THE ERROR IF THE DELAY WAS TOO LONG.
014120  020304                  CMP     R3,R4           ;COMPARE LOLIM WITH ACTUAL RX-INT DELAY.
014122  002417                  BLT     60$             ;AVOID ERROR IF DELAY WAS WITHIN THE LIMITS.
                        ;+
                        ; REPORT THE ERROR, INCORRECT DELAY ON RX-INT.
                        ;-
```

```
2690 014124  010201                  2$:    MOV    R2,R1            ;PASS THE EXPECTED VALUE OF THE RX-INT DELAY.
2691 014126  010402                         MOV    R4,R2            ;PASS THE ACTUAL VALUE OF THE RX-INT DELAY.
2692 014130  012703  007003                 MOV    #EM5802,R3       ;PASS THE MESSAGE,
2693                                                                 ; "RXTIMER BAD, INCORRECT MILLI SEC DELAY ON
2694                                                                 ;  RX-INT".
2695 014134  012767  012170  167714         MOV    #ER5901,ERRBLK   ;SET UP THE ERROR REPORTING ROUTINE.
2696 014142                               ERROR
     014142  104460                                                                  TRAP    C$ERROR
2697
2698 014144  032767  000100  166036         BIT    #BIT06,OPTION    ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
2699 014152  001003                       BNE    60$              ;AVOID SETTING THE FLAG IF IT HAS.
2700 014154  012767  000001  166070        MOV    #1,EXOERR        ;SET THE "EXIT ON ERROR" FLAG.
2701 014162                         60$:  PASS
     014162  004736                                JSR    PC,@(SP)+                ;RETURN TO PREG05 SUBRT.
2702 014164  000207                       RTS    PC               ;RETURN FROM SUBROUTINE.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - CKTRAP -
                        ;**************************************************************************
                        ;*      CHECK TRAP ROUTINE -
                        ;*      THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
                        ;*      WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION.
                        ;*      IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
                        ;*
                        ;* INPUTS:      R0 - SOURCE ADDRESS FOR MOVE.
                        ;*              R1 - DESTINATION ADDRESS FOR MOVE.
                        ;*              (R0) - SOURCE FOR THE MOVE.
                        ;*
                        ;* OUTPUTS:     (R1) - WRITTEN TO THE CONTENTS OF (R0).
                        ;*              CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
                        ;*              TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CKTRAP
                        ;*
                        ;* COMMENTS:    IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
                        ;*              IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;**************************************************************************

014166                  CKTRAP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
014166  004567  167666                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014172  005067  166102          CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
014176  011011                  MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
014200  005767  166074  ADRPTR:: TST     TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
014204  000261                  SEC                     ;INDICATE SUCCESS.
014206  001401                  BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
014210  000241                  CLC                     ;INDICATE FAILURE.
014212                  60$:    PASS                    ;RESTORE GPRS.
014212  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014214  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - CLNRST -
                        ;*************************************************************************
                        ;*              - CLEAN RESET OF THE DEVICE UNDER TEST -
                        ;*      THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
                        ;*      THE DUT'S SELF-TEST IS SKIPPED,AND THE FIFO IS PURGED OF ANY ERROR
                        ;*      CODES, ETC.
                        ;*      IF THE RESET DOES NOT SUCCESFULLY COMPLETE, THEN THE CARRY BIT IS
                        ;*      PASSED BACK TO THE CALLING ROUTINE (CLEAR).
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR
                        ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
                        ;*              ERRNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
                        ;*              ERRTBL- ERRTYP,ERNBR,AND ERRMSG SET UP CORRECTLY.
                        ;*
                        ;* OUTPUTS:     THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
                        ;*              CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
                        ;*              ERRBLK - VALUE MAY BE DESTROYED.
                        ;*              IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
                        ;*              TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CLNRST
                        ;*
                        ;* COMMENTS:    THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.
                        ;*              THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
                        ;*************************************************************************

014216                  CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014216  004567  167636                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; RESET THE DUT.
                        ;  THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2
                        ;-
014222  004767  002054          JSR     PC,RESETT       ;RESET THE DUT TO A KNOWN STATE.
014226  103002                  BCC     60$             ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                        ;+
                        ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
                        ;-
014230  004767  001134          JSR     PC,PUFIFO       ;PURGE THE FIFO.

014234                  60$:                            ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
014234                          PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
014234  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT;IF CLEAR, THEN ABORT THE TEST.
014236  000207          RTS     PC
```

```
2782                                    .SBTTL  GLOBAL SUBROUTINE                - CLR16W -
2783                                    ;++ ************************************************************************
2784                                    ;*                      - CLEAR SIXTEEN WORDS ROUTINE -
2785                                    ;*      THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
2786                                    ;*
2787                                    ;* INPUTS:      R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
2788                                    ;*
2789                                    ;* OUTPUTS:     (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
2790                                    ;*
2791                                    ;* CALLING SEQUENCE:    JSR     PC,CLR16W
2792                                    ;*
2793                                    ;* COMMENTS:
2794                                    ;*
2795                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
2796                                    ;-- ************************************************************************
2797
2798 014240                            CLR16W:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     014240  004567  167614                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2799 014244  012701  000020                    MOV     #16.,R1         ;SET THE LOOP COUNTER TO 16.
2800 014250  005020                    2$:     CLR     (R0)+           ;CLEAR A WORD OF MEMORY.
2801 014252  005301                            DEC     R1              ;COUNT THIS LOOP.
2802 014254  001?75                    BNE     2$              ;LOOP IF NOT 16 WORD CLEARED.
2803 014256                            60$:    PASS                    ;RESTORE GPRS.
     014256  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2804 014260  000207                            RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - DELAY -
                        ;*************************************************************************
                        ;*                              - DELAY SUBROUTINE -
                        ;*      THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI-SECONDS.
                        ;*
                        ;* INPUTS:      R4 - CONTAINS THE NUMBER OF MS TO DELAY.
                        ;*               MSLCNT.
                        ;*
                        ;* OUTPUTS:     NONE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,DELAY
                        ;*
                        ;* COMMENTS:    IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURING, CONTROL-CS WILL
                        ;*              NOT BE HONORED FOR THE DURATION OF THE DELAY.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;*************************************************************************

014262                  DELAY:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
014262  004567  167572                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014266  010401                  MOV     R4,R1           ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
014270  012702  177777          MOV     #-1,R2          ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
014274  005003                  CLR     R3              ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
014276  012704  014320          MOV     #62$,R4         ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
014302  004767  000536          JSR     PC,MSLOOP       ;DELAY THE REQUESTED # OF MS.
014306  103002                  BCC     60$             ;EXIT ROUTINE IF WE TIMED-OUT.)
014310  004767  000544          JSR     PC,OOPS         ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
014314                  60$:    PASS                    ;RESTORE GPRS.
014314  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014316  000207                  RTS     PC
                        
014320  177777          62$:    .WORD   -1              ;DUMMY, NON-ZERO WORD.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - DMABUF -
                        ;** ************************************************************************
                        ;*                          - DMA FROM ADDR BUFBAS -
                        ;*      THIS SUBROUTINE INITIATES A DMA FROM THE ADDRESS BUFBAS AND WAITS
                        ;*      FOR IT TO COMPLETE AND THE LAST CHARACTER TO BE RECIEVED.  THE LINE
                        ;*      ON WHICH TO PERFORM THE DMA AND THE NUMBER OF CHARACTERS TO TRANSMIT
                        ;*      ARE PASSED AS INPUT PARAMETERS.
                        ;*
                        ;* INPUTS:      R1 - LINE NUMBER ON WHICH TO PERFORM THE DMA.
                        ;*              R3 - NUMBER OF CHARACTERS TO TRANSMIT.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE CSR.
                        ;*              ERRNBR - CONTAINS THE ERROR NUMBER OF THE FIRST ERROR
                        ;*                       IN THIS SUBROUTINE.
                        ;*
                        ;* OUTPUTS:     CARRY - SUCCESS FLAG -  SET IF DMA COMPLEATED SUCCESFULLY.
                        ;*                                      CLEAR IF DMA-START BIT FOUND SET OR
                        ;*                                            NO TX-ACTION RECIEVED.
                        ;*              ERRNBR - CONTENTS ALTERED TO INITIAL ERRNBR + 1.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,DMABUF
                        ;*
                        ;* COMMENTS:    THIS ROUTINE RETURNS CARRY CLEAR IF AN ERROR OCCURED, WITH
                        ;*              THE APPROPRIATE ERROR NUMBER IN ERRNBR.  A NON-RELATED TEST
                        ;*              ERROR SHOULD BE REPORTED, UPON RETURN TO THE TEST, IF THE CARRY
                        ;*              FLAG IS CLEAR.  THIS ROUTINE REPORTS ERRORS,
                        ;*              INITIAL ERRNBR THRU INITIAL ERRNBR + 1.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,DODMA,WAIBIS.
                        ;** ************************************************************************
014322                  DMABUF:: SAVE
        014322  004567  167532                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014326  012702  002650          MOV     #BUFBAS,R2      ;PASS THE START ADDR OF THE BUFFER TO TX.
014332  004767  000042          JSR     PC,DODMA        ;INITIATE THE DMA.
014336  103016                  BCC     60$             ;RETURN WITH FAILURE IF DMA START BIT FOUND SET.
                        ;*
                        ; WAIT FOR THE DMA TO COMPLEATE I.E  A TX-ACTION BEING RETURNED, AND THEN WAIT
                        ; FOR THE LAST CHARACTER TO BE RECIEVED.
                        ;*
014340  005267  167506          INC     ERRNBR          ;SET THE ERROR NUMBER TO INITIAL ERRNBR + 1.
014344  012701  170536          MOV     #170536,R1      ;INDICATE TO TEST BIT15 WITH TIME OUT OF 350 MS
014350  016702  165652          MOV     CSRA,R2         ;PASS THE ADDR OF THE REG TO TEST.
014354  004767  003056          JSR     PC,WAIBIS       ;WAIT FOR THE TX-ACTION.
014360  103005                  BCC     60$             ;RETURN WITH FAILURE IF NO TX-ACTION FOUND.
014362  012704  000005          MOV     #5,R4           ;SET THE DELAY OF 5 MS.
014366  004767  177670          JSR     PC,DELAY        ;WAIT FOR THE LAST CHARACTER TO BE RECIEVED.
014372  000261                  SEC                     ;SET THE CARRY FLAG TO INDICATE SUCCESS.
014374                  60$:    PASS
        014374  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014376  000207                  RTS     PC              ;RETURN.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - DODMA -
                        ;** ************************************************************************
                        ;*                        - INITIATE DMA TRANSMISSION ROUTINE -
                        ;*      THIS ROUTINE WRITES THE DMA PARAMETER TO THE SPECIFIED DEVICE AND
                        ;*      INITIATES THE DMA TRANSMISSION.
                        ;*
                        ;* INPUTS:       R1 - LINE NUMBER ON WHICH TO INITIATE THE DMA.
                        ;*               R2 - START ADDRESS OF THE DMA BUFFER (16 BIT VIRTUAL).
                        ;*               R3 - CHARACTER COUNT OF THE DMA BUFFER.
                        ;*               CSRA - CONTAINS ADDRESS OF THE DUT CSR.
                        ;*               IESTAT - STORAGE FOR STATES OF THE INTERRUPT ENABLE BITS.
                        ;*               TXAD1A - CONTAINS ADDRESS OF DMA TX BUFFER ADDRESS REG #1.
                        ;*               TXAD2A - CONTAINS ADDRESS OF DMA TX BUFFER ADDRESS REG #2.
                        ;*               TXBFCA - CONTAINS ADDRESS OF DMA CHARACTER COUNT REGISTER.
                        ;*
                        ;* OUTPUTS:      CARRY - SUCCESS FLAG (SET IF DMA_START FOUND CLEAR).
                        ;*               DUT TBUFFAD1 - LS 16 BITS OF DMA BUFFER ADDRESS (INITIALIZED).
                        ;*               DUT TBUFFAD2 - MS 6 BITS OF DMA BUFFER ADDRESS (INITIALIZED),
                        ;*                              DMA_START BIT SET.
                        ;*               DUT TBUFFCT - DMA BUFFER CHARACTER COUNT (INITIALIZED).
                        ;*
                        ;* CALLING SEQUENCE:     JSR     PC,DODMA
                        ;*
                        ;* COMMENTS:     THIS ROUTINE ASSUMES MEMORY MANAGEMENT IS DISABLED AND
                        ;*               CLEARS THE  TWO MSB OF THE DMA ADDRESS, I.E. BITS 0 AND 1
                        ;*               OF THE TBUFFAD2 REG.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

2916 014400             DODMA:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     014400  004567  167454                     JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2917 014404  012704  000200                     MOV     #200,R4         ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.
2918
2919                            ;*
2920                            ; WRITE THE DMA PARAMETERS OUT TO THE DUT DMA REGISTERS.
2921                            ;    DISABLE INTERRUPTS.
2922                            ;    SET UP DUT CSR IND.ADR.REG FIELD.
2923                            ;    WRITE THE DMA TRANSMIT CHARACTER COUNT.
2924                            ;    WRITE THE LEAST SIGNIFICANT 16 BITS OF THE DMA BUFFER START ADDRESS.
2925                            ;    WRITE THE MOST SIGNIFICANT 6 BITS OF THE ADDRESS,
2926                            ;      SETTING THE DMA_START BIT, AND INITIATING THE DMA TRANSMISSION.
2927                            ;
2928 014410             6$:        GETPRI  R5              ;GET THE PRESENT PROCESSOR PRIORITY.
     014410  104440                                                      TRAP    C$GPRI
     014412  010005                                                      MOV     R0,R5
2929 014414                        SETPRI  #PRI07          ;DISABLE ALL HARDWARE INTERRUPTS.
     014414  012700  00034C                                              MOV     #PRI07,R0
     014420  104441                                                      TRAP    C$SPRI
2930 014422  056701  165630                      BIS     IESTAT,R1       ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
2931 014426  010177  165574                      MOV     R1,@CSRA        ;SET UP THE DUT CSR IND.ADR.REG FIELD.
2932 014432  105777  165604                      TSTB    @TXAD2A         ;TEST THE DUT DMA_START BIT.
2933 014436  000241                              CLC                     ;INDICATE FAILURE IN CASE DMA.HO BIT IS SET.
2934 014440  100411                              BPL     60$             ;EXIT WITH FAILURE IF DMA.HO BIT IS SET.
2935 014442  010377  165576                      MOV     R3,@TXBFCA      ;WRITE THE DMA CHARACTER COUNT.
2936 014446  010277  165566                      MOV     R2,@TXAD1A      ;WRITE THE LS 16 BITS OF BUFFER ADDRESS.
2937 014452  110477  165564                      MOVB    R4,@TXAD2A      ;WRITE MS 6 BITS OF ADR AND START DMA TX.
```

```
2938 014456                          SETPRI  R5          ;RESTORE THE PROCESSOR PRIORITY.
     014456  010500                                                  MOV     R5,R0
     014460  104441                                                  TRAP    C$SPRI
2939 014462  000261                   SEC                ;INDICATE SUCCESS.
2940
2941 014464                    60$:   PASS               ;RESTORE GPRS,
     014464  004736                                JSR   PC,@(SP)+          ;RETURN TO PREGOS SUBRT.
2942 014466  000207                   RTS     PC         ;  CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - FINACT -
                                ;++ ************************************************************************
                                ;*                      - FIND FIRST ACTIVE LINE -
                                ;*      THIS SUBROUTINE CALCULATES THE NUMBER OF THE FIRST ACTIVE LINE THAT
                                ;*      IS FOUND IN THE ACTIVE LINE BIT MAP ACTLNS.
                                ;*
                                ;* INPUTS:      ACTLNS - CONTAINS THE ACTIVE LINE BIT MAP.
                                ;*
                                ;* OUTPUTS:     R1 - CONTAINS THE NUMBER OF THE FIRST ACTIVE LINE.
                                ;*              R5 - CONTAINS THE BIT MAP REPRESENTATION OF THE ACTIVE LINE.
                                ;*              CARRY SET INDICATES SUCCESS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,FINACT
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

014470                          FINACT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
014470  004567  167364                   JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                ;-
014474  005001                      CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
014476  012703  000020              MOV     #NUMLNS,R3      ;GET MAX LINE NUMBER.
014502  016700  165512              MOV     ACTLNS,R0       ;GET THE ACTIVE LINE BIT MAP.
014506  012705  000001              MOV     #1,R5           ;SET UP A LINE BIT MASK.
014512  030500                  2$: BIT     R5,R0           ;LOOK FOR AN ACTIVE LINE.
014514  001006                  BNE     4$              ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
014516  006305                  ASL     R5              ;SHIFT THE BIT MASK FOR THE NEXT LINE.
014520  005201                  INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
014522  020103                  CMP     R1,R3           ;CHECK IF ALL LINES HAVE BEEN TRIED.
014524  002772                  BLT     2$              ;LOOP TO TRY THE NEXT LINE.
014526  000241                  CLC                     ;CLEAR CARRY BIT, NO ACTIVE LINE FOUND.
014530  000401                  BR      60$             ;EXIT WITH FAILURE.
014532  000261                  4$: SEC                 ;SET CARRY, SUCCESS.

014534                          60$:    PASS    R1,R5           ;RESTORE GPRS, EXCEPT
014534  010166  000004                                  MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
014540  010566  000014                                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
014544  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                ;R1 - CONTAINS THE NUMBER OF FIRST ACTIVE LINE.
                                ;R5 - CONTAINS THE BIT MAP OF THE ACTIVE LINE.
                                ;CARRY - SET INDICATES SUCCESS.
014546  000207                      RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - INDATP -
                        ;++ ************************************************************************
                        ;*                        - INITIALISE DATA PATTERN -
                        ;*      THIS SUBROUTINE IS USED TO INITIALISE AN INCREMENTAL BYTE DATA PATTERN
                        ;*      IN THE GENERAL BUFFER AREA.
                        ;*      THE DATA PATTERN WILL BE SEQUENTIAL FROM 0 TO 255 (DECIMAL).
                        ;*
                        ;* INPUTS:      BUFBAS - ADDRESS OF THE START OF THE GENERAL BUFFER AREA.
                        ;*              BUFMID - ADDRESS OF THE 255 TH LOCATION.
                        ;*
                        ;* OUTPUTS:     THE FIRST 255 LOCATIONS OF THE GENERAL BUFFER AREA CONTAIN DATA
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,INIDATP
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

014550                  INDATP:: SAVE                           ;SAVE CONTENTS OF GPRS RO THRU R5.
014550  004567  167304                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

014554  012702  002650          MOV     #BUFBAS,R2      ;INITIALIZE THE DATA PATTERN IN THE GENERAL
014560  005003                  CLR     R3              ;  DATA BUFFER TO A 256 BYTE PATTERN.
014562  110322          2$:     MOVB    R3,(R2)+        ;
014564  005203                  INC     R3              ;SELECT THE NEXT CHARACTER.
014566  020227  003250          CMP     R2,#BUFMID      ;CHECK IF WE HAVE 256 DATA PATTERNS.
014572  103773                  BLO     2$              ;

014574                  60$:    PASS                    ;RESTORE GPRS.
014574  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014576  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - INDTPX -
                                ;** ************************************************************************
                                ;*                - INITIALISE DATA PATTERN WITHOUT XON OR XOFF -
                                ;*      THIS SUBROUTINE IS USED TO INITIALISE AN INCREMENTAL BYTE DATA PATTERN
                                ;*      IN THE GENERAL BUFFER AREA.
                                ;*      THE DATA PATTERN WILL BE FROM 0 TO 255, BUT WLL EXCLUDE THE FOLLOWING
                                ;*      TWO CHARACTERS; (ASCII DC1, DC3) XON AND XOFF.  THIS WILL CAUSE THE
                                ;*      LAST TWO DATA CHARACTERS TO BE THE SAME AS THE FIRST TWO.
                                ;*
                                ;* INPUTS:      BUFBAS - ADDRESS OF THE START OF THE GENERAL BUFFER AREA.
                                ;*              BUFMID - ADDRESS OF THE 255 TH LOCATION.
                                ;*
                                ;* OUTPUTS:     THE FIRST 255 LOCATIONS OF THE GENERAL BUFFER AREA CONTAIN DATA
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,INDTPX
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

014600                          INDTPX:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
014600  004567  167254                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;*
                                ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                ;-
014604  012702  002650                MOV     #BUFBAS,R2      ;INITIALIZE THE DATA PATTERN IN THE GENERAL
014610  005003                        CLR     R3              ;  DATA BUFFER TO A 256 BYTE PATTERN.
014612  110322                2$:     MOVB    R3,(R2)+        ;
014614  105203                        INCB    R3              ;SELECT THE NEXT CHARACTER.
014616  122703  000021                CMPB    #21,R3          ;CHECK FOR AN XON CHARACTER.
014622  001001                        BNE     4$              ;BRANCH IF CHAR NOT AN XON.
014624  105203                        INCB    R3              ;FORCE THE NEXT CHARACTER.
014626  122703  000023        4$:     CMPB    #23,R3          ;CHECK FOR AN XOFF CHARACTER.
014632  001001                        BNE     6$              ;BRANCH IF NOT AN XOFF CHARACTER.
014634  105203                        INCB    R3              ;FORCE THE NEXT CHARACTER.
014636  020227  003250        6$:     CMP     R2,#BUFMID      ;CHECK IF WE HAVE 256 DATA PATTERNS.
014642  103763                        BLO     2$              ;

014644                          60$:    PASS                    ;RESTORE GPRS.
014644  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014646  000207                        RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - LINBIT -
                        ;** ************************************************************************
                        ;*                - LINE NUMBER TO BIT MAP CONVERSION SUBROUTINE -
                        ;*       THIS SUBROUTINE IS USED TO GENERATE A BIT MAP (ONE BIT OF 16 SET)
                        ;*       BASED ON A LINE NUMBER (RANGE: 1 TO 16).  ONLY THE LS 4 BITS OF THE
                        ;*       LINE NUMBER WORD ARE USED, THE OTHERS ARE MASKED OUT (SO UNMASKED
                        ;*       MSBYTES OF OUT CSRS CAN BE PASSED TO THIS ROUTINE WITHOUT ERROR).
                        ;*
                        ;* INPUTS:       R1 - LINE NUMBER (ONLY LS 4 BITS USED, OTHERS DISREGARDED).
                        ;*               BITTBL - BASE LABEL OF A 16 WORD BIT TABLE.
                        ;*
                        ;* OUTPUTS:      R0 - BIT MAP, BIT CORRESPONDING TO LINE NUMBER IS SET:
                        ;*                       IF LINE NUMBER IS 3, THEN BIT3 IS SET, ETC.
                        ;*
                        ;* CALLING SEQUENCE:     JSR     PC,LINBIT
                        ;*
                        ;* COMMENTS:     NO CHECKING IS PERFORMED TO VERIFY THAT THE LINE NUMBER IS
                        ;*               A LEGAL LINE NUMBER FOR THE DUT (IE - LESS THAN NUMLNS).
                        ;*               NOTE: THE LINE NUMBER IS NOT DESTROYED OF ALTERED, SO THIS
                        ;*                       ROUTINE CAN BE USED EASILY IN LOOPS.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;** ************************************************************************

014650                  LINBIT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014650  004567  167204                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
014654  042701  177760                 BIC     #177760,R1      ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
014660  006301                         ASL     R1              ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
014662  016100  002336                 MOV     BITTBL(R1),R0   ;GET THE SINGLE BIT BIT MAP.
014666                  60$:    PASS    R0                      ;RESTORE GPRS, EXCEPT THE FOLLOWING,
014666  010066  000002                         MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
014672  004736                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
014674  000207                 RTS     PC                      ;R0 - BIT MAP WITH LINE # BIT SET.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - MAPCNT -
                        ;** ************************************************************************
                        ;*                        - COUNT BITS IN BIT MAP ROUTINE -
                        ;*      THIS SUBROUTINE COUNTS THE NUMBER OF BITS WHICH ARE SET IN A BIT MAP.
                        ;*
                        ;* INPUTS:      R2 - THE BIT MAP FOR WHICH TO COUNT THE BITS.
                        ;*
                        ;* OUTPUTS:     R2 - COUNT OF THE NUMBER OF BITS THAT WERE SET.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MAPCNT
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

014676                  MAPCNT:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
014676  004567  167156                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014702  010201                  MOV     R2,R1
014704  001405                 BEQ     60$             ;EXIT WITH ZERO IF NO BITS ARE SET IN MAP.

014706  005002                  CLR     R2              ;CLEAR THE BIT COUNT.
014710  000261                  SEC                     ;COUNT THE LAST BIT TO BE SHIFTED OUT.

014712  005502          2$:     ADC     R2              ;COUNT THE BIT IF IT WAS SET.
014714  006301                  ASL     R1              ;SHIFT ANOTHER BIT OUT OF THE MAP.
014716  001375                  BNE     2$              ;LOOP IF ALL BITS NOT SHIFTED OUT OF MAP.

014720                  60$:    PASS    R2              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
014720  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
014724  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014726  000207                  RTS     PC              ;  R2 - COUNT OF BITS SET IN BIT MAP.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - MSLGET -
                        ;**********************************************************************
                        ;*      - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
                        ;*      THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE.  IT IS USED
                        ;*      TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD.  THE
                        ;*      CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
                        ;*      DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
                        ;*      THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
                        ;*      ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
                        ;*      UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
                        ;*      IS RETURNED BY THIS SUBROUTINE.
                        ;*
                        ;* INPUTS:      R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
                        ;*              R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
                        ;*              R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
                        ;*              R4 - ADDRESS OF THE WORD TO TEST.
                        ;*              MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
                        ;*
                        ;* OUTPUTS:     R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
                        ;*              R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
                        ;*              CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLGET
                        ;*
                        ;* COMMENTS:    THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
                        ;*              CALIBRATION IS ONLY GUARENTEED WHEN A LINC CLOCK IS AVAILABLE
                        ;*              ON THE SYSTEM.
                        ;*              THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
                        ;*              DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
                        ;*              LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
                        ;*              IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
                        ;*              THE DESIRED CONDITION BEFORE RETURNING.  IT INDICATES SUCCESS
                        ;*              IF THE CONDITION IS MET, FAILURE OTHERWISE.
                        ;*
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;**********************************************************************

014730                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014730  004567  167124                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
                        ; BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
                        ;-
014734  005102                 COM     R2              ;GET MASK OF UNUSED BITS.
014736  040203                 BIC     R2,R3           ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
                        ;*
                        ; HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
                        ;-
014740  005701                 TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
014742  001011                 BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
014744  011400                 MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
014746  010067  000070         MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
014752  040200                 BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
014754  020003                 CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
014756  000261                 SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
014760  001420                 BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

```
3178 014762  000241                    CLC                     ;INDICATE FAILURE (TIME-OUT).
3179 014764  000416                    BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
3180                                ;+
3181                                ; NON-ZERO TIME-OUT VALUE.  LOOP, WAITING FOR CONDITION OR TIME-OUT.
3182                                ;-
3183 014766  016705  165334        2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
3184 014772  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
3185 014774  010067  000042                MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
3186 015000  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
3187 015002  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
3188 015004  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
3189 015006  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
3190 015010  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
3191 015012  001367                        BNE     4$              ;LOOP IF MS NOT UP.
3192 015014  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
3193 015016  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
3194 015020  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
3195                                ;+
3196                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
3197                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD.  EXIT ROUTINE.
3198                                ;-
3199 015022  016700  000014        6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
3200 015026                        60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     015026  010066  000002                        MOV     R0,ROSLOT(SP)       ;PUT R0 IN STACK SLOT.
     015032  010166  000004                        MOV     R1,R1SLOT(SP)       ;PUT R1 IN STACK SLOT.
     015036  004736                                JSR     PC,@(SP)+           ;RETURN TO PREGOS SUBRT.
3201                                                                    ;R0 - LAST READ WORD CHECKED FOR CONDITION.
3202                                                                    ;R1   REMAINING TIME (0 IF TIME-OUT OCCURED).
3203 015040  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
3204                                ;+
3205                                ; LOCAL STORAGE.
3206                                ;-
3207 015042  000000                62$:    .WORD   0               ;STORAGE FOR THE LAST READ WORD.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - MSLOOP -
                        ;*************************************************************************
                        ;*      - TEST LOOP SUBROUTINE -
                        ;*      THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE.  IT IS USED
                        ;*      TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD.  THE
                        ;*      CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
                        ;*      DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
                        ;*      THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
                        ;*      ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
                        ;*
                        ;* INPUTS:      R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
                        ;*              R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
                        ;*              R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
                        ;*              R4 - ADDRESS OF THE WORD TO TEST.
                        ;*              MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
                        ;*
                        ;* OUTPUTS:     CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLOOP
                        ;*
                        ;* COMMENTS:    THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
                        ;*              CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
                        ;*              ON THE SYSTEM.
                        ;*              THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
                        ;*              DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
                        ;*              LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
                        ;*              IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
                        ;*              THE DESIRED CONDITION BEFORE RETURNING.  IT INDICATES SUCCESS
                        ;*              IF THE CONDITION IS MET, FAILURE OTHERWISE.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                        ;*************************************************************************

015044                  MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015044  004567  167010                          JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.

                        ;*
                        ; CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
                        ; MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.
                        ;-
015050  004767  177654          JSR     PC,MSLGET       ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

015054                  60$:    PASS                            ;RESTORE GPRS.
015054  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
015056  000207          RTS     PC                      ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
```

```
                                .SBTTL  GLOBAL SUBROUTINE                       - OOPS -
                                ;++ ************************************************************************
                                ;*                      - PROGRAM ABORT SUBROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
                                ;*      DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE.  AN ERROR MESSAGE
                                ;*      IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
                                ;*
                                ;* INPUTS:      R1 - ERROR CODE GIVING REASON FOR ABORT.
                                ;*
                                ;* OUTPUTS:     AN ERROR MESSAGE IS PRINTED.
                                ;*              A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,OOPS
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

015060                          OOPS::  SAVE                    ;SAVE CONTENTS OF GPRS RO THRU R5.
015060  004567  166774                          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                                    ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
                                        ERRSF   101,EM0101
                                                                                        TRAP    C$ERSF
                                                                                        .WORD   101
                                                                                        .WORD   EM0101
                                                                                        .WORD   0
                                    ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
                                        PRINTF  #EM0102
015074  012746  015210                                                                 MOV     #EM0102,-(SP)
015100  012746  000001                                                                 MOV     #1,-(SP)
015104  010600                                                                         MOV     SP,R0
015106  104417                                                                         TRAP    C$PNTF
015110  062706  000004                                                                 ADD     #4,SP
015114                          2$:     BREAK                   ;LOOK FOR OPERATOR CONTROL-C INPUT.
015114  104422                                                                         TRAP    C$BRK
015116  000776                          BR      2$              ;INFINITE LOOP.
015120                          60$:    PASS                    ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
015120  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
015122  000207                          RTS     PC              ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

015124    110     117     123   EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
015127    124     040     103
015132    117     115     120
015135    125     124     105
015140    122     040     110
015143    101     122     104
015146    127     101     122
015151    105     040     117
015154    122     040     123
015157    117     106     124
015162    127     101     122
015165    105     040     102
015170    125     107     040
015173    105     116     103
015176    117     125     116
015201    124     105     122
```

```
            015205      105     104     056
            015207      000
   3283     015210      045     116     045  EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
            015213      101     120     122
            015216      117     107     122
            015221      101     115     040
            015224      110     125     116
            015227      107     054     040
            015232      127     101     111
            015235      124     111     116
            015240      107     040     106
            015243      117     122     040
            015246      101     040     103
            015251      117     116     124
            015254      122     117     114
            015257      055     103     056
            015262      040     074     052
            015265      052     052     052
            015270      052     052     052
            015273      052     052     052
            015276      052     052     052
            015301      045     116     045
            015304      116     000
   3284                                     .EVEN
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - PRTLPR -
                        ;** ************************************************************************
                        ;*                      -PRINT THE CONTENTS OF THE LPR.
                        ;*      THIS ROUTINE IS USED TO PRINT OUT EXTENDED INFORMATION ON THE
                        ;*      CONTENTS OF THE LINE PARAMETER REGISTER (LPR).
                        ;*
                        ;* INPUTS:      R3 - CONTAINS THE NUMBER OF THE LINE YOU WISH TO EXAMINE.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT'S CSR.
                        ;*              IESTAT - CONTAINS THE CURRENT STATUS OF THE TX AND RX INTERRUPT
                        ;*                       ENABLE BITS IN THE DUT'S CSR.
                        ;*              LPRA - CONTAINS THE ADDRESS OF THE DUT'S LPR REGISTER.
                        ;*
                        ;* OUTPUTS:     AN EXTENDED INFORMATION MESSAGE IS PRINTED ON THE OPERATORS
                        ;*              CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PRTLPR
                        ;*
                        ;* COMMENTS:    THIS ROUTINE CHANGES THE INDIRECT ADDRESS FIELD OF THE DEVICE
                        ;*              UNDER TEST'S CSR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

015306                  PRTLPR::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
015306  004567  166546                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015312  016701  164710        MOV     CSRA,R1         ;GET THE CSR ADDRESS.
015316  016702  164710        MOV     LPRA,R2         ;GET THE LPR ADDRESS.
015322  042703  177760        BIC     #177760,R3      ;CLEAR ANY UNWANTED BITS.
015326  056703  164724        BIS     IESTAT,R3       ;SET STATE OF TX AND RX INTERRUPT ENABLE BITS.
015332  010311                MOV     R3,(R1)         ;SELECT LINE.
015334  011204                MOV     (R2),R4         ;GET CONTENTS OF THE LPR.
                               ;PRINT MESSAGE"CONTENTS OF THE LPR:NNNNNN"
015336                         PRINTX #EF9019,#EM9026,R4;PRINT OUT MESSAGE ON OPERATORS CONSOLE.
015336  010446                                                          MOV     R4,-(SP)
015340  012746  011100                                                  MOV     #EM9026,-(SP)
015344  012746  005034                                                  MOV     #EF9019,-(SP)
015350  012746  000003                                                  MOV     #3,-(SP)
015354  010600                                                          MOV     SP,R0
015356  104415                                                          TRAP    C$PNTX
015360  062706  000010                                                  ADD     #10,SP
015364                  60$:    PASS                    ;RESTORE GPRS.
015364  004736                        JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015366  000207                RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - PUFIFO -
                        ;*************************************************************************
                        ;*      - PURGE THE FIFO
                        ;*      THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
                        ;*      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
                        ;*
                        ;* INPUTS:      RBUFA- CONTAINS THE  ADDRESS OF THE RECEIVER.
                        ;*
                        ;*
                        ;* OUTPUTS:     CARRY BIT - INDICATES THE STATE OF THE FIFO, SET:= PURGED.
                        ;*              BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PUFIFO
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
                        ;*************************************************************************

015370                  PUFIFO::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
015370  004567  166464                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015374  012701  001000          MOV     #512.,R1        ;SET MAXIMUM TRY COUNT OF 512.
015400  016704  164624          MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

015404  011402              2$: MOV     (R4),R2         ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
015406  100016              BPL     6$              ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
                        ;*
                        ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
                        ;  IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
                        ;-
015410  012700  070000          MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
015414  040200              BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
015416  001006              BNE     4$              ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                          ;*
                          ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                          ;-
015420  012700  000301          MOV     #301,R0         ; CHECK IF BMP.
015424  040200              BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
015426  001002              BNE     4$              ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
015430  004767  001044          JSR     PC,SAVBMP       ;SAVE BMP CODE ON THE QUEUE.

015434  005301              4$: DEC     R1              ;DECREMENT THE TRY COUNT.
015436  001362              BNE     2$              ;LOOP TO TRY AGAIN.
015440  000241                CLC                     ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
015442  000401              BR      60$             ;EXIT WITH CARRY CLEAR.
015444  000261              6$:     SEC             ;SET CARRY, TO INDICATE FIFO PURGED.

015446                      60$:    PASS                    ;RESTORE GPRS.
015446  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT, SET INDICATES FIFO PURGED.
015450  000207              RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - PUFIFR -
                        ;*************************************************************************
                        ;*      - PURGE FIFO REPORT ANY ERRORS FOUND.
                        ;*      THIS ROUTINE REMOVES ALL DATA FROM THE FIFO.  ANY BMP CODES THAT ARE
                        ;*      FOUND ARE SAVE ON THE QUEUE TO BE REPORTED LATER IN THE BMP REPORT TEST.
                        ;*      ANY UNXEPECTED DATA (IE ANY NON-STATUS INFORAMTION) THAT ARE FOUND,
                        ;*      ARE REPORTED AS AN ERROR.
                        ;*      IF THE FIFO WILL NOT PURGE AFTER 512 ATTEMPTS, THEN THE CURRENT TEST
                        ;*      THAT CALLED THIS ROUTINE RECEIVES A FAILURE FLAG THAT SHOULD BE USED
                        ;*      TO ABORT THE TEST.
                        ;*
                        ;* INPUTS:      ERRTBL - ERRTYPE, ERRMSG, ERRNBR ARE SET UP CORRECTLY.
                        ;*              RBUFA- CONTAINS THE  ADDRESS OF THE RECEIVER.
                        ;*
                        ;* OUTPUTS:     CARRY BIT - ABORT TEST FLAG, CLR = ABORT TEST, SET = OK.
                        ;*              ERRBLK - VALUE WILL BE DASTROYED.
                        ;*              BMPCQP - THE BMP CODE QUEUE POINTER MAY BE UPDATED.
                        ;*              THE CONTENTS OF THE BMP CODE QUEUE MAY BE UDATED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PUFIFR
                        ;*
                        ;* COMMENTS:    THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERRNBR
                        ;*              THRU TO ERRNBR+2.
                        ;*              THE ERRNBR IS RESTORED TO ITS INITIAL VALUE BEFORE RETURNING.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER1603,ER9001,ER9002,SAVBMP.
                        ;*************************************************************************

015452                  PUFIFR::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
015452  004567  166402                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015456  016746  166370          MOV     ERRNBR,-(SP)    ;SAVE THE CONTENTS OF THE ERROR NUMBER.
015462  012705  001000          MOV     #512.,R5        ;SET MAXIMUM READ COUNTER TO 2*FIFO SIZE.
                        ;*
                        ; READ DATA FROM THE FIFO UNTIL DATA VALID IS CLEAR OF READ COUNTER IS ZERO.
                        ;  REPORT ANY BMP OR UNEXPECTED DATA AS ERRORS.
                        ;-
015466  017702  164536  2$:       MOV     @RBUFA,R2       ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
015472  100063                  BPL     8$              ;EXIT IF DATA VALID CLEAR, IE. FIFO PURGED.
                        ;*
                        ; CHECK IF READ DATA IS STATUS OR UNEXPECTED CHARACTER.
                        ;-
015474  012700  070000          MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
015500  040200                  BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
015502  001012                  BNE     4$              ;SKIP BMP CHECK IF IT IS UNEXPECTED DATA.
                        ;*
                        ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                        ;  IF IT IS A BMP CODE THEN SAVE IT ON THE QUEUE.
                        ;-
015504  012767  012544  166344          MOV     #ER9001,ERRBLK  ;SET UP THE CORRECT ERROR REPORTING ROUTINE.
015512  012700  000300          MOV     #300,R0         ; CHECK IF BMP OR SELFTEST?.
015516  040200                  BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA
015520  001003                  BNE     4$              ;SKIP BMP ERROR REPORT IF MODEM OR SELFTEST?.
015522  004767  000752          JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
015526  000430                  BR      6$              ;BRANCH TO CHECK READ COUNT.
                        ;*
                        ; CHECK IF THE READ DATA IS MODEM, SELFTEST OR UNEXPECTED DATA.
                        ;-
```

```
3427 015530  032702  000001          4$:     BIT     #BIT0,R2        ;TEST THE MODEM STATUS INDICATION BIT.
3428 015534  001425                          BEQ     6$              ;DO NOT REPORT ANY ERROR IF MODEM STATUS.
3429 015536  012701  011124                  MOV     #EM9104,R1      ;PASS THE CORRECT ERROR MESSAGE TO REPORT.
3430 015542  010203                          MOV     R2,R3           ;EXTRACT THE LINE NUMBER FROM
3431 015544  000303                          SWAB    R3              ; THE READ DATA.
3432 015546  042703  177760                  BIC     #177760,R3      ;
3433 015552  006303                          ASL     R3              ;FORM LINE NUMBER TIMES 2 FOR ER9002 ROUTINE.
3434 015554  052704  100000                  BIS     #BIT15,R4       ;SET THE "NONE" EXPECTED MAESSAGE FLAG.
3435 015560  005267  166266                  INC     ERRNBR          ;SET ERROR NUMBER TO INTIAL ERRBR+1.
3436 015564  012767  012644  166264          MOV     #ER9002,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3437                                         ;REPORT ERROR "UNEXPECTED DATA FOUND IN FIFO".
3438 015572                                  ERROR                   ;               >>>>> ERROR <<<<<.
     015572  104460                                                                  TRAP    C$ERROR
3439                                 ;+
3440                                 ; EXIT WITH FAILURE IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
3441                                 ;-
3442 015574  032767  000100  164406          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
3443 015602  001415                          BEQ     7$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
3444                                                                 ;DURING THE SOFTWARE QUESTIONS.
3445
3446 015604  005367  166242                  DEC     ERRNBR          ;RESTORE ERROR NUMBER TO INTIAL ERRNBR.
3447
3448 015610  005305                  6$:     DEC     R5              ;DECREMENT READ COUNTER.
3449 015612  001325                          BNE     2$              ;LOOP TO READ NEXT CHAR FROM FIFO IF COUNT > 0.
3450                                 ;+
3451                                 ; THE FIFO WILL NOT CLEAR, REPORT THE ERROR AND INDICATE THAT THE TEST IS TO
3452                                 ; BE ABORTED.
3453                                 ;-
3454 015614  062767  000002  166230          ADD     #2,ERRNBR       ;SET ERROR NUMBER TO INTIAL ERRNBR+2.
3455 015622  012767  012020  166226          MOV     #ER1603,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3456 015630  012701  010767                  MOV     #EM9017,R1      ;PASS THE MESSAGE TO BE REPORTED.
3457                                         ;REPORT THE ERROR "FIFO WILL NOT PURGE, (DATA VALID STUCK SET)"
3458                                         ;                 "?????? TEST ABORTED".
3459 015634                                  ERROR                   ;               >>>>> ERROR <<<<<.
     015634  104460                                                                  TRAP    C$ERROR
3460 015636  000241                  7$:     CLC                     ;INDICATE THE TEST IS TO BE ABORTED.
3461 015640  000401                          BR      10$             ;EXIT THIS ROUTINE AND ABORT THE CURRENT TEST.
3462
3463 015642  000261                  8$:     SEC                     ;SET THE CARRY, DO NOT ABORT THE TEST.
3464
3465 015644  012667  166202          10$:    MOV     (SP)+,ERRNBR    ;RESTORE INITIAL ERROR NUMBER.
3466 015650                          60$:    PASS                    ;RESTORE GPRS.
     015650  004736                                      JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
3467                                                                 ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
3468                                                                 ; ABORT THE TEST.
3469 015652  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - READBX -
                        ;++ ************************************************************************
                        ;*      - READ CHARACTERS FROM THE FIFO AND CHECKS FOR BMPS AND XONS-
                        ;*      THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
                        ;*      IT READS THE SPECIFIED NUMBER OF CHARACTERS FROM THE FIFO AND CHECKS
                        ;*      FOR BMP CODES AND XON CHARACTERS.
                        ;*
                        ;* INPUTS:      R0 - CONTAINS THE NUMBER OF CHARS TO READ FROM THE FIFO.
                        ;*
                        ;* OUTPUTS:     R1 - CONTAINS ADDRESS OF ERROR MESSAGE TO BE REPORTED
                        ;*                   CLEAR IF NO ERROR FOUND.
                        ;*              CARRY USED TO INDICATE IF FIFO WAS FOUND EMPTY, CARRY CLEAR.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,READ
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CHKBMP.
                        ;-- ************************************************************************

015654                  READBX:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015654  004567  166200                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015660  005001          CLR     R1      ;CLEAR GPR THAT HOLDS THE ADDRESS OF ERRMSG.
015662  016703  164342  MOV     RBUFA,R3        ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
015666  011302          2$:     MOV     (R3),R2         ;READ A CHARACTER FROM THE FIFO.
015670  100015          BPL     8$              ;BRANCH IF FIFO IS EMPTY.
                        ;*
                        ; CHECK IF THE READ CHARACTER IS A BMP CODE.
                        ;  IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
                        ;  ABORT THE TEST.
                        ;-
015672  004767  176100  JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
015676  103410          BCS     6$              ;BRANCH IF A BMP CODE WAS FOUND.
015700  120227  000021  CMPB    R2,#21          ;CHECK IF IT IS AN XON.
015704  001003          BNE     4$              ;BRANCH IF NOT AN XON.
015706  012701  006520  MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
015712  000402          BR      6$              ;GO EXIT TEST.
015714  005300          4$:     DEC     R0              ;DECREMENT THE READ COUNT.
015716  001363          BNE     2$
015720  000261          6$:     SEC                     ;SET CARRY TO INDICATE SUCCESS.
015722  000401          BR      60$             ;EXIT
015724  000241          8$:     CLC                     ;CLEAR CARRY BIT TO INDICATE FAILURE.

015726                  60$:    PASS    R1              ;RESTORE GPRS.
015726  010166  000004                          MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
015732  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015734  000207          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - REPDER -
                        ;** ************************************************************************
                        ;*                        - REPORT DATA ERRORS -
                        ;*      THIS SUBROUTINE IS USED TO REPORT INCORRECT CHARACTERS AND LINE
                        ;*      NUMBERS IN A WORD OF DATA READ FROM THE RXFIFO.  THIS ROUTINE
                        ;*      CHECKS FOR THE NUMBER OF INDIVIDUAL DATA ERRORS ON A LINE EXCEEDING
                        ;*      THE REQUESTED AMOUNT AND SETS THE APPROAIATE ERROR SUMMARY FLAG, IT
                        ;*      THEN STOPS REPORTING ANY FURTHER ERRORS ON THAT LINE.  ANY BMP CODES
                        ;*      FOUND ARE SAVED ON THE BMP CODE QUEUE TO BE REPORTED LATER AND ANOTHER
                        ;*      CHARACTER IS READ FROM THE RXFIFO.
                        ;*
                        ;* INPUTS:      R1 - CONTAINS THE NUMBER OF THE UUT.
                        ;*              R3 - LOW BYTE CONTAINS THE EXPECTED CHAR.
                        ;*              ERCNTB - ADDRESS OF THE BASE OF THE ERROR SUMMARY TABLE.
                        ;*              ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE IN ERROR TABLE.
                        ;*              ERRNBR - SET TO THE ERROR NUMBER OF THE FIRST ERROR IN THIS ROUTINE.
                        ;*              EXOERR - ADDRESS OF "EXIT ON ERROR" FLAG.
                        ;*
                        ;* OUTPUTS:     CARRY - CLEAR IF RXFIFO WAS EMPTY, SET OTHERWISE.
                        ;*              ERCNTB - CONTENTS UPDATED TO REFLECT THE CURRENT STATE.
                        ;*              ERRBLK - CONTENTS MAY BE ALTERED.
                        ;*              ERSMRF - ERROR SUMMARY FLAGS UPDATED.
                        ;*              EXOERR - 1 IF AT LEAST ONE ERROR OCCURED.
                        ;*                       0 IF NO ERRORS OCCURED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,REPDER
                        ;*
                        ;* COMMENTS:    THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERRNBR
                        ;*              THRU INITIAL ERRNBR+1.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;-- ************************************************************************

015736                  REPDER:: SAVE
015736  004567  166116                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015742  005067  000304          CLR     62$             ;CLEAR THE "AN ERROR OCCURED" INDICATOR.
015746  012705  000200          MOV     #128.,R5        ;SET THE MAX BMP READ COUNT.
015752  000241          2$:     CLC                     ;CLEAR THE CARRY IN CASE THE FIFO IS EMPTY.
015754  017702  164250          MOV     @RBUFA,R2       ;READ THE RXFIFO.
015760  100132                  BPL     61$             ;EXIT THIS SUBROUTINE WITH CARRY CLEAR IF FIFO EMPTY.
                        ;*
                        ; CHECK IF THE CHARACTER IS A BMP CODE.
                        ;-
015762  012700  170301          MOV     #170301,R0      ;SET UP THE BIT MASK FOR A BMP CODE.
015766  040200          BIC     R2,R0           ;TRY AND CLEAR THE BIT MASK.
015770  001004          BNE     4$              ;AVOID SAVING THE CODE IF IT IS NOT A BMP CODE.
015772  004767  000502          JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
015776  005305          DEC     R5              ;DECREMENT THE MAX BMP READ COUNT.
016000  001364          BNE     2$              ;READ ANOTHER CHARACTER FROM THE RXFIFO.
                        ;*
                        ; VERIFY THAT THE READ CHARACTER IS CORRECT.
                        ;-
016002  010205          4$:     MOV     R2,R5           ;SAVE THE READ DATA.
016004  042702  177400          BIC     #177400,R2      ;CLEAR THE CLUTTER FROM THE DATA CHAR.
016010  120203          CMPB    R2,R3           ;COMPARE THE READ CHAR WITH THE EXPECTED CHAR.
016012  001432          BEQ     6$              ;AVOID THE ERROR REPORT IF THE DATA IS CORRECT.
016014  012767  000001  000230          MOV     #1,62$          ;INDICATE AN ERROR HAS OCCURED.
```

```
3572 016022  010104                          MOV     R1,R4           ;COPY THE LINE NUMBER.
3573 016024  006304                          ASL     R4              ;CONVERT THE LINE NUMBER TO AN OFFSET VALUE.
3574 016026  036467  002336  164556          BIT     BITTBL(R4),ERSMRF ;TEST WHETHER ERROR SUMMARY REPORTING IS IN
3575                                                                 ;EFFECT FOR THIS LINE.
3576 016034  001021                          BNE     6$              ;AVOID THE ERROR REPORT IF IT IS.
3577 016036  010104                          MOV     R1,R4           ;PASS THE LINE NUMBER TO THE ERROR ROUTINE.
3578 016040  110301                          MOVB    R3,R1           ;PASS THE EXPECTED DATA CHAR.
3579 016042  012703  010120                  MOV     #EM6004,R3      ;PASS THE ERROR MESSAGE,
3580                                                                 ; "TXFIFO BAD, RECIEVED CHAR INCORRECT, ON LINE XX ".
3581 016046  012767  012276  166002          MOV     #ER6001,ERRBLK  ;SET UP THE ERROR REPORTING ROUTINE.
3582 016054                                  ERROR                   ;REPORT THE ERROR.
     016054  104460                                                                      TRAP    C$ERROR
3583 016056  010401                          MOV     R4,R1           ;RESTORE THE LINE NUMBER.
3584 016060  032767  000100  164122          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
3585 016066  001004                          BNE     6$              ;AVOID THE EXITING IF IT HAS.
3586 016070  012767  000001  164154          MOV     #1,EXOERR       ;INDICATE THAT THE TEST IS TO BE EXITED.
3587 016076  000462                          BR      60$             ;EXIT THIS SUBROUTINE.
3588                                 ;+
3589                                 ; NOW CHECK THAT THE LINE NUMBER OF THE DATA READ FROM THE RXFIFO IS CORRECT.
3590                                 ;-
3591
3592 016100  010502                  6$:     MOV     R5,R2           ;RESTORE THE DATA READ FROM THE RXFIFO.
3593 016102  000302                          SWAB    R2              ;PLACE THE LINE NUMBER IN THE LOW BYTE.
3594 016104  042702  177760                  BIC     #177760,R2      ;CLEAR THE CLUTTER FROM THE LINE NUMBER.
3595 016110  020201                          CMP     R2,R1           ;COMPARE WITH THE EXPECTED LINE NUMBER.
3596 016112  001433                          BEQ     8$              ;AVOID THE ERROR REPORT IF THE NUMBER IS CORRECT.
3597 016114  012767  000001  000130          MOV     #1,62$          ;INDICATE THAT AT LEAST ONE ERROR HAS BEEN FOUND.
3598 016122  010104                          MOV     R1,R4           ;COPY THE LINE NUMBER.
3599 016124  006304                          ASL     R4              ;CONVERT THE LINE NUMBER TO AN OFFSET VALUE.
3600 016126  036467  002336  164456          BIT     BITTBL(R4),ERSMRF ;TEST WHETHER ERROR SUMMARY REPORTING IS IN
3601                                                                 ;EFFECT FOR THIS LINE.
3602 016134  001022                          BNE     8$              ;AVOID THE ERROR REPORT IF IT IS.
3603 016136  005267  165710                  INC     ERRNBR          ;SET THE ERROR NUMBER TO INITIAL+1.
3604 016142  012703  010165                  MOV     #EM6005,R3      ;PASS THE MESSAGE,
3605                                                                 ; "TXFIFO BAD, CHAR TRANSMITTED ON WRONG LINE"
3606 016146  012767  012170  165702          MOV     #ER5901,ERRBLK  ;SET UP THE ERROR REPORTING ROUTINE.
3607 016154                                  ERROR                   ;REPORT THE ERROR.
     016154  104460                                                                      TRAP    C$ERROR
3608 016156  005367  165670                  DEC     ERRNBR          ;SET THE ERROR NUMBER TO INITTAL.
3609 016162  032767  000100  164020          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3610 016170  001004                          BNE     8$              ;AVOID SETTING THE "EXIT ON ERROR" FLAG IF IT HAS.
3611 016172  012767  000001  164052          MOV     #1,EXOERR       ;SET THE FLAG.
3612 016200  000421                          BR      60$             ;EXIT THIS SUBROUTINE.
3613
3614                                 ;+
3615                                 ; UPDATE THE ERROR COUNT IN THE ERROR SUMMARY TABLE IF NECESSARY, AND UPDATE
3616                                 ; THE ERROR SUMMARY FLAGS IF NECESSARY.
3617                                 ;-
3618 016202  005767  164004          8$:     TST     NDERPT          ;TEST IF SUMMARY REPORTING HAS BEEN REQUESTED.
3619 016206  001416                          BEQ     60$             ;EXIT THIS SUBROUTINE IF IT HASN'T.
3620 016210  005767  000036                  TST     62$             ;TEST IF AN ERROR HAS BEEN FOUND.
3621 016214  001413                          BEQ     60$             ;EXIT THIS SUBROUTINE IF NO ERRORS.
3622 016216  010104                          MOV     R1,R4           ;COPY THE LINE NUMBER.
3623 016220  006304                          ASL     R4              ;CONVERT THE LINE NUMBER TO AN OFFSET VALUE.
3624 016222  005264  002614                  INC     ERCNTB(R4)      ;INCREMENT THE APPROPRIATE ERROR COUNT.
3625 016226  026764  163760  002614          CMP     NDERPT,ERCNTB(R4) ;COMPARE THE NUMBER OF ERRORS REPORTED WITH,
3626                                                                 ;THE NUMBER REQUESTED.
```

```
3627 016234  101003                          BHI     60$             ;AVOID SETTING THE ERROR SUMMARY FLAG IF THE
3628                                                                 ;NUMBER OF ERRORS REPORTED IS LESS THAN THE
3629                                                                 ;NUMBER REQUESTED.
3630 016236  056467  002336  164346          BIS     BITTBL(R4),ERSMRF ;OTHERWISE SET THE APPROIATE ERROR FLAG.
3631
3632 016244  000261                  60$:    SEC                     ;SET THE CARRY FLAG AND INDICATE SUCCESS.
3633 016246                          61$:    PASS
     016246  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3634 016250  000207                          RTS     PC
3635 016252  000000                  62$:    .WORD   0               ;STRORAGE FOR THE "AT LEAST ONE ERROR" INDICATOR.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - REPSMR -
                        ;** ************************************************************************
                        ;*                        - REPORT ERROR SUMMARY ROUTINE -
                        ;*      THIS SUBROUTINE REPORTS AN ERROR SUMMARY FOR THOSE LINES WHICH HAVE
                        ;*      EXCEEDED THE NUMBER OF INDIVIDUAL ERRORS TO REPORT FOR A SINGLE LINE
                        ;*      IN A SINGLE TEST.  THIS PARAMETER CAN BE SPECIFIED BY THE OPERATOR IF
                        ;*      HE/SHE ANSWERS THE SOFTWARE PARAMETER QUESTIONS.
                        ;*
                        ;* INPUTS:      ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
                        ;*              ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE.
                        ;*              ERRNBR - ERROR NUMBER OF ERRORS IN THIS ROUTINE.
                        ;*              ERSMRF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
                        ;*
                        ;* OUTPUTS:     ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE (DESTROYED).
                        ;*              SUMMARY MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,REPSMR
                        ;*
                        ;* COMMENTS:    IF NO LINES HAVE EXCEEDED THE MAXIMUM NUMBER OF INDIVIDUAL
                        ;*              ERRORS TO REPORT, NO MESSAGES ARE PRINTED BY THIS ROUTINE.
                        ;*              ERROR SUMMARIES IN THIS ROUTINE ARE REPORTED AS ERRORS.
                        ;*              THE CONTENTS OF ERRBLK ARE DESTROYED.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED:
                        ;-- ************************************************************************
016254                  REPSMR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016254  004567  165600                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016260  005767  164326          TST     ERSMRF                  ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
016264  001404          BEQ     60$                             ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                        ;+
                        ; WE HAVE SOME ERROR SUMMARIES TO REPORT.
                        ;-
016266  012767  013022  165562          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                            ;+
                            ; REPORT
                            ;     "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                            ;-
016274                          ERROR
016274  104460                                                  TRAP    C$ERROR
016276                  60$:    PASS                            ;RESTORE GPRS.
016276  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
016300  000207          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - RESETT -
                        ;**************************************************************************
                        ;*              - RESET DEVICE UNDER TEST -
                        ;*      THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
                        ;*      IF RESET DOES NOT SUCCESFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
                        ;*      AN ABORT TEST ERROR MESSAGE IS REPORTED.
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR
                        ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
                        ;*              ERRTBL - ERRTYP,ERNBR,AND ERRMSG SET UP CORRECTLY.
                        ;*
                        ;* OUTPUTS:     THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
                        ;*              CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
                        ;*              ERRBLK - VALUE MAY BE DESTROYED.
                        ;*              IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
                        ;*              TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RESETT
                        ;*
                        ;* COMMENTS:    THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
                        ;*              THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
                        ;**************************************************************************

016302                  RESETT:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
016302  004567  165552                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
016306  012702  000040          MOV     #BIT05,R2       ;SET BIT MASK OF MASTER RESET BIT.
                        ;*
                        ; TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
                        ;  IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
                        ;   IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
                        ;-
016312  016704  163710          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
016316  030214          BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
016320  001406          BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
016322  005003          CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
016324  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
016330  004767  176374          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
016334  103012          BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.

                        ;*
                        ; SET MASTER RESET BIT IN CSR.  CLEAR TX AND RX ENABLE BITS, ETC.
                        ;  SKIP THE SELFTEST.
                        ;  TIME-OUT OF 5 SECS, JUST IN CASE THE SELF-TEST EXECUTES.
                        ;-
016336  010277  163664  2$:     MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
016342  004767  000246          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                          ;*
                          ;   SET SELF-TEST TIME-OUT OF 5 SECONDS, AND WAIT FOR MR TO CLEAR.
                          ;   IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
                          ;   TEST INDICATOR.
                          ;-
016346  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
016350  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
016354  004767  176350          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
016360  103410          BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```
                                        ;+
                                        ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
                                        ; INDICATE TEST IS TO BE ABORTED BY CLEARTNG THE CARRY BIT.
                                        ;-
016362  012701  005320          4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
016366  012767  012020  165462          MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
                                        ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
                                        ; "TEST ABORTED"
016374                                  ERROR                   ;               >>>>> ERROR <<<<<
016374  104460                                                                    TRAP    C$ERROR
016376  000241                          CLC                     ;INDICATE TEST IS TO BE ABORTED.
016400  000403                          BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
                                        ;+
                                        ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
                                        ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
                                        ;-
016402  005067  163650          6$:     CLR     IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
016406  000261                          SEC                     ;INDICATE SUCCESS, CONTINUE TEST.

016410                          60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
016410  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
                                                                ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
016412  000207                          RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXIEO -
                                ;++ ************************************************************************
                                ;*                        - RECEIVER INTERRUPT DISABLE -
                                ;*      THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIEO
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
016414  010046                  RXIEO:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
016416                                  GETPRI -(SP)            ;SAVE PROCESSOR PRIORITY ON STACK.
016416  104440                                                                    TRAP    C$GPRI
016420  010046                                                                    MOV     R0,-(SP)
016422                                  SETPRI @PRI07           ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
016422  012700  000340                                                           MOV     @PRI07,R0
016426  104441                                                                    TRAP    C$SPRI
016430  042767  137777  163620        BIC     @137777,IESTAT  ;CLEAR RX.INT.ENBL BIT IN IESTAT.
016436  016777  163614  163562        MOV     IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
016444                                  SETPRI (SP)+            ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
016444  012600                                                                    MOV     (SP)+,R0
016446  104441                                                                    TRAP    C$SPRI
016450  012600                          MOV     (SP)+,R0        ;RESTORE R0.
016452  000207                      RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXIE1 -
                                ;++ ************************************************************************
                                ;*                          - RECEIVER INTERRUPT ENABLE -
                                ;*      THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIE1
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

016454  052767  000100  163574  RXIE1::  BIS     #BIT06,IESTAT   ;SET RX.INT.ENBL BIT IN IESTAT.
016462  042767  137677  163566           BIC     #137677,IESTAT  ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
016470  016777  163562  163530           MOV     IESTAT,@CSRA    ;ENABLE RX INTERRUPTS.
016476  000207                           RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - SAVBMP -
                        ;++ ************************************************************************
                        ;*                        - SAVE BMP CODES ROUTINE -
                        ;*      THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
                        ;*      TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
                        ;*
                        ;* INPUTS:      R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
                        ;*              BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
                        ;*              BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
                        ;*              BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
                        ;*              TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
                        ;*
                        ;* OUTPUTS:     BMPCQP -  INCREMENTED BY 4.
                        ;*              THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SAVBMP
                        ;*
                        ;* COMMENTS:    IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
                        ;*              OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

016500                  SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016500  004567  165354                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016504  016704  163700         MOV     BMPCQP,R4               ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
016510  116724  163540         MOVB    TSTNUM,(R4)+            ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
016514  005204                 INC     R4                      ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
016516  042702  177400         BIC     #177400,R2              ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
016522  010224                 MOV     R2,(R4)+                ;SAVE THE BMP CODE ON THE QUEUE.
016524  020427  002612         CMP     R4,#BMPCQE              ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
016530  103402                 BLO     2$                      ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
016532  162704  000004         SUB     #4,R4                   ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
016536  010467  163646         2$:     MOV     R4,BMPCQP       ;SAVE THE POINTER.

016542                  60$:    PASS                            ;RESTORE GPRS.
016542  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016544  000207                 RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - SETPAR
                        ;++ ************************************************************************
                        ;*                        - SET TX AND CONTROL PARAMETERS -
                        ;*      THIS SUROUTINE IS USED IN THE FIHAVL.TST.
                        ;*      IT INITIALISES THE SELECTED LINE TO THE FOLLOWING STATE:
                        ;*      INTERNAL LOOPBACK, IAUTO ENABLED, LPR:38.4K, 8 BITS/CHAR, 2 STOP,
                        ;*      ODD PARITY.
                        ;*
                        ;* INPUTS:      R1 - CONTAINS NUMBER OF THE LINE TO BE INITIALISED.
                        ;*
                        ;* OUTPUTS:     LNCTRL AND LPR REGISTERS FOR THE SELECTED LINE ARE DESTROYED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SETPAR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,WTWLNC,WTWLPR.
                        ;-- ************************************************************************

016546                  SETPAR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016546  004567  165306                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
016552  004767  176072          JSR     PC,LINBIT               ;GET A BIT MAP FOR THIS LINE.
016556  010005                   MOV     R0,R5                   ;COPY THE LINE BIT MAP.
016560  012700  000206           MOV     #206,R0                 ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
016564  004767  000762          JSR     PC,WTWLNC               ;INITILAISE THE LINE CONTROL REGISTER.
016570  012700  177670           MOV     #177670,R0              ;PASS THE LPR CONTENTS.
016574  004767  001002          JSR     PC,WTWLPR               ;SET THE LPR CONTENTS TO 38.4K BAUD.
016600  012704  000012           MOV     #10.,R4                 ;PASS DELAY TIME OF 10 MILLI SECONDS.
016604  004767  175452          JSR     PC,DELAY                ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

016610                  60$:    PASS                            ;RESTORE GPRS.
016610  004736                              JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
016612  000207                  RTS     PC
```

```
                                        .SBTTL  GLOBAL SUBROUTINE                - SKPSTS -
                                        ;** ************************************************************************
                                        ;*                          - SKIP SELFTEST ROUTINE -
                                        ;*      THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
                                        ;*      INITIATED.  IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
                                        ;*      RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
                                        ;*      CONSIDERATIONS).
                                        ;*
                                        ;* INPUTS:      CSRA - CONTAINS ADDRESS OF THE DUT CSR.
                                        ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
                                        ;*
                                        ;* OUTPUTS:     SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
                                        ;*
                                        ;* CALLING SEQUENCE:    JSR     PC,SKPSTS
                                        ;*
                                        ;* COMMENTS:
                                        ;*
                                        ;* SUBORDINATE ROUTINES CALLED: DELAY.
                                        ;-- ************************************************************************

016614                                  SKPSTS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016614  004567  165240                                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016620  012704  000012                          MOV     #10.,R4                 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
016624  004767  175432                         JSR     PC,DELAY                ;DELAY FOR 10 MILLI-SECONDS.
                                        ;*
                                        ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
                                        ;-
016630  012701  000060                          MOV     #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
                                        ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
                                        ;  LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHU-11.
016634  012703  052525                          MOV     #52525,R3               ;INITIALISE THE SKIP SELF-TEST CODE.
016640  005301                          4$:     DEC     R1                      ;SELECT THE NEXT SET OF DEVICE REGISTERS.
016642  016704  163360                          MOV     CSRA,R4                 ;GET THE ADDRESS OF THE CSR OF THE DUT.
016646  010124                          MOV     R1,(R4)+                ;SELECT A BANK OF DUT REGISTERS.
016650  010324                          6$:     MOV     R3,(R4)+                ;WRITE THE CODE TO A DUT REGISTER.
016652  020467  163366                          CMP     R4,TXBFCA               ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
016656  103774                         BLO     6$                      ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
016660  032701  000017                          BIT     #17,R1                  ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
016664  001365                         BNE     4$                      ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

016666                                  60$:    PASS                            ;RESTORE GPRS.
016666  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016670  000207                         RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TSABRT -
                        ;** ************************************************************************
                        ;*                        - TEST ABORT ROUTINE -
                        ;*      THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
                        ;*      DURING THE EXECUTION OF THE CURRENT TEST.
                        ;*      IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
                        ;*      ABORTED.
                        ;*
                        ;* INPUTS:      ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
                        ;*              ERRNBR - CONTAINS THE CORRECT ERROR NUMBER.
                        ;*              THE REMAINDER OF THE ERRTBL IS CORRECTLY INITIALISED.
                        ;*
                        ;* OUTPUTS:     MESSAGES ARE REPORTED TO THE OPERATOR.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TSABRT
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER1603.
                        ;-- ************************************************************************

016672                  TSABRT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
016672  004567  165162                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016676  012701  016714          MOV     #2$,R1          ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
016702  012767  012020  165146  MOV     #ER1603,ERRBLK  ;SET-UP THE ERROR REPORTING ROUTINE.
016710                          ERROR                   ;               >>>>> ERROR <<<<<.
016710  104460                                                                  TRAP    C$ERROR
016712  000432                  BR      60$             ;
016714     040  116  117  2$:     .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
016717     116  055  122
016722     105  114  101
016725     124  105  104
016730     040  124  105
016733     123  124  040
016736     105  122  122
016741     111  122  040
016744     106  117  125
016747     116  104  040
016752     104  125  122
016755     111  116  107
016760     040  124  105
016763     123  124  040
016766     105  130  105
016771     103  125  124
016774     111  117  116
016777     000
                            .EVEN
017000                  60$:    PASS                    ;RESTORE GPRS.
017000  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017002  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXDATP -
                        ;++ ************************************************************************
                        ;*                      - TRANSMIT DATA PATTERN -
                        ;*      THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
                        ;*      IT TRANSMITS A SPECIFIED NUMBER OF DATA BYTES ON THE SPECIFIED LINE.
                        ;*
                        ;* INPUTS:      R0 - CONTAINS THE NUMBER OF DATA BYTES TO TX.
                        ;*              R1 - CONTAINS LINE NUMB ON WHICH TRANSMISSION IS TO TAKE PLACE.
                        ;*              BUFBAS TO BUFMID CONTAINS A 256 BYTE DATA PATTERN.
                        ;*
                        ;* OUTPUTS:     DATA IS SENT OUT ON THE SPECIFIED LINE.
                        ;*              CARRY SET = TX SUCCESSFUL.
                        ;*
                        ;* CALLING SEQUENCE:    TXDATP
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DODMA.
                        ;-- ************************************************************************

017004                  TXDATP:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
017004  004567  165050                   JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017010  010003                   MOV     R0,R3           ;PASS THE NUMBER OF CHARS TO TX.
017012  012702  002650           MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
017016  004767  175356           JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
017022                  60$:    PASS                    ;RESTORE GPRS.
017022  004736                           JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017024  000207                   RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXDSBL -
                        ;++ ************************************************************************
                        ;*                      - TRANSMITTER DISABLE -
                        ;*      THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
                        ;*      CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
                        ;*
                        ;* INPUTS:      R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                        ;*              IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
                        ;*              NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
                        ;*              TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
                        ;*
                        ;* OUTPUTS:     R5 - BIT'S SET INDICATE THE INITIAL STATES OF ALL TX.ENBLE BITS.
                        ;*              TBUFFAD2 - THE STATE OF THE TX.ENBLE BIT MAY BE ALTERED.
                        ;*              THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXDSBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

017026                  TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017026  004567  165026                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017032  010500                  MOV     R5,R0           ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
017034  012701  000001          MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
017040  016702  163176          MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
017044  005202                  INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
017046  012703  000020          MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER PLUS ONE.
017052  016704  163200          MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
017056  005005                  CLR     R5              ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                        ;*
                        ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
                        ;-
017060  010477  163142  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
017064  105712                  TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
017066  100001                BPL     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
017070  050105                  BIS     R1,R5           ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
                        ;*
                        ;  CLEAR TX.ENBLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
                        ;  LINE BIT MAP.
                        ;-
017072  030100          4$:     BIT     R1,R0           ;CHECK STATE OF DISABLE LINE BIT MAP.
017074  001402                BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
017076  142712  000200          BICB    #BIT7,(R2)      ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
017102  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
017104  006301                  ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
017106  005303                  DEC     R3              ;DECREMENT LINE NUMBER.
017110  001363                BNE     2$              ;LOOP TO CHECK NEXT LINE.

017112                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
017112  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
017116  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
017120  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXENBL -
                        ;** ************************************************************************
                        ;*                        - TRANSMITTER ENABLE -
                        ;*      THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
                        ;*      SETTING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
                        ;*
                        ;* INPUTS:      R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                        ;*              IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
                        ;*              NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
                        ;*              TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
                        ;*
                        ;* OUTPUTS:     R5 - BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
                        ;*              TBUFFAD2 - THE STATE OF THE TX.ENBLE BIT MAY BE ALTERED.
                        ;*              THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXENBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

017122                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017122  004567  164732                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017126  010500                  MOV     R5,R0                   ;COPY BIT MAP OF LINES TO ENABLE.
017130  012701  000001          MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
017134  016702  163102          MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
017140  005202                  INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
017142  012703  000020          MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER.
017146  016704  163104          MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
017152  005005                  CLR     R5                      ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
                        ;*
                        ; SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENBLE BIT THAT IS CLEAR.
                        ;-
017154  010477  163046  2$:      MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
017160  105712                  TSTB    (R2)                    ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
017162  100401                BMI     4$                      ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
017164  050105                  BIS     R1,R5                   ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
                        ;*
                        ; SET TX.ENBLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
                        ; LINE BIT MAP.
                        ;-
017166  030100          4$:     BIT     R1,R0                   ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
017170  001402                BEQ     6$                      ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
017172  152712  000200          BISB    #BIT7,(R2)              ;ENABLE TRANSMISSION ON SELECTED LINE.
017176  005204          6$:     INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
017200  006301                  ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
017202  005303                  DEC     R3                      ;DECREMENT LINE NUMBER.
017204  001363                BNE     2$                      ;LOOP TO CHECK NEXT LINE.

017206                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
017206  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
017212  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                ; PREVIOUS LINES THAT WERE DISABLED.
017214  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIEO -
                                ;** ************************************************************************
                                ;*                        - TRANSMITTER INTERRUPT DISABLE -
                                ;*      THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIEO
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
017216  010046                  TXIEO:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
017220                                  GETPRI  -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
017220  104440                                                                  TRAP    C$GPRI
017222  010046                                                                  MOV     R0,-(SP)
017224                                  SETPRI  #PRI07          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
017224  012700  000340                                                          MOV     #PRI07,R0
017230  104441                                                                  TRAP    C$SPRI
017232  042767  177677  163016          BIC     #177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
017240  016777  163012  162760          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
017246                                  SETPRI  (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
017246  012600                                                                  MOV     (SP)+,R0
017250  104441                                                                  TRAP    C$SPRI
017252  012600                          MOV     (SP)+,R0        ;RESTORE R0.
017254  000207                          RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIE1 -
                                ;++ ************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT ENABLE -
                                ;*      THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE TX.INT.ENBL BIT IS SET IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE1
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

017256  052767  040000  162772  TXIE1::  BIS     #BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
017264  042767  137677  162764          BIC     #137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
017272  016777  162760  162726          MOV     IESTAT,@CSRA    ;ENABLE TX INTERRUPTS.
017300  000207                          RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - UNSDIV -
                                ;** ************************************************************************
                                ;*                      - UNSIGNED DIVIDE ROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
                                ;*      16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT.  ALL NUMBERS ARE
                                ;*      CONSIDERED TO BE UNSIGNED.  A SUCCESS FLAG IS NOT SET ON RETURN IF
                                ;*      THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
                                ;*
                                ;* INPUTS:      R1 - THE DIVISOR, UNSIGNED, 16 BITS.
                                ;*              R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
                                ;*              R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
                                ;*
                                ;* OUTPUTS:     R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
                                ;*              CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,UNSDIV
                                ;*
                                ;* COMMENTS:    IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
                                ;*              (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

017302                          UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017302  004567  164552                         JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
                                ;-
017306  010204                          MOV     R2,R4                   ;GET MSW OF DIVIDEND FOR SUBTRACT.
017310  160104                          SUB     R1,R4                   ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
017312  103403                          BCS     2$                      ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
017314  012701  177777                  MOV     #-1,R1                  ;SET QUOTIENT TO ALL ONES (177777),
017320  000442                          BR      60$                     ;EXIT WITH CARRY CLEAR.
                                ;+
                                ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
                                ;-
017322  005004                  2$:     CLR     R4                      ;CLEAR THE LSW OF THE DIVISOR.
017324  000241                          CLC                             ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
017326  006001                          ROR     R1                      ; DIVISOR BY
017330  006004                          ROR     R4                      ; 2(UNSIGNED)
017332  012700  000020                  MOV     #16.,R0                 ;SET UP INITIAL SHIFT COUNT TO 16.
                                ;+
                                ; THE SUBTRACT AND SHIFT LOOP.
                                ;-
017336  010246                  4$:     MOV     R2,-(SP)                ;SAVE MSWORD OF DIVIDEND.
017340  010346                          MOV     R3,-(SP)                ;SAVE LSWORD OF DIVIDEND.
017342  160403                          SUB     R4,R3                   ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
017344  005602                          SBC     R2                      ;MSWORD DIVIDEND - BORROW .
017346  103402                          BCS     6$                      ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
017350  160102                          SUB     R1,R2                   ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
017352  103003                          BCC     8$                      ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
                                    ;+
                                    ; IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
                                    ; CARRY IS SET.
                                    ;-
017354  012603                  6$:     MOV     (SP)+,R3                ;RESTORE LSWORD OF DIVIDEND.
017356  012602                          MOV     (SP)+,R2                ;RESTORE MSWORD OF DIVIDEND.
```

```
4194 017360  000401                        BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
4195                                    ;+
4196                                    ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
4197                                    ; COMPLEMENTED LATER).   CARRY IS CLEAR.
4198                                    ;-
4199 017362  012626                 8$:    MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
4200                                    ;+
4201                                    ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
4202                                    ;-
4203 017364  006105                 10$:   ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
4204 017366  000241                        CLC                     ;DIVIDE THE
4205 017370  006001                        ROR     R1              ;  DEVISOR BY
4206 017372  006004                        ROR     R4              ;  2 (UNSIGNED).
4207 017374  005300                        DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
4208 017376  001357                        BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
4209 017400  005105                        COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
4210                                    ;+
4211                                    ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
4212                                    ;-
4213 017402  000241                        CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
4214 017404  006103                        ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
4215 017406  103402                        BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
4216 017410  160403                        SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
4217 017412  103403                        BCS     14$             ;IF BORROW, DON'T ROUND UP.
4218                                    ;+
4219                                    ; ROUND UP, EXTRA SUBTRACT WENT.
4220                                    ;-
4221 017414  005205                 12$:   INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
4222 017416  001001                        BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
4223 017420  005305                        DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
4224                                    ;+
4225                                    ; ALL DONE, PASS QUOTIENT AND EXIT.
4226                                    ;-
4227 017422  010501                 14$:   MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
4228 017424  000261                        SEC                     ;INDICATE NO OVERFLOW.
4229
4230 017426                         60$:   PASS    R1                      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
     017426  010166  000004                        MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     017432  004736                        JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
4231                                                                    ;R1 - 16 BIT, UNSIGNED QUOTIENT.
4232 017434  000207                        RTS     PC                      ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WAIBIS -
                        ;++ ************************************************************************
                        ;*                        - WAIT FOR BIT SET ROUTINE -
                        ;*      THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET.  IF THE
                        ;*      SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME-OUT
                        ;*      PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
                        ;*      THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
                        ;*      ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
                        ;*
                        ;* INPUTS:      R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
                        ;*                BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
                        ;*                BITS 11 THRU 0  - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
                        ;*              R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
                        ;*                   MSLCNT.
                        ;*
                        ;* OUTPUTS:     R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
                        ;*              CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME-OUT).
                        ;*
                        ;* CALLING SEQUENCE:    MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
                        ;*                                               ;  32 (40 OCTAL) MS DELAY.
                        ;*                       MOV     #LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
                        ;*                       JSR     PC,WAIBIS       ;WAIT 32 MS FOR BIT 11 TO SET.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                        ;-- ************************************************************************
                        
017436                  WAIBIS:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
017436  004567  164416                   JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017442  010204                   MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
017444  010102                   MOV     R1,R2
017446  042701  170000           BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
017452  042702  007777           BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
017456  000302                   SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
017460  006202                   ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
017462  006202                   ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
017464  006202                   ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
017466  016202  002336           MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
017472  010203                   MOV     R2,R3                   ;INDICATE THAT THE BIT SHOULD BE SET.
017474  004767  175230           JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                 ;  CARRY IS CORRECT UPON MSLGET RETURN.
017500  010002                   MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
017502                  60$:     PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
017502  010266  000006                           MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
017506  004736                           JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                 ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
017510  000207                   RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WAITTX -
                        ;++ ************************************************************************
                        ;*                        - WAIT FOR TX TO FINISH -
                        ;*       THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
                        ;*      IT WAITS FOR TRANSMISSION TO COMPLETE IE TX_ACTION. THEN DELAYS
                        ;*      FOR 5 MILLISECONDS TO ALLOW TIME FOR THE LAST CHARACTER TO GET INTO
                        ;*      THE FIFO.
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
                        ;*
                        ;* OUTPUTS:     CARRY - SET INDICATES SUCCESS.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WAITTX
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,WAIBIS.
                        ;-- ************************************************************************

017512                  WAITTX:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
017512  004567  164342                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017516  012701  170536          MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
017522  016702  162500          MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
017526  004767  177704          JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
017532  103005                  BCC     60$             ;BRANCH IF NO TX_ACTION, ABORT THE TEST.
017534  012704  000005          MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
017540  004767  174516          JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
017544  000261                  SEC                     ;SET CARRY TO INDICATE SUCCESS.

017546                  60$:    PASS                    ;RESTORE GPRS.
017546  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
                                                        ;PASS THE CARRY BIT, SET INDICATES SUCCESS.
017550  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - WTWLNC
                                ;** ************************************************************************
                                ;*                        - LINE CONTROL REGISTER SETUP ROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
                                ;*      CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE.  ONLY THE LNCTRLS
                                ;*      FOR THE SPECIFIED LINES ARE ALTERED.
                                ;*
                                ;* INPUTS:      R0 - NEW LINE PARAMETERS.
                                ;*              R5 - BIT MAP OF LINES TO BE ALTERED.
                                ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
                                ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS IN THE CSR.
                                ;*              LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
                                ;*
                                ;* OUTPUTS:     LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,WTWLNC
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                                ;-- ************************************************************************

017552                          WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017552  004567  164302                                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;*
                                ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                                ;*
                                        MOV     LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
                                        MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
                                        MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
                                        MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                                ;*
                                ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                                ;*
                                        JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

                                60$:    PASS                            ;RESTORE GPRS.
                                                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                        RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WTWLPR -
                        ;** ************************************************************************
                        ;*                      - LINE PARAMETER REGISTER SETUP ROUTINE -
                        ;*      THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
                        ;*      PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE.  ONLY THE LPRS FOR
                        ;*      THE SPECIFIED LINES ARE ALTERED.
                        ;*
                        ;* INPUTS:      R0 - NEW LINE PARAMETERS.
                        ;*              R5 - BIT MAP OF LINES TO BE ALTERED.
                        ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
                        ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
                        ;*               ENABLE BITS IN THE CSR.
                        ;*              LPRA - CONTAINS ADDRESS OF THE DUT LPR.
                        ;*
                        ;* OUTPUTS:     LPR - SPECIFIED DUT LINE PARAMTER REGISTERS ARE ALTERED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLPR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ;-- ************************************************************************

017602                  WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017602  004567  164252                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
017606  016701  162420          MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
017612  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
017614  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
017616  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;*
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
017622  004767  173542          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

017626                  60$:    PASS                            ;RESTORE GPRS.
017626  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017630  000207                  RTS     PC
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINE       - CLKINT
                                ;** ************************************************************************
                                ;*      THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND.  IT DECREMENTS THE
                                ;*      TWO TIMER COUNTERS DOWN TO ZERO.
                                ;*
                                ;* INPUTS:      TIMER1 - TIMER COUNTER #1.
                                ;*              TIMER2 - TIMER COUNTER #2.
                                ;*              TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
                                ;*
                                ;* OUTPUTS:     THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
                                ;*
                                ;* CALLING SEQUENCE:    PUT @CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
                                ;*                      PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
                                ;*                      EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
                                ;*                      COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
                                ;*
                                ;* COMMENTS:    THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0.  THIS
                                ;*              ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME-OUT
                                ;*              HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
017632  005767  162456       CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
017636  001402                       BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
017640  005367  162450               DEC     TIMER1          ;DECREMENT TIME COUNT.
017644  005767  162446         2$:   TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
017650  001402                       BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
017652  005367  162440               DEC     TIMER2          ;DECREMENT TIME COUNT.
017656  005367  162436         4$:   DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
017662  001006                       BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
017664  016767  162432  162426       MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
017672  010046                       MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
017674                               BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
017674  104422                                                               TRAP    C$BRK
017676  012600                       MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
017700  000002                 60$:  RTI
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINE       - RXDECT -
                                ;++ ************************************************************************
                                ;*                          - RX INT DECTION ROUTINE -
                                ;*      THIS ROUTINE DETECTS AN RX INTERRUPT BY SETTING THE RXINTC WORD TO 1.
                                ;*      THIS ROUTINE IS USED IN THE RXTIMER TESTS.
                                ;*
                                ;* INPUTS:      RXINTC - STORGE FOR THE INTERRUPT COUNT.
                                ;*
                                ;* OUTPUTS:     RXINTC - SET TO 1.
                                ;*
                                ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL RXDECT IN THE VECTOR
                                ;*                      LOCATION.
                                ;*
                                ;* COMMENTS:    THIS ROUTINE DOES NOT READ THE RXFIFO.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

017702  012767  000001  162356  RXDECT:: MOV    #1,RXINTC       ;INDICATE THAT AN RX-INT HAS OCCURED.
017710  000002                          RTI
```

```
                                .SBTTL  GLOBAL TRAP SERVICE ROUTINE     - TP4RTN -
                                ;*************************************************************************
                                ;*      BUS TIME-OUT TRAP (004 TRAP) SERVICE ROUTINE -
                                ;*      THIS ROUTINE DETERMINES IF THE 004 TRAP WAS CAUSED BY
                                ;*      AN "EXPECTED" ERROR OR NOT BY EXAMINING THE RETURN PC VALUE ON THE
                                ;*      STACK. IF THE TRAP IS UNEXPECTED, THIS ROUTINE JUMPS TO THE NORMAL
                                ;*      DIAGNOSTIC SUPERVISOR 004 TRAP HANDLING ROUTINE.
                                ;*
                                ;*
                                ;* INPUTS:      SP - POINTS TO THE PC WHERE THE TRAP OCCURED.
                                ;*              ADRPTR - LABEL AT THE ADDRESS WHERE "EXPECTED" TRAPS OCCUR.
                                ;*              TP4FLG - 004 TRAP FLAGS.
                                ;*
                                ;* OUTPUTS:     TP4FLG - BIT 15 IS SET IF "EXPECTED" TRAP OCCURED.
                                ;*
                                ;* CALLING SEQUENCE:    PUT ADDRESS POINTED TO BY TP4RTN IN 004 VECTOR.
                                ;*                      OCCURENCE OF 004 TRAP VECTORS TO THIS ROUTINE.
                                ;*
                                ;* COMMENTS:    ANY 004 TRAP WHICH OCCURS AT AN ADDRESS OTHER THAN THAT LABELED
                                ;*              ADRPTR WILL BE HANDLED BY THE NORMAL 004 TRAP SERVICE ROUTINE.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;*************************************************************************

017712  021627  014200        TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
017716  001402                         BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
017720  000177  162352                 JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
017724  052767  100000  162346  2$:    BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
017732  000002                         RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

```
                        .SBTTL  INTERRUPT SERVICE ROUTINE       - TXAINT -
                        ;** ****************************************** ********************************
                        ;*                       - TRANSMIT ACTION INTERRUPT SERVICE ROUTINE -
                        ;*      THIS ROUTINE HANDLES A TX INTERRUPT BY COUNTING THE INTERRUPT,
                        ;*      SETTING A FLAG IF THE TX_ACTION BIT IS CLEAR, AND READING THE CSR
                        ;*      UNTIL THE TX_ACTION BIT CLEARS OR THE MAXIMUM READ COUNT IS EXCEEDED.
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
                        ;*              TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS.
                        ;*              TXINTF - TX INTERRUPT FLAGS.
                        ;*
                        ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED TX INTERRUPT COUNT.
                        ;*              TXINTF - TX INTERRUPT FLAGS (BIT 15 SET IF TX_ACTION CLEAR
                        ;*                                          BIT 14 SET IF MAX READ COUNT EXCEEDED)
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL TXAINT IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;*
                        ;-- ****************************************************************************

                        TXAINT:: SAVE
                                                        JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
                                                        MOV     TXINTC,R1       ;GET THE TX-INT COUNT.
                                                        INC     R1              ;INCREMENT THE COUNT.
                                                        BVC     2$              ;BRANCH IF NO OVERFLOW OCCURED.
                                                        DEC     R1              ;RESET THE COUNT TO 177777.
                                2$:                     MOV     R1,TXINTC       ;SAVE THE NEW COUNT.
                                                        MOV     TXINTF,R3       ;GET THE TX-INT FLAGS.
                                                        TST     @CSRA           ;READ THE CSR.
                                                        BMI     4$              ;AVOID SETTING THE ERROR FLAG IF
                                                                                ;THERE IS A TX_ACTION.
                                                        BIS     #BIT15,R3       ;SET THE FLAG.
                                4$:                     MOV     R3,TXINTF       ;UPDATE THE TX-INT FLAGS.
                                                        MOV     #32.,R2         ;SET THE MAX TX_ACTION READ COUNT.
                                                        ;*
                                                        ; READ THE CSR UNTIL THE TX_ACTION FIFO IS EMPTY OR THE MAX READ COUNT
                                                        ; IS EXCEEDED.
                                                        ;-
                                6$:                     TST     @CSRA           ;READ THE CSR.
                                                        BPL     60$             ;EXIT IF TX_ACTION FIFO IS EMPTY.
                                                        DEC     R2              ;DECREMENT THE MAX READ COUNT.
                                                        BNE     6$              ;BRANCH TO READ ANOTHER TX_ACTION IF MAX READ
                                                                                ;COUNT IS NOT EXCEEDED.
                                                        BIS     #BIT14,TXINTF   ;SET THE "MAX TX_ACTION COUNT EXCEEDED" FLAG.
                                60$:                    PASS
                                                                        JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                        RTI
```

```
4536
4537
4538                                    .SBTTL  REPORT CODING SECTION
4539
4540                                    ;++
4541                                    ; THE REPORT CODING SECTION CONTAINS THE
4542                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4543                                    ;--
4544
4545 020030                                     BGNRPT
     020030                                                                L$RPT::
4546
4547 020030                                     EXIT    RPT
     020030  000167                                                                .WORD   J$JMP
     020032  000000                                                                .WORD   L10017-2-.
4548
4549                                    .EVEN
4550
4551 020034                                     ENDRPT
     020034                                                                L10017:
     020034  104425                                                                TRAP    C$RPT
```

PROTECTION TABLE

```
                                .SBTTL  PROTECTION TABLE


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

020036                                  BGNPROT
020036                                                                  L$PROT::

020036  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
020040  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
020042  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

020044                                  ENDPROT
```

```
4597
4598
4599                                    .SBTTL  INITIALIZE SECTION
4600                                    ;++
4601                                    ;*****************************************************************************
4602                                    ;*      THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
4603                                    ;*      EACH PASS OR AFTER A CONTINUE COMMAND.
4604                                    ;*      THIS CODE PERFORMS THE FOLLOWING ACTIONS:
4605                                    ;*
4606                                    ;*      MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
4607                                    ;*      DATA AREA.
4608                                    ;*
4609                                    ;*****************************************************************************
4610                                    ;--
4611 020044                                     BGNINIT
     020044                                                                                     L$INIT::
4612                                    ;SEE IF PROGRAM JUST STARTED, BR IF YES
4613 020044                                     READEF  #EF.START
     020044  012700  000040                                                                     MOV     #EF.START,R0
     020050  104447                                                                             TRAP    C$REFG
4614 020052                                     BCOMPLETE       NEWSTA
     020052  103416                                                                             BCS     NEWSTA
4615                                    ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
4616 020054                                     READEF  #EF.RESTART
     020054  012700  000037                                                                     MOV     #EF.RESTART,R0
     020060  104447                                                                             TRAP    C$REFG
4617 020062                                     BCOMPLETE       NEWRES
     020062  103556                                                                             BCS     NEWRES
4618                                    ;SEE IF THIS IS A NEW PASS, BR IF YES
4619 020064                                     READEF  #EF.NEW
     020064  012700  000035                                                                     MOV     #EF.NEW,R0
     020070  104447                                                                             TRAP    C$REFG
4620 020072                                     BCOMPLETE       NEWPAS
     020072  103555                                                                             BCS     NEWPAS
4621                                    ;SEE IF PROGRAM WAS JUST CONTINUED
4622 020074                                     READEF  #EF.CONTINUE
     020074  012700  000036                                                                     MOV     #EF.CONTINUE,R0
     020100  104447                                                                             TRAP    C$REFG
4623 020102                                     BNCOMPLETE      GETPRM
     020102  103161                                                                             BCC     GETPRM
4624 020104  000167  000540                    JMP     ENDIT
4625 020110                             NEWSTA:
4626 020110                                     BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     020110  104433                                                                             TRAP    C$RESET
4627                                    ;+
4628                                    ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
4629                                    ;-
4630 020112                                     CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
     020112  012700  000114                                                                     MOV     #L,R0
     020116  104462                                                                             TRAP    C$CLCK
     020120  010001                                                                             MOV     R0,R1
4631 020122  012167  162156                     MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
4632 020126  012167  162154                     MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
4633 020132  012167  162152                     MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
4634 020136  012167  162150                     MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
4635 020142  026727  162144  000062             CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
4636 020150  001004                             BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
```

INITIALIZE SECTION

```
4637 020152  012767  000024  162144          MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
4638 020160  000403                          BR      4$
4639 020162  012767  000021  162134  2$:     MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
4640 020170                        4$:     SETVEC  CLKVEC,#CLKINT,#PRI06   ;INITIALIZE CLOCK INTERRUPT VECTOR.
     020170  012746  000300                                                          MOV     #PRI06,-(SP)
     020174  012746  017632                                                          MOV     #CLKINT,-(SP)
     020200  016746  162104                                                          MOV     CLKVEC,-(SP)
     020204  012746  000003                                                          MOV     #3,-(SP)
     020210  104437                                                                  TRAP    C$SVEC
     020212  062706  000010                                                          ADD     #10,SP
4641 020216  016700  162070                  MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
4642 020222  006300                          ASL     R0                      ; TO CAUSE A BREAK
4643 020224  010067  162072                  MOV     R0,BCOUNT               ; EVERY 2 SECONDS.
4644 020230                                  SETPRI  #PRI05                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
     020230  012700  000240                                                          MOV     #PRI05,R0
     020234  104441                                                                  TRAP    C$SPRI
4645                                 ;+
4646                                 ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4647                                 ; IS ACCESSABLE.
4648                                 ;    FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4649                                 ;-
4650 020236  016767  157542  162032          MOV     4,TP4VEC                ;SAVE THE EXISTING 004 TRAP VECTOR.
4651 020244  012767  017712  157532          MOV     #TP4RTN,4               ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4652                                 ;+
4653                                 ;    ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4654                                 ;-
4655 020252  005067  162022                  CLR     TP4FLG                  ;CLEAR THE 004 TRAP FLAG.
4656 020256  012767  000100  162016          MOV     #BIT6,WORD1             ;SET UP TO SET BIT6 OF THE LTC CSR.
4657 020264  012700  002302                  MOV     #WORD1,R0               ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4658 020270  016701  162010                  MOV     CLKCSR,R1               ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4659 020274  004767  173666                  JSR     PC,CKTRAP               ;MOVE AND CHECK FOR TRAP.
4660 020300  016767  161772  157476          MOV     TP4VEC,4                ;RESTORE THE NORMAL 004 TRAP VECTOR.
4661 020306  103403                          BCS     6$                      ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4662 020310  005067  161776                  CLR     CLKHRZ                  ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4663 020314  000402                          BR      8$                      ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4664                                 ;+
4665                                 ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4666                                 ;-
4667 020316  004767  173230          6$:     JSR     PC,CALMSL
4668                                 ;+
4669                                 ; CHECK FOR MEMMORY MANAGEMENT PRESENT ON THIS MACHINE.
4670                                 ; IF MEM MGT IS PRESENT, DISABLE IT.
4671                                 ;-
4672 020322  016767  157456  161746  8$:     MOV     4,TP4VEC                ;SAVE THE EXISTING 004 TRAP VECTOR.
4673 020330  012767  017712  157446          MOV     #TP4RTN,4               ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4674 020336  005067  161736                  CLR     TP4FLG                  ;CLEAR THE 004 TRAP FLAG.
4675 020342  005067  161734                  CLR     WORD1                   ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
4676 020346  012700  002302                  MOV     #WORD1,R0               ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4677 020352  016701  161752                  MOV     MMSR0,R1                ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
4678 020356  005067  161750                  CLR     MMPRES                  ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
4679 020362  005067  161746                  CLR     MMENAB                  ;INDICATE MEM MGT IS NOT ENABLED.
4680 020366  004767  173574                  JSR     PC,CKTRAP               ;CLEAR THE MEM MGT SR0 REG AND CHECK FOR TRAP.
4681 020372  016767  161700  157404          MOV     TP4VEC,4                ;RESTORE THE NORMAL 004 TRAP VECTOR.
4682 020400  103003                          BCC     10$                     ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4683 020402  012767  000001  161722          MOV     #1,MMPRES               ;INDICATE THAT MEM MGT IS PRESENT.
4684 020410  005067  161650          10$:    CLR     PASCNT                  ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4685 020414  000167  000006                  JMP     NEWPAS                  ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
```

```
4686
4687 020420                          NEWRES: BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     020420  104433                                                                  TRAP    C$RESET
4688 020422  005067  161636                  CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4689 020426                          NEWPAS:
4690 020426  012767  177777  161570          MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
4691                                 ;+
4692                                 ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4693                                 ;  THIS COUNTER IS USED IN THE ROM VERSION TEST.
4694                                 ;-
4695 020434  005267  161624                   INC    PASCNT          ;INCREMENT THE PASS COUNTER.
4696 020440  001002                          BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4697 020442  005367  161616                   DEC    PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4698
4699                                 ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4700 020446                          GETPRM:
4701 020446  005267  161552                  INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
4702 020452  026767  161546  161332          CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4703 020460  002362                          BGE     NEWPAS          ;BR IF YES
4704
4705 020462                                  GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
     020462  016700  161536                                                          MOV     UNITN,R0
     020466  104442                                                                  TRAP    C$GPHRD
     020470  010001                                                                  MOV     R0,R1
4706 020472                                  BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
     020472  103401                                                                  BCS     30$
4707 020474  000764                          BR      GETPRM          ;SKIP THIS DEVICE
4708
4709
4710                                         ;*********** HARDWARE PARAMETER MOVING CODE ****************
4711 020476  012167  161524          30$:    MOV     (R1)+,CSRA      ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4712 020502  012102                          MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
4713 020504  010267  161504                  MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
4714 020510  062702  000004                  ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4715 020514  010267  161476                  MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
4716 020520  012167  161474                  MOV     (R1)+,ACTLNS    ;STORE DHU-11 ACTIVE LINE BIT MAP
4717 020524  111167  161472                  MOVB    (R1),LOPBCK     ;STORE DHU-11 LOOPBACK MODE
4718                                 ;+
4719                                 ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4720                                 ;       DEVICE REGISTER ADDRESS TABLE.
4721                                 ;-
4722 020530  016701  161472                  MOV     CSRA,R1         ;COPY CSR ADDRESS
4723 020534  005201                          INC     R1              ;INCREMENT CSR ADDRESS
4724 020536  005201                          INC     R1              ; COPY BY 2.
4725 020540  012703  000007                  MOV     #7,R3           ;SET UP REGISTER COUNT
4726 020544  012702  002230                  MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4727 020550  010122                  12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
4728 020552  005201                          INC     R1              ;INCREMENT REGISTER ADDRESS
4729 020554  005201                          INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
4730 020556  005303                          DEC     R3              ;DECREMENT REGISTER COUNT
4731 020560  001373                          BNE     12$             ;LOOP IF NOT DONE
4732
4733                                 ;+
4734                                 ; INITIALISE THE BMP CODE QUEUE.
4735                                 ;-
4736 020562  012700  002412                   MOV    #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
4737 020566  012701  002612                   MOV    #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
```

```
4738 020572  010067  161612                MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
4739 020576  005020                 14$:    CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4740 020600  020001                         CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4741 020602  103775                        BLO     14$             ;LOOP IF NOT ALL DONE.
4742                                ;+
4743                                ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES.
4744                                ;  AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4745                                ;-
4746 020604  032767  000020  161376          BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
4747 020612  001416                        BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4748 020614  026727  161172  000001          CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4749 020622  003412                        BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4750 020624                                 PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
     020624  016746  161374                                                        MOV     UNITN,-(SP)
     020630  012746  005231                                                        MOV     #MFUNIT,-(SP)
     020634  012746  000002                                                        MOV     #2,-(SP)
     020640  010600                                                                MOV     SP,R0
     020642  104417                                                                TRAP    C$PNTF
     020644  062706  000006                                                        ADD     #6,SP
4751 020650                         16$:
4752
4753 020650  005067  161374         ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
4754                                ;+
4755                                ; SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.
4756                                ;-
4757 020654                                 SETPRI  #PRI07                  ;SET PROCESSOR PRIORITY TO 7.
     020654  012700  000340                                                        MOV     #PRIC7,R0
     020660  104441                                                                TRAP    C$SPRI
4758 020662                                 ENDINIT
     020662                                                             L10021:
     020662  104411                                                                TRAP    C$INIT
4759
4760            000000                      TNUM == 0               ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

INITIALIZE SECTION

```
                                .SBTTL  AUTODROP SECTION


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

020664                                  BGNAUTO
020664                                                                  L$AUTO::


020664                                  ENDAUTO
020664                                                                  L10022:
020664  104461                                                                  TRAP    C$AUTO
```

```
4799
4800
4801                                    .SBTTL  CLEANUP CODING SECTION
4802
4803                                    ;**
4804                                    ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4805                                    ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4806                                    ;--
4807
4808 020666                                     BGNCLN
     020666                                                                          L$CLEAN::
4809
4810 020666  005767  161356                       TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
4811 020672  001401                               BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
4812 020674                                       BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
     020674  104433                                                                  TRAP    C$RESET
4813 020676                                     2$:
4814
4823
4824 020676                                     EXIT    CLN
     020676  104432                                                                  TRAP    C$EXIT
     020700  000002                                                                  .WORD   L10023-.
4825
4837
4838                                            .EVEN
4839
4840 020702                                     ENDCLN
     020702                                                                  L10023:
     020702  104412                                                                  TRAP    C$CLEAN
```

```
                                ;SBTTL  DROP UNIT SECTION

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

020704                                  BGNDU
020704                                                                  L$DU::
                                 PRINTF #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
020704  010046                                                          MOV     R0,-(SP)
020706  012746  020730                                                  MOV     #DROP,-(SP)
020712  012746  000002                                                  MOV     #2,-(SP)
020716  010600                                                          MOV     SP,R0
020720  104417                                                          TRAP    C$PNTF
020722  062706  000006                                                  ADD     #6,SP
020726  000427                  BR      EDROP           ;BRANCH AROUND THE MESSAGE.

020730      045     101     040 DROP:   .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
020733      125     116     111
020736      124     045     104
020741      066     045     101
020744      040     104     122
020747      117     120     120
020752      105     104     040
020755      106     122     117
020760      115     040     106
020763      125     122     124
020766      110     105     122
020771      040     124     105
020774      123     124     111
020777      116     107     056
021002      045     116     000

                                .EVEN
021006                  EDROP:

021006                          EXIT    DU
021006  000167                                                          .WORD   J$JMP
021010  000000                                                          .WORD   L10024-2-.

021012                          ENDDU
021012                                                          L10024:
021012  104453                                                          TRAP    C$DU
```

DROP UNIT SECTION

```
4879
4880
4881                                    .SBTTL  ADD UNIT SECTION
4882
4883                                    ;**
4884                                    ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4885                                    ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4886                                    ;  TO THE TEST CYCLE.
4887                                    ;--
4888
4889 021014                                    BGNAU
     021014                                                                  L$AU::
4890
4891 021014                                    EXIT    AU
     021014  000167                                                                 .WORD   J$JMP
     021016  000000                                                                 .WORD   L10025-2-.
4892
4893
4894                                    .EVEN
4895
4896 021020                                    ENDAU
     021020                                                                  L10025:
     021020  104452                                                                TRAP    C$AU
4897
```

```
                                .SBTTL  HARDWARE TEST           - ADRA -
                                ;++
                                ;*************************************************************************
                                ;*                        - REGISTER ADDRESS TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DEVICE REGISTERS WILL RESPOND TO THE PROPER
                                ;*      UNIBUS HANDSHAKING SIGNALS WHEN ACCESSED. IF THE DHU11 DOES NOT RESPOND
                                ;*      TO THE ACCESS ATTEMPTS (IF THE DHU11 IS AT THE WRONG ADDRESS, FOR EXAMPLE)
                                ;*      THE 004 BUS TIME-OUT TRAP IS DETECTED BY THIS ROUTINE AND AN ERROR
                                ;*      IS REPORTED. THIS TEST IS PERFORMED ON LINE 0 ONLY.
                                ;*
                                ;*************************************************************************
                                ;--

021022                                  BGNTST
021022                                                                  T1::
                    000001              TNUM == TNUM + 1            ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021022  012767  000001  161224           MOV    @TNUM,TSTNUM         ;SET UP THE TEST NUMBER.        (1)
021030  012767  177777  161212           MOV    @-1,CTRLCF           ;INDICATE THAT WE ARE IN A TEST.
021036  012767  000145  163006           MOV    @101..ERRNBR         ;SET THE TEST ERROR NUMBER IN THE TABLE.
021044  012767  005262  163002           MOV    @EMO103,ERRMSG       ;SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
021052  012767  011430  162776           MOV    @ERO101,ERRBLK       ;SET-UP THE ERROR ROUTINE IN THE ERROR TABLE.
                                ;+
                                ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
                                ;-
021060  016767  156720  161210           MOV    4,TP4VEC             ;SAVE THE EXISTING 004 TRAP VECTOR.
021066  012767  017712  156710           MOV    @TP4RTN,4            ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
021074  005005                           CLR    R5                   ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; HERE BEGINS THE LOOP TO TEST THE REGISTERS FOR A LINE.
                                ;    FIRST TEST THE CSR AND SET THE IND.ADR.REG (I.A.R) FIELD.
                                ;-
021076  016700  161124                   MOV    CSRA,R0              ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
021102  012701  021274                   MOV    @52$,R1              ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
021106  004767  173054                   JSR    PC,CKTRAP            ;MOVE AND CHECK FOR TRAP.
021112  103402                           BCS    4$                   ;IF NO TRAP, BYPASS ERROR.
021114  052705  100001                   BIS    @100001,R5           ;SET FATAL READ ERROR FLAGS.
021120  042767  000017  000146  4$:      BIC    @17,52$              ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
021126  010100                           MOV    R1,R0                ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
021130  016701  161072                   MOV    CSRA,R1              ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
021134  004767  173026                   JSR    PC,CKTRAP            ;MOVE AND CHECK FOR TRAP.
021140  103403                           BCS    6$                   ;IF NO TRAP, BYPASS ERROR.
021142  052705  100002                   BIS    @100002,R5           ;SET FATAL WRITE ERROR FLAGS.
021146  000434                           BR     40$                  ;EXIT AND REPORT FATAL ERROR.
                                ;+
                                ;    NOW, WE TEST EACH REGISTER FOR THIS LINE.
                                ;-
021150  012702  000010           6$:     MOV    @8.,R2               ;INIT REGISTER COUNTER TO 8.
021154  016767  161046  000110           MOV    CSRA,50$             ;INITIALIZE THE REGISTER POINTER.
021162  016700  000104           8$:     MOV    50$,R0               ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
021166  012701  021274                   MOV    @52$,R1              ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
021172  004767  172770                   JSR    PC,CKTRAP            ;PERFORM THE MOVE, CHECK FOR TRAP.
021176  103402                           BCS    10$                  ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
021200  052705  100001                   BIS    @100001,R5           ;SET FATAL READ ERROR FLAGS.
021204  010100                  10$:     MOV    R1,R0                ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
021206  016701  000060                   MOV    50$,R1               ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
```

```
 4955 021212  004767  172750                JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 4956 021216  103402                        BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4957 021220  052705  100002                 BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
 4958 021224  005267  000042          12$:    INC     50$             ;INCREMENT THE REGISTER
 4959 021230  005267  000036                 INC     50$             ; POINTER BY 2.
 4960 021234  005302                        DEC     R2              ;COUNT THE REGISTER.
 4961 021236  001351                        BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
 4962
 4963                                ;+
 4964                                ; DONE CHECKING DEVICE REGISTER ADDRESSES.
 4965                                ;    REPORT ANY ERRORS AND EXIT.
 4966                                ;-
 4967
 4968 021240  016767  161032  156536 40$:     MOV     IP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 4969 021246  005705                 TST     R5              ;CHECK THE ERROR FLAGS.
 4970 021250  100012                        BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
 4971                                ;+
 4972                                ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
 4973                                ;-
 4974 021252                                ERROR
      021252  104460                                                                TRAP    C$ERROR
 4975
 4976
 4977 021254                                DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
      021254  016700  160744                                                        MOV     UNITN,R0
      021260  104451                                                                TRAP    C$DODU
 4978 021262  005067  160762                CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
 4979 021266                                DOCLN                   ;ABORT THIS SUB PASS.
      021266  104444                                                                TRAP    C$DCLN
 4980 021270  000402                        BR      60$             ;
 4981
 4982                                ;***************** LOCAL STORAGE. *********************************************
 4983 021272  000000                50$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4984 021274  000000                52$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4985                                ;******************** END ****************************************************
 4986
 4987 021276  005067  160746          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 4988 021302                                ENDTST
      021302                                                                    L10026:
      021302  104401                                                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DMASTA -
                                ;++ ************************************************************************
                                ;*                          - DMA START BIT TEST -
                                ;*      THIS TEST VERIFIES THAT THE DMA_START BIT IN THE DUT'S LINE CONTROL
                                ;*      REGISTERS WILL INITIATE DMA TRANSMISSION ON THE SELECTED LINE.
                                ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, ON ALL ACTIVE LINES.
                                ;*
                                ;-- ************************************************************************
021304                                  BGNTST
021304                                                                          T2::
021304                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
021304  012700  000240                                                        MOV     #PRIO5,R0
021310  104441                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021312  012767  000002  160734           MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (40)
021320  012767  177777  160722           MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
021326  012767  000001  162514           MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021334  012767  007641  162510           MOV     #4001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
021342  012767  005403  162504           MOV     #EM4001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
021350  012767  013136  162500           MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 4001 <<<<<.
                                ;-
021356  004767  172634                  JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
021362  103145                          BCC     50$             ;RESET FAILURE?, ABORT THIS TEST.
021364  004767  173160                  JSR     PC,INDATP       ;INITIALSE THE 256 BYTE DATA PATTERN.
                                ;+
                                ; SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS 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 ACTIVE LINES.
                                ;-
021370  016705  160624                   MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
021374  012700  000204                   MOV     #204,R0         ;PASS THE LNCTRL CONTENTS.
021400  004767  176146                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
021404  012700  177670                   MOV     #177670,R0      ;PASS THE LPR CONTENTS.
021410  004767  176166                  JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
021414  004767  175502                  JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; SET-UP OUTER LOOP TO TEST THE DMA_START BIT ON ALL ACTIVE LINES.
                                ;-
021420  016705  160574                   MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
021424  005001                           CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
021426  012767  007642  162416  2$:      MOV     #4002.,ERRNBR   ;SET THE ERROR NUMBER TO 4002.
021434  000241                           CLC                     ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
021436  006005                           ROR     R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
021440  103112                          BCC     14$             ;DO NOT TEST THE LINE IF IT IS INACTIVE.
021442  004767  173722                  JSR     PC,PUFIFO       ;PURGE THE FIFO.
021446  103113                          BCC     50$             ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
                                ;+
                                ; PERFORM DMA_START BIT TESTING ON EACH LINE INDIVIDUALLY.
                                ; TEST EACH DMA_START BIT BEFORE TX'ING DATA PATTERN, REPORT ERROR IF SET.
                                ; SET DMA_START BIT ON LUT, VERIFY IT IS SET, REPORT ERROR IF CLEAR.
                                ; WAIT FOR DMA TO COMPLETE.
```

```
                                        ; VERIFY DMA_START BIT IS CLEAR, REPORT ERROR IF SET.
                                        ; VERIFY CORRECT NUMBER OF CHARS WERE RECEIVED, REPORT ERROR IF < EXPECTED.
                                        ;-
021450  005267  162376                    INC     ERRNBR          ;SET ERROR NUMBER TO 4003.
021454  012702  002650                    MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
021460  012703  000144                    MOV     #100.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
021464  004767  172710                    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
021470  103062                            BCC     12$             ;GO REPORT ERROR IF DMA_START BIT SET.
                                        ;+
                                        ; TEST THE STATE OF THE DMA_START BIT ON THE LINE UNDER TEST.
                                        ;  REPORT ERROR IF DMA_START BIT IS CLEAR.
                                        ;-
021472  005267  162354                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4004.
021476  010177  160524                    MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
021502  105777  160534                    TSTB    @TXAD2A         ;TEST THE STATE OF THE DMA_START BIT.
021506  100060                            BPL     12$             ;GO REPORT ERROR IF BIT IS CLEAR.
                                        ;+
                                        ; WAIT FOR DMA TRANSMISSION TO COMPLETE.
                                        ;-
021510  005267  162336          4$:       INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4005.
021514  010103                            MOV     R1,R3           ;SAVE THE LINE NUMBER.
021516  012701  170226                    MOV     #170226,R1      ;TEST BIT 15, TIMEOUT OF 150 MILLI SECS.
021522  016702  160500                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
021526  004767  175704                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
021532  103045                            BCC     10$             ;GO REPORT ERROR IF TIMEOUT OCCURRED.
021534  012704  000005                    MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
021540  004767  172516                    JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
021544  010301                            MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
                                        ;+
                                        ; TEST THE STATE OF THE DMA_START BIT ON THE LINE UNDER TEST.
                                        ;  REPORT ERROR IF DMA_START BIT IS SET.
                                        ;-
021546  005267  162300                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4006.
021552  010177  160450                    MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
021556  105777  160460                    TSTB    @TXAD2A         ;TEST THE STATE OF THE DMA_START BIT.
021562  100432                            BMI     12$             ;GO REPORT ERROR IF BIT IS STILL SET.
                                        ;+
                                        ; VERIFY THE NUMBER OF CHARS RECEIVED = NUMBER OF CHARS EXPECTED.
                                        ; REPORT ERROR IF COUNT IS INCORRECT.
                                        ; IF MORE THAN 128 BMP CODES ARE FOUND THEN REPORT ERROR AND EXIT TEST.
                                        ;-
021564  005003                            CLR     R3              ;CLEAR THE READ COUNTER.
021566  012704  000200                    MOV     #128.,R4        ;SET UP MAX BMP CODE READ COUNT.
021572  012767  007647  162252  6$:       MOV     #4007.,ERRNBR   ;SET ERROR NUMBER TO 4007.
021600  017702  160424                    MOV     @RBUFA,R2       ;READ THE CHARACTER FROM THE FIFO.
021604  100021                            BPL     12$             ;GO REPORT ERROR IF FIFO EMPTY TOO SOON.
021606  012700  170301                    MOV     #170301,R0      ;SET-UP BIT MASK OF A BMP CODE.
021612  040200                            BIC     R2,R0           ;TRY TO CLEAR THE BMP CODE MASK.
021614  001007                            BNE     8$              ;BRANCH IF NOT A BMP CODE.
021616  005267  162230                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4008.
021622  004767  174652                    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
021626  005304                            DEC     R4              ;DECREMENT MAX BMP CODE READ COUNT.
021630  001422                            BEQ     50$             ;GO REPORT ERROR IF TOO MANY BMP CODES FOUND.
021632  000757                            BR      6$              ;DO NOT COUNT THE BMP CODE AS A VALID CHAR.
021634  005203                  8$:       INC     R3              ;COUNT THIS CHARACTER.
021636  020327  000144                    CMP     R3,#100.        ;HAVE WE RECIEVED 100 CHARACTERS?.
021642  002753                            BLT     6$              ;LOOP UNTIL 100 (NON-BMP) CHARS ARE READ.
```

```
5101 021644  000410                        BR      14$             ;SKIP AROUND THE ERROR REPORT.
5102
5103                                       ;+
5104                                       ; REPORT ERROR, SKIP FURTHER TESTING ON THIS LINE.
5105                                       ;-
5106 021646  010301                 10$:    MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
5107
5108 021650  012702  005435          12$:    MOV     #EM4002,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
5109                                                                 ; "DMA_START BIT BAD ON LINE NN".
5110 021654                                 ERROR                   ;                >>>>> ERROR <<<<<.
     021654  104460                                                                      TRAP    C$ERROR
5111
5112                                   ;+
5113                                   ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5114                                   ;-
5115 021656  032767  000100  160324        BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5116 021664  001406                       BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5117
5118 021666  005201                 14$:    INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
5119 021670  005705                         TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5120 021672  001255                       BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
5121 021674  000402                       BR      60$             ;NO; EXIT THIS TEST.
5122
5123 021676  004767  174770          50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5124 021702  005067  160342          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5125
5126 021706                                ENDTST
     021706                                L10027:
     021706  104401                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DMABRT -
                                ;++ ************************************************************************
                                ;*                      - DMA ABORT/RESTART TEST -
                                ;*      THIS TEST VERIFIES THAT EACH DMA_ABORT BIT WILL CORRECTLY HALT
                                ;*      A DMA TRANSMISSION, AND RETURN A TX_ACTION.
                                ;*      IT WILL ALSO VERIFY THAT THE ABORTED DMA TRANSMISSION CAN BE RESUMMED,
                                ;*      AND THAT A TX_ACTION IS RETURNED UPON COMPLETION.
                                ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, ON ALL ACTIVE LINES.
                                ;*
                                ;-- ************************************************************************
021710                                  BGNTST
021710                                                                          T3::
021710                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
021710  012700  000240                                                        MOV     #PRI05,R0
021714  104441                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021716  012767  000003  160330           MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (41)
021724  012767  177777  160316           MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
021732  012767  000001  162110           MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021740  012767  010005  162104           MOV     #4101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
021746  012767  005471  162100           MOV     #EM4101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
021754  012767  013136  162074           MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 4101 <<<<<.
                                ;-
021762  004767  172230                  JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
021766  103164                          BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
021770  004767  172554                  JSR     PC,INDATP       ;INITIALISE 256 BYTE DATA PATTERN.
                                ;*
                                ; SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS 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 ACTIVE LINES.
                                ;-
021774  016705  160220                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
022000  012700  000204                  MOV     #204,R0         ;PASS THE LNCTRL CONTENTS.
022004  004767  175542                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
022010  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
022014  004767  175562                  JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
022020  004767  175076                  JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;*
                                ; PERFORM DMA_ABORT BIT TESTING ON EACH INDIVIDUAL (ACTIVE) LINE.
                                ;-
022024  016705  160170                  MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
022030  005001                          CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
022032  012767  010006  162012  2$:     MOV     #4102.,ERRNBR   ;SET THE ERROR NUMBER TO 4102.
022040  000241                          CLC                     ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
022042  006005                          ROR     R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
022044  103127                          BCC     10$             ;DO NOT TEST THE LINE IF IT IS INACTIVE.
022046  004767  173316                  JSR     PC,PUFIFO       ;PURGE THE FIFO.
022052  103130                          BCC     50$             ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
                                ;*
                                ;  CHECK THE DMA_ABORT BIT BEFORE ENABLING DMA, REPORT ERROR IF SET.
                                ;-
```

```
5182 022054  005267  161772                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4103.
5183 022060  010177  160142                    MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5184 022064  032777  000001  160144            BIT     #BIT0,@LNCTRA   ;TEST THE STATE OF THE DMA_ABORT BIT.
5185 022072  001105                           BNE     6$              ;GO REPORT ERROR IF BIT IS SET.
5186                                        ;+
5187                                        ; ENABLE DMA TX ON SELECTED LINE, WAIT FOR DMA TO TX APPROX 1/4 OF DATA.
5188                                        ;  ABORT THE DMA TRANSMISSION. WAIT FOR TX_ACTION TO BE RETURNED.
5189                                        ;-
5190 022074  005267  161752                    INC     ERRNBR          ;SET ERROR NUMBER TO 4104.
5191 022100  012702  002650                    MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
5192 022104  012703  000400                    MOV     #256.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
5193 022110  004767  172264                    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
5194 022114  103107                           BCC     50$             ;GO REPORT ERROR IF THERE ARE TX PROBLEMS.
5195                                        ;+
5196                                        ; WAIT FOR DMA TO TRANSMIT 1/4 OF THE DATA BEFORE ABORTING.
5197                                        ;-
5198 022116  010177  160104                    MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5199 022122  012704  000050                    MOV     #40.,R4         ;PASS THE DELAY TIME OF 40 MILLI SECONDS.
5200 022126  004767  172130                    JSR     PC,DELAY        ;WAIT FOR APPROX 1/4 OF DATA TO BE TX'D.
5201 022132  052777  000001  160076            BIS     #BIT0,@LNCTRA   ;ABORT THE DMA TRANSMISSION.
5202                                        ;+
5203                                        ; WAIT FOR TX_ACTION TO BE RETURNED, REPORT ERROR IF TIME-OUT OCCURS.
5204                                        ;-
5205 022140  005267  161706                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4105.
5206 022144  010103                            MOV     R1,R3           ;SAVE THE LINE NUMBER.
5207 022146  012701  170012                    MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5208 022152  016702  160050                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5209 022156  004767  175254                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
5210 022162  103050                           BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
5211 022164  010301                            MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
5212                                        ;+
5213                                        ; VERIFY DMA_START BIT CLEAR, REPORT ERROR IF SET.
5214                                        ;-
5215 022166  005267  161660                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4106.
5216 022172  012702  005557                    MOV     #EM4103,R2      ;SELECT MESSAGE TO BE REPORTED.
5217                                                                   ; "DMA_START BIT FOUND SET AFTER DMA ABORTED".
5218 022176  010177  160024                    MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5219 022202  105777  160034                    TSTB    @TXAD2A         ;TEST THE STATE OF THE DMA_START BIT.
5220 022206  100441                           BMI     8$              ;GO REPORT ERROR IF IT IS SET.
5221                                        ;+
5222                                        ; RESUME DMA TRANSMISSION BY CLEARING DMA_ABORT AND SETTING DMA_START.
5223                                        ;-
5224 022210  042777  000001  160020            BIC     #BIT0,@LNCTRA   ;CLEAR THE DMA_ABORT BIT.
5225 022216  052777  000200  160016            BIS     #BIT7,@TXAD2A   ;SET THE DMA_START BIT.
5226                                        ;+
5227                                        ; WAIT FOR DMA TRANSMISSION TO COMPLETE.
5228                                        ;-
5229 022224  005267  161622                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4107.
5230 022230  010103                            MOV     R1,R3           ;SAVE THE LINE NUMBER.
5231 022232  012701  170536                    MOV     #170536,R1      ;TEST BIT 15, TIMEOUT OF 350 MILLI SECS.
5232 022236  016702  157764                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5233 022242  004767  175170                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
5234 022246  103016                           BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
5235 022250  012704  000002                    MOV     #2,R4           ;PASS TIME-OUT OF 2 MILLI SECS.
5236 022254  004767  172002                    JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5237 022260  010301                            MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
5238                                        ;+
```

```
                                        ; TEST THE STATE OF THE DMA_ABORT BIT ON THE LINE UNDER TEST.
                                        ;  REPORT ERROR IF DMA_ABORT BIT IS SET.
                                        ;-
022262  005267  161564                      INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4108.
022266  010177  157734                      MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
022272  032777  000001  157736              BIT     #BIT0,@LNCTRA   ;TEST THE STATE OF THE DMA_ABORT BIT.
022300  001002                            BNE     6$              ;GO REPORT ERROR IF BIT IS SET.
022302  000410                            BR      10$             ;BRANCH TO CHECK FOR ANY MORE LINES TO TEST.
                                        ;+
                                        ; REPORT ERROR, SKIP FURTHER TESTING ON THIS LINE.
                                        ;-
022304  010301                  4$:       MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.

022306  012702  005523            6$:     MOV     #EM4102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
                                                                  ; "DMA_ABORT BIT BAD ON LINE NN".
022312                            8$:     ERROR                   ;                >>>>> ERROR <<<<<.
022312  104460                                                                              TRAP    C$ERROR

                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
022314  032767  000100  157666              BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
022322  001406                            BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                  ;DURING THE SOFTWARE QUESTIONS.
                                        ;+
                                        ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
                                        ;-
022324  005201                  10$:      INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
022326  005705                            TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
022330  001240                            BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
022332  000402                            BR      60$             ;NO; EXIT THIS TEST.

022334  004767  174332            50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
022340  005067  157704            60$:     CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.

022344                                    ENDTST
022344                                                                    L10030:
022344  104401                                                                    TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - DMAERR -
                                        ;++ ******************************************************************
                                        ;*                      - DMA ERROR BIT TEST -
                                        ;*      THIS TEST VERIFIES THAT THE TX.DMA.ERROR BIT IN THE CSR IS
                                        ;*      FUNCTIONING CORRECTLY.  THE DMA ERROR IS FORCED BY MAKING THE DUT
                                        ;*      ATTEMPT TO PERFORM A DMA TRANSFER FROM THE ADDRESS OF ITS OWN CSR.
                                        ;*      SINCE THE DEVICE CANNOT BE BOTH A BUS MASTER AND SLAVE AT THE SAME
                                        ;*      TIME, TIMEOUT WILL OCCUR WAITING FOR THE APPROPRIATE HANDSHAKE SIGNAL
                                        ;*      FROM THE DMA ADDRESS.
                                        ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK.
                                        ;*
                                        ;-- ******************************************************************

 022346                                         BGNTST
 022346                                                                                  T4::

 022346                                         SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
 022346  012700  000240                                                                          MOV     #PRI05,R0
 022352  104441                                                                                  TRAP    C$SPRI

         000004                                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 022354  012767  000004  157672                  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.       (42)
 022362  012767  177777  157660                  MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 022370  012767  000001  161452                  MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 022376  012767  010151  161446                  MOV     #4201.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 022404  012767  005643  161442                  MOV     #EM4201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
 022412  012767  011762  161436                  MOV     #ER0503,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 >>>>> 4201 <<<<<.
                                        ;-
 022420  004767  171572                         JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 022424  103120                                 BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
                                        ;+
                                        ; SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS ON ALL LINES.
                                        ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
                                        ; 2 STOP BITS.
                                        ; ENABLE TRANSMITTERS ON ALL LINES.
                                        ;-
 022426  004767  172036                         JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
 022432  103115                                 BCC     60$             ;EXIT THE TEST IF NO ACTIVE LINES.
 022434  010104                                  MOV    R1,R4           ;SAVE THE LINE NUMBER.
 022436  012700  000204                          MOV    #204,R0         ;PASS THE LNCTRL CONTENTS.
 022442  004767  175104                         JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
 022446  012700  177670                          MOV    #177670,R0      ;PASS THE LPR CONTENTS.
 022452  004767  175124                         JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON THE ACTIVE LINE.
 022456  004767  174440                         JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON THE ACTIVE LINE.
                                        ;+
                                        ; VERIFY THAT THE DMA.START BIT IS CLEAR BEFORE ATTEMPTING THE DMA TRANSFER.
                                        ;-
 022462  005267  161364                          INC    ERRNBR          ;SET THE ERROR NUMBER TO 4202.
 022466  032777  000200  157546                  BIT    #BIT07,@TXAD2A  ;TEST THE DMA START BIT.
 022474  001072                                 BNE     50$             ;BRANCH TO REPORT THE ERROR IF THE BIT IS SET.
                                        ;+
                                        ; SET UP THE DMA REGISTERS TO PERFORM THE TRANSFER FROM THE ADDRESS OF THE CSR.
                                        ;-
```

```
5329 022476  016777  157524  157534          MOV     CSRA,@TXAD1A    ;SET UP THE LOW 16 BITS OF THE DMA ADDR.
5330 022504  012777  000001  157532          MOV     #1,@TXBFCA      ;SET UP TO DMA ONE CHARACTER.
5331 022512  112777  000203  157522          MOVB    #203,@TXAD2A    ;SET UP THE 2 MSB'S AND INITIATE THE DMA.
5332 022520  012701  170012                  MOV     #170012,R1      ;TEST BIT 15, TIME OUT OF 10 MS.
5333 022524  016702  157476                  MOV     CSRA,R2         ;INDICATE TO TEST THE CSR.
5334 022530  005267  161316                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4203.
5335 022534  004767  174676                  JSR     PC,WAIBIS       ;WAIT FOR A TX-ACTION.
5336 022540  103050                          BCC     50$             ;REPORT THE ERROR IF NO TX-ACTION.
5337                                  ;+
5338                                  ; VERIFY THAT THE DMA ERROR BIT IS SET AND THE DMA START BIT IS CLEAR.
5339                                  ;-
5340 022542  005267  161304                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4204.
5341 022546  032777  010000  157452          BIT     #BIT12,@CSRA    ;TEST THE DMA ERROR BIT.
5342 022554  001436                          BEQ     2$              ;REPORT THE ERROR IF BIT IS CLEAR
5343 022556  005267  161270                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4205.
5344 022562  032777  000200  157452          BIT     #BIT07,@TXAD2A  ;TEST THE DMA-START BIT.
5345 022570  001034                          BNE     50$             ;REPORT THE ERROR IF THE BIT IS SET.
5346                                  ;+
5347                                  ; VERIFY THAT THE DMA ERROR BIT CLEARS WHEN A "GOOD" DMA TRANSFER IS PERFORMED.
5348                                  ;-
5349 022572  010401                          MOV     R4,R1           ;SET UP THE ACTIVE LINE NUMBER.
5350 022574  012702  002650                  MOV     #BUFBAS,R2      ;SET UP THE START ADDRESS OF THE DMA BUFFER.
5351 022600  012703  000001                  MOV     #1,R3           ;SET UP TO DMA ONE CHARACTER.
5352 022604  005267  161242                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4206.
5353 022610  004767  171564                  JSR     PC,DODMA        ;START THE DMA.
5354 022614  103022                          BCC     50$             ;REPORT THE ERROR IF ONE OCCURED.
5355 022616  012701  170036                  MOV     #170036,R1      ;SET UP TO TEST BIT15 WITH TIMEOUT OF 30 MS.
5356 022622  016702  157400                  MOV     CSRA,R2         ;INDICATE TO TEST THE CSR.
5357 022626  005267  161220                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4207.
5358 022632  004767  174600                  JSR     PC,WAIBIS       ;WAIT FOR A TX-ACTION.
5359 022636  103011                          BCC     50$             ;REPORT THE ERROR IF NO TX-ACTION OCCURED.
5360 022640  005267  161206                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4208.
5361 022644  032702  010000                  BIT     #BIT12,R2       ;TEST THE DMA ERROR BIT OF THE LAST CSR WORD READ.
5362 022650  001406                          BEQ     60$             ;EXIT THE TEST IF THE BIT IS CLEAR.
5363                                  ;+
5364                                  ; REPORT THE ERROR, DMA ERROR BIT BAD.
5365                                  ;-
5366 022652  012701  005675          2$:     MOV     #EM4202,R1      ;SET THE MESSAGE
5367                                                                  ; "DMA ERROR BIT BAD".
5368 022656                                  ERROR
     022656  104460                                                                      TRAP    C$ERROR
5369 022660  000402                          BR      60$             ;EXIT THE TEST.
5370
5371 022662  004767  174004          50$:    JSR     PC,TSABRT       ;REPORT THE NON-RELATED TEST ERROR.
5372 022666  005067  157356          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5373
5374 022672                                  ENDTST
     022672                                                                  L10031:
     022672  104401                                                                  TRAP    C$ETST
```

```
                                     .SBTTL  HARDWARE TEST           - OAUTOI -
                                     ;***********************************************************************
                                     ;*                   - OAUTO BIT INACTIVE TEST -
                                     ;*
                                     ;*      THIS TEST VERIFIES THAT THE DUT'S OAUTO FUNCTION BEHAVES CORRECTLY
                                     ;*      WHEN INACTIVE, IE OAUTO BIT CLEAR.
                                     ;*      THIS TEST WILL ONLY EXECUTE IF STAGGERED LOOPBACK MODE IS SELECTED.
                                     ;*      THE SPECIAL STAGGERED LOOPBACK CONNECTOR MUST BE FITTED.
                                     ;*
                                     ;--*********************************************************************

022674                                       BGNTST
022674                                                                              T5::
022674  126727  157322  000002                CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
022702  001402                                BEQ     .+6             ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
022704  000167  000556                        JMP     60$             ;EXIT THIS TEST.
022710                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
022710  012700  000240                                                                  MOV     #PRIO5,R0
022714  104441                                                                          TRAP    C$SPRI
        000005                                 TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
022716  012767  000005  157330                MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (49)
022724  012767  177777  157316                MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
022732  012767  000001  161110                MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
022740  012767  011445  161104                MOV     #4901.,ERRNBR   ;SET ERROR NUMBER TO 4901.
022746  012767  005717  161100                MOV     #EM4901,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
022754  012767  013136  161074                MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                     ;+
                                     ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                     ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                     ; THIS SUBROUTINE REPORTS ERROR >>>>> 4901 <<<<<.
                                     ;-
022762  004767  171230                        JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
022766  103402                                BCS     .+6             ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
022770  000167  000472                        JMP     60$             ;EXIT THIS TEST.
                                     ;+
                                     ; SET-UP THE ASSOCIATED TX/RX LINE NUMBER TABLES.
                                     ;-
022774  004767  170442                        JSR     PC,ASLNTL       ;INITIALISE THE ASSOCIATED TX/RX TABLES.
                                     ;+
                                     ; SET EXTERNAL LOOPBACK, DISABLE OAUTO AND ENABLE RECEIVER 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.
                                     ;-
023000  016705  157214                        MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
023004  012700  000004                        MOV     #4,R0           ;PASS THE LNCTRL CONTENTS.
023010  004767  174536                        JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
023014  012705  177777                        MOV     #MAPLNS,R5      ;PASS BIT MAP OF ALL LINES.
023020  012700  177670                        MOV     #177670,R0      ;PASS THE LPR CONTENTS.
023024  004767  174552                        JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
023030  004767  174066                        JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                     ;+
                                     ; SET UP OUTER LOOP FOR TESTING ACTIVE LINES IN BOTH LINE GROUPS.
                                     ;-
023034  012703  100000                        MOV     #100000,R3      ;SET UP LOOP CONTROL FLAG.
023040  016705  157154                        MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
023044  046705  157212                        BIC     LGRP2M,R5       ;REMOVE LINES IN GROUP 2.
```

```
5430 023050  010567  000404          2$:     MOV     R5,45$          ;SAVE THE CURRENT LINE GROUP.
5431 023054  005067  000376                  CLR     40$             ;CLEAR THE LINE NUMBER COUNTER.
5432 023060  016701  000372          4$:     MOV     40$,R1          ;COPY THE LINE NUMBER.
5433 023064  000241                          CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5434 023066  006005                          ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5435 023070  103064                        BCC     8$              ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5436                                    ;+
5437                                    ; TEST THE STATE OF THE OAUTO BIT ON THE LINE UNDER TEST.
5438                                    ;  REPORT ERROR IF IT IS FOUND SET, AND SKIP FURTHER TESTING OF THAT LINE.
5439                                    ;-
5440 023072  012767  011446  160752          MOV     #4902.,ERRNBR   ;SET THE ERROR NUMBER TO 4902.
5441 023100  010177  157122                  MOV     R1,@CSRA        ;SELECT THE LINE TO BE TESTED.
5442 023104  032777  000020  157124          BIT     #BIT4,@LNCTRA   ;TEST THE STATE OF THE OAUTO BIT.
5443 023112  001410                        BEQ     6$              ;SKIP ERROR REPORT IF OAUTO BIT IS CLEAR.
5444 023114  012702  005760                  MOV     #EM4902,R2      ;PASS THE ERROR MESSAGE.
5445                                                                ;  "OAUTO BIT BAD ON LINE NN"
5446 023120                                ERROR                   ;                  >>>>> ERROR #4902 <<<<<.
     023120  104460                                                                          TRAP    C$ERROR
5447
5448                                    ;+
5449                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5450                                    ;-
5451 023122  032767  000100  157060          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5452 023130  001556                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5453                                                                ;DURING THE SOFTWARE QUESTIONS.
5454
5455 023132  000443                        BR      8$              ;SKIP FURTHER TESTING OF THIS LINE.
5456                                    ;+
5457                                    ; TRANSMIT THE XOFF (ASCII DC3) ON THE ASSOCIATED LINE.
5458                                    ;-
5459 023134  116177  004010  157064  6$:     MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5460 023142  112777  000023  157064          MOVB    #23,@TDATA      ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
5461                                    ;+
5462                                    ; WAIT FOR TRANSMISSION TO COMPLETE.
5463                                    ;-
5464 023150  005267  160676                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4903.
5465 023154  012701  170012                  MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5466 023160  016702  157042                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5467 023164  004767  174246                JSR     PC,WAIBIS       ;WAIT FOR TRANSMISSION TO COMPLETE.
5468 023170  103134                        BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
5469 023172  012704  000005                  MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
5470 023176  004767  171060                JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5471                                    ;+
5472                                    ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5473                                    ;  REPORT ERROR IF TX_ENABLE BIT IS CLEAR.
5474                                    ;-
5475 023202  005267  160644                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4904.
5476 023206  016701  000244                  MOV     40$,R1          ;GET THE NUMBER OF THE LINE TEST.
5477 023212  010177  157010                  MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5478 023216  005777  157020                  TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
5479 023222  100407                        BMI     8$              ;SKIP ERROR REPORT IF BIT IS SET.
5480 023224  012702  005760                  MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
5481                                                                ;  "OAUTO BIT BAD ON LINE NN".
5482 023230                                ERROR                   ;                  >>>>> ERROR #4904 <<<<<.
     023230  104460                                                                          TRAP    C$ERROR
5483
5484                                    ;+
```

```
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
5487 023232  032767  000100  156750          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5488 023240  001512                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5489                                                                ;DURING THE SOFTWARE QUESTIONS.
5490
5491 023242  005267  000210          8$:     INC     40$             ;INCREMENT THE LINE NUMBER.
5492 023246  005705                        TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5493 023250  001303                        BNE     4$              ;
5494                                    ;+
5495                                    ; DISABLE TRANSMITTERS ON THE SELECTED LINES IN THE CURRENT LINE GROUP.
5496                                    ;-
5497 023252  016705  000202                MOV     45$,R5          ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
5498 023256  004767  173544                JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
5499 023262  016705  000172                MOV     45$,R5          ;GET THE CURRENT ACTIVE LINE GROUP AGAIN.
5500 023266  005067  000164                CLR     40$             ;CLEAR THE LINE COUNTER.
5501 023272  012767  011451  160552  10$:  MOV     #4905.,ERRNBR   ;SET ERROR NUMBER TO 4905.
5502 023300  016701  000152                MOV     40$,R1          ;COPY THE LINE NUMBER.
5503 023304  000241                        CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5504 023306  006005                        ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5505 023310  103041                        BCC     12$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5506                                    ;+
5507                                    ; TRANSMIT THE XON (ASCII DC1) ON THE ASSOCIATED LINE.
5508                                    ;-
5509 023312  116177  004010  156706        MOVB    TXRLN8(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5510 023320  112777  000021  156706        MOVB    #21,@TDATA      ;TRANSMIT THE XON CHARACTER TO THE LUT.
5511                                    ;+
5512                                    ; WAIT FOR TRANSMISSION TO COMPLETE.
5513                                    ;-
5514 023326  012701  170012                MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5515 023332  016702  156670                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5516 023336  004767  174074                JSR     PC,WAIBIS       ;WAIT FOR TRANSMISSION  TO COMPLETE.
5517 023342  103047                        BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
5518 023344  012704  000005                MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
5519 023350  004767  170706                JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5520                                    ;+
5521                                    ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5522                                    ;  REPORT ERROR IF TX_ENABLE BIT IS SET.
5523                                    ;-
5524 023354  005267  160472                INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4906.
5525 023360  016701  000072                MOV     40$,R1          ;GET THE NUMBER OF THE LINE UNDER TEST.
5526 023364  010177  156636                MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5527 023370  005777  156646                TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
5528 023374  100007                        BPL     12$             ;SKIP ERROR REPORT IF BIT IS CLEAR.
5529 023376  012702  005760                MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
5530                                                                ;  "DAUTO BIT BAD ON LINE NN".
5531 023402                                ERROR                   ;                  >>>>> ERROR #4906 <<<<<.
     023402  104460                                                                          TRAP    C$ERROR
5532
5533                                    ;+
5534                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5535                                    ;-
5536 023404  032767  000100  156576        BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5537 023412  001425                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5538                                                                ;DURING THE SOFTWARE QUESTIONS.
5539
5540 023414  005267  000036          12$:    INC     40$             ;INCREMENT THE LINE NUMBER.
```

```
5541 023420  005705                        TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5542 023422  001323                        BNE     10$             ;
5543                                ;*
5544                                ; CHECK LOOP CONTROL FLAG TO DETERMINE IF BOTH SETS OF LINES HAVE BEEN TESTED
5545                                ;  IF THIS IS THE FIST TIME AROUND, RE-ENABLE TX ON ALL LINES, GENERATE ACTIVE
5546                                ;  BIT MAP FOR SECOND LINE GROUP.
5547                                ;-
5548 023424  005703                        TST     R3              ;HAVE BOTH LINE GROUPS BEEN TESTED?.
5549 023426  001417                        BEQ     60$             ;YES; THEN EXIT THIS TEST.
5550 023430  005003                        CLR     R3              ;NO; CLEAR THE LOOP CONTROL FLAG.
5551 023432  012705  177777                MOV     @MAPLNS,R5      ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5552 023436  004767  173460                JSR     PC,TXENBL       ;RE-ENABLE TRANSMISSION ON ALL LINES.
5553 023442  016705  156552                MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
5554 023446  046705  156606                BIC     LGRP1M,R5       ;REMOVE ALL ACTIVE LINES IN GROUP 1.
5555 023452  000167  177372                JMP     2$              ;ONCE MORE AROUND AND WE ARE DONE.
5556
5557 023456  000000                40$:    .WORD   0               ;STORAGE FOR CURRENT LINE NUMBER.
5558 023460  000000                45$:    .WORD   0               ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5559 023462  004767  173204        50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5560 023466  005067  156556        60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5561
5562 023472                                ENDTST
     023472                                                                L10032:
     023472  104401                                                              TRAP    C$ETST
```

```
5564
5565                                    .SBTTL  HARDWARE TEST           - OAUTOA -
5566                                    ;+**********************************************************************
5567                                    ;*                      - OAUTO BIT ACTIVE TEST -
5568                                    ;*
5569                                    ;*      THIS TEST VERIFIES THAT THE DUT'S OAUTO FUNCTION BEHAVES CORRECTLY
5570                                    ;*      WHEN ACTIVE, IE OAUTO BIT ASSERTED HIGH.
5571                                    ;*      THIS TEST WILL ONLY EXECUTE IF THE STAGGERED LOOPBACK MODE IS SELECTED.
5572                                    ;*      THE SPECIAL STAGGERED LOOPBACK CONNECTOR MUST BE FITTED.
5573                                    ;*
5574                                    ;-**********************************************************************
5575
5576 023474                                    BGNTST
       023474                                                                   T6::
5577 023474  126727  156522  000002            CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
5578 023502  001402                     BEQ     .+6             ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
5579 023504  000167  000556             JMP     60$             ;EXIT THIS TEST.
5580 023510                             SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
       023510  012700  000240                                                   MOV     #PRIO5,R0
       023514  104441                                                           TRAP    C$SPRI
5581           000006                    TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5582 023516  012767  000006  156530      MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (50)
5583 023524  012767  177777  156516      MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
5584 023532  012767  000001  160310      MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5585 023540  012767  011611  160304      MOV     #5001.,ERRNBR   ;SET ERROR NUMBER TO 5001.
5586 023546  012767  006012  160300      MOV     #EM5001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5587 023554  012767  013136  160274      MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5588                                    ;*
5589                                    ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5590                                    ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5591                                    ; THIS SUBROUTINE REPORTS ERROR >>>>> 5001 <<<<<.
5592                                    ;-
5593 023562  004767  170430                    JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
5594 023566  103402                            BCS     .+6             ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
5595 023570  000167  000472                    JMP     60$             ;EXIT THIS TEST.
5596                                    ;*
5597                                    ; SET-UP THE ASSOCIATED TX/RX LINE NUMBER TABLES.
5598                                    ;-
5599 023574  004767  167642                    JSR     PC,ASLNTL       ;INITIALISE THE ASSOCIATED TX/RX TABLES.
5600                                    ;*
5601                                    ; SET EXTERNAL LOOPBACK,ENABLE OAUTO AND RECEIVER FUNCTIONS ON ALL ACTIVE LINES
5602                                    ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
5603                                    ; 2 STOP BITS.
5604                                    ; ENABLE TRANSMITTERS ON ALL LINES.
5605                                    ;-
5606 023600  016705  156414                    MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
5607 023604  012700  000024                    MOV     #24,R0          ;PASS THE LNCTRL CONTENTS.
5608 023610  004767  173736                    JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
5609 023614  012705  177777                    MOV     #MAPLNS,R5      ;PASS BIT MAP OF ALL LINES.
5610 023620  012700  177670                    MOV     #177670,R0      ;PASS THE LPR CONTENTS.
5611 023624  004767  173752                    JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
5612 023630  004767  173266                    JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
5613                                    ;*
5614                                    ; SET UP OUTER LOOP FOR TESTING ACTIVE LINES IN BOTH LINE GROUPS.
5615                                    ;-
5616 023634  012703  100000                    MOV     #100000,R3      ;SET-UP LOOP CONTROL FLAG.
5617 023640  016705  156354                    MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
```

```
5618 023644  046705  156412                BIC     LGRP2M,R5       ;REMOVE LINES IN GROUP 2.
5619 023650  010567  000404          2$:    MOV     R5,45$          ;SAVE THE CURRENT LINE GROUP.
5620 023654  005067  000376                CLR     40$             ;CLEAR THE LINE NUMBER COUNTER.
5621 023660  016701  000372          4$:    MOV     40$,R1          ;COPY THE LINE NUMBER.
5622 023664  000241                        CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5623 023666  006005                        ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5624 023670  103064                       BCC     8$              ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5625                                    ;+
5626                                    ; TEST THE STATE OF THE OAUTO BIT ON THE LINE UNDER TEST.
5627                                    ;  REPORT ERROR IF IT IS FOUND CLEAR, AND SKIP FURTHER TESTING OF THAT LINE.
5628                                    ;-
5629 023672  012767  011612  160152          MOV     #5002.,ERRNBR   ;SET THE ERROR NUMBER TO 5002.
5630 023700  010177  156322                MOV     R1,@CSRA        ;SELECT THE LINE TO BE TESTED.
5631 023704  032777  000020  156324          BIT     #BIT4,@LNCTRA   ;TEST THE STATE OF THE OAUTO BIT.
5632 023712  001010                       BNE     6$              ;SKIP ERROR REPORT IF OAUTO BIT IS SET.
5633 023714  012702  005760                MOV     #EM4902,R2      ;PASS THE ERROR MESSAGE.
5634                                                                ; "OAUTO BIT BAD ON LINE NN"
5635 023720                                ERROR                   ;               >>>>> ERROR #5002 <<<<<.
     023720  104460                                                                    TRAP    C$ERROR
5636
5637                                    ;+
5638                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5639                                    ;-
5640 023722  032767  000100  156260          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5641 023730  001556                       BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5642                                                                ;DURING THE SOFTWARE QUESTIONS.
5643
5644 023732  000443                        BR      8$              ;SKIP FURTHER TESTING OF THIS LINE.
5645                                    ;+
5646                                    ; TRANSMIT THE XOFF (ASCII DC3) ON THE ASSOCIATED LINE.
5647                                    ;-
5648 023734  116177  004010  156264  6$:    MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5649 023742  112777  000023  156264          MOVB    #23,@TDATA      ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
5650                                    ;+
5651                                    ; WAIT FOR TRANSMISSION TO COMPLETE.
5652                                    ;-
5653 023750  005267  160076                INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5003.
5654 023754  012701  170012                MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5655 023760  016702  156242                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5656 023764  004767  173446                JSR     PC,WAIBIS       ;WAIT FOR TRANSMISSION TO COMPLETE.
5657 023770  103134                       BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
5658 023772  012704  000005                MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
5659 023776  004767  170260                JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5660                                    ;+
5661                                    ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5662                                    ;  REPORT ERROR IF TX_ENABLE BIT IS SET.
5663                                    ;-
5664 024002  005267  160044                INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5004.
5665 024006  016701  000244                MOV     40$,R1          ;GET THE NUMBER OF THE LINE TEST.
5666 024012  010177  156210                MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5667 024016  005777  156220                TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
5668 024022  100007                       BPL     8$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
5669 024024  012702  005760                MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
5670                                                                ; "OAUTO BIT BAD ON LINE NN".
5671 024030                                ERROR                   ;               >>>>> ERROR #5004 <<<<<.
     024030  104460                                                                    TRAP    C$ERROR
5672
```

```
                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
024032  032767  000100  156150            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
024040  001512                          BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                ;DURING THE SOFTWARE QUESTIONS.

024042  005267  000210          8$:     INC     40$             ;INCREMENT THE LINE NUMBER.
024046  005705                          TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
024050  001303                          BNE     4$              ;
                                    ;+
                                    ; DISABLE TRANSMITTERS ON THE SELECTED LINES IN THE CURRENT LINE GROUP.
                                    ;-
024052  016705  000202                    MOV     45$,R5          ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
024056  004767  172744                  JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
024062  016705  000172                    MOV     45$,R5          ;GET THE CURRENT LINE ACTIVE LINE GROUP AGAIN.
024066  005067  000164                    CLR     40$             ;CLEAR THE LINE COUNTER.
024072  012767  011615  157752  10$:      MOV     #5005.,ERRNBR   ;SET ERROR NUMBER TO 5005.
024100  016701  000152                    MOV     40$,R1          ;COPY THE LINE NUMBER.
024104  000241                          CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
024106  006005                          ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
024110  103041                        BCC     12$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                  ;+
                                  ; TRANSMIT THE XON (ASCII DC1) ON THE ASSOCIATED LINE.
                                  ;-
024112  116177  004010  156106            MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
024120  112777  000021  156106            MOVB    #21,@TDATA      ;TRANSMIT THE XON CHARACTER TO THE LUT.
                                  ;+
                                  ; WAIT FOR TRANSMISSION TO COMPLETE.
                                  ;-
024126  012701  170012                    MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
024132  016702  156070                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
024136  004767  173274                  JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
024142  103047                          BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
024144  012704  000005                    MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
024150  004767  170106                  JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
                                  ;+
                                  ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
                                  ;  REPORT ERROR IF TX_ENABLE BIT IS CLEAR.
                                  ;-
024154  005267  157672                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5006.
024160  016701  000C72                    MOV     40$,R1          ;GET THE NUMBER OF THE LINE UNDER TEST.
024164  010177  156036                    MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
024170  005777  156046                    TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
024174  100407                          BMI     12$             ;SKIP ERROR REPORT IF BIT IS SET.
024176  012702  005760                    MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
                                                                ; "OAUTO BIT BAD ON LINE NN".
024202                                  ERROR                   ;                >>>>> ERROR #5006 <<<<<.
024202  104460                                                                            TRAP    C$ERROR

                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                ;-
024204  032767  000100  155776            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
024212  001425                          BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                ;DURING THE SOFTWARE QUESTIONS.
```

```
5729 024214  005267  000036          12$:     INC     40$             ;INCREMENT THE LINE NUMBER.
5730 024220  005705                           TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5731 024222  001323                          BNE     10$             ;
5732                                        ;+
5733                                        ; CHECK LOOP CONTROL FLAG TO DETERMINE IF BOTH SETS OF LINES HAVE BEEN TESTED
5734                                        ;  IF THIS IS THE FIST TIME AROUND, RE-ENABLE TX ON ALL LINES, GENERATE ACTIVE
5735                                        ;  BIT MAP FOR SECOND LINE GROUP.
5736                                        ;-
5737 024224  005703                           TST     R3              ;HAVE BOTH LINE GROUPS BEEN TESTED?.
5738 024226  001417                          BEQ     60$             ;YES; THEN EXIT THIS TEST.
5739 024230  005003                           CLR     R3              ;NO; CLEAR THE LOOP CONTROL FLAG.
5740 024232  012705  177777                   MOV     #MAPLNS;R5      ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5741 024236  004767  172660                  JSR     PC,TXENBL       ;RE-ENABLE TRANSMISSION ON ALL LINES.
5742 024242  016705  155752                   MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
5743 024246  046705  156006                   BIC     LGRP1M,R5       ;REMOVE ALL ACTIVE LINES IN GROUP 1.
5744 024252  000167  177372                  JMP     2$              ;ONCE MORE AROUND AND WE ARE DONE.
5745
5746 024256  000000                  40$:     .WORD   0               ;STORAGE FOR CURRENT LINE NUMBER.
5747 024260  000000                  45$:     .WORD   0               ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5748 024262  004767  172404          50$:     JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5749 024266  005067  155756          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5750
5751 024272                                  ENDTST
     024272                                                                  L10033:
     024272  104401                                                                  TRAP    C$ETST
```

```

                                   .SBTTL  HARDWARE TEST           - IAUTOI -
                                   ;++*********************************************************************
                                   ;*                        - IAUTO BIT INACTIVE TEST -
                                   ;*
                                   ;*      THIS TEST VERIFIES THAT THE DUT'S IAUTO FUNCTION BEHAVES CORRECTLY
                                   ;*      WHEN INACTIVE, IE. IAUTO BIT CLEAR.
                                   ;*      ALL ACTIVE LINES ARE TESTED INDIVIDUALLY BY FILLING THE FIFO
                                   ;*      THEN READING THE RECEIVED DATA CHECKING FOR THE PRESENCE OF
                                   ;*      XOFF(ASCII DC3) OR XON (ASCII DC1) CHARACTERS.
                                   ;*      IF ANY ARE FOUND THEN APPROPRIATE ERRORS ARE REPORTED.
                                   ;*      ANY BMP CODES THAT ARE FOUND WILL BE PLACED ON THE BMP CODE QUEUE,
                                   ;*      TO BE REPORTED LATER.
                                   ;*      THE CHARACTERS ARE TRANSMITTED ON ALL ACTIVE LINES, IN INTERNAL
                                   ;*      LOOPBACK MODE.
                                   ;*
                                   ;--*********************************************************************

024274                                     BGNTST
024274                                                                      T7::
024274                                     SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
024274  012700  000240                                                                      MOV     #PRI05,R0
024300  104441                                                                              TRAP    C$SPRI
        000007                              TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024302  012767  000007  155744              MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (51)
024310  012767  177777  155732              MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
024316  012767  000001  157524              MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
024324  012767  011755  157520              MOV     #5101.,ERRNBR  ;SET ERROR NUMBER TO 5101.
024332  012767  006051  157514              MOV     #EM5101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
024340  012767  013136  157510              MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                   ;+
                                   ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                   ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                   ; THIS SUBROUTINE REPORTS ERROR >>>>> 5101 <<<<<.
                                   ;-
024346  004767  167644                     JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
024352  103156                             BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.


                                   ;+
                                   ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                   ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                   ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                   ;-
024354  004767  170220                     JSR     PC,INDTPX       ;INITIALISE DATA PATTERN.

                                   ;+
                                   ; SET INTERNAL LOOPBACK, DISABLE IAUTO, ENABLE RECEIVER ON THE SELECTED LINE.
                                   ; SET LPR TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                   ;-
024360  016705  155634                     MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
024364  012700  000204                     MOV     #204,R0         ;PASS INT'L LOPBCK, ENABLE RX, DISABLE IAUTO.
024370  004767  173156                     JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGISTER.
024374  012700  177670                     MOV     #177670,R0      ;PASS THE LPR CONTENTS.
024400  004767  173176                     JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
024404  012704  000012                     MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
024410  004767  167646                     JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.


                                   ;+
```

```
                                        ; SET UP LOOP FOR ALL ACTIVE LINES.
                                        ; TEST THE STATE OF THE IAUTO BIT PRIOR TO TRANSMITTING THE DATA PATTERN.
                                        ;  IF THE BIT IS SET, THEN REPORT THE ERROR AND SKIP TRANSMITTING
                                        ;  THE DATA PATTERN ON THE SELECTED LINE.
                                        ; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                        ; EMPTY THE FIFO, AND VERIFY NO XOFF OR XON CHARS WERE FOUND.
                                        ;-
024414  005001                                CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
024416  005067  000264                      CLR     55$             ;CLEAR STORAGE FOR LINE NUMBER.
024422  012767  011756  157422  2$:         MOV     #5102.,ERRNBR   ;SET THE ERROR NUMBER TO 5102.
024430  004767  170734                    JSR     PC,PUFIFO       ;PURGE THE FIFO.
024434  103121                           BCC     50$             ;GO REPORT ERROR IF FIFO DID NOT PURGE.
024436  000241                            CLC                     ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
024440  006005                            ROR     R5              ;ROTATE THE BIT MAP INTO THE CARRY BIT.
024442  103107                           BCC     12$             ;BRANCH IF LINE IS INACTIVE.
                                        ;+
                                        ; TEST THE IAUTO BIT ON THE SELECTED ACTIVE LINE.
                                        ; REPORT ERROR IF IT IS SET.
                                        ;  DO NOT TRANSMIT THE DATA PATTERN ON THE SELECTED LINE.
                                        ;-
024444  005267  157402                        INC     ERRNBR          ;SET ERROR NUMBER TO 5103.
024450  010177  155552                        MOV     R1,@CSRA        ;SELECT LINE TO TEST.
024454  032777  000002  155554                BIT     #BIT1,@LNCTRA   ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
024462  001410                           BEQ     4$              ;SKIP ERROR IF IAUTO BIT CLEAR.
024464  012702  006106                        MOV     #EM5102,R2      ;PASS THE CORRECT ERROR MESSAGE.
024470                                        ERROR                   ;                >>>>> ERROR <<<<<.
024470  104460                                                                                TRAP    C$ERROR
                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
024472  032767  000100  155510                BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
024500  001503                           BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                ;DURING THE SOFTWARE QUESTIONS.

024502  000467                           BR      12$             ;SKIP TRANSMITTING DATA PATTERN.

                                        ;+
                                        ; TRANSMIT DATA PATTERN OF 256 CHARS.
                                        ;-
024504  005267  157342          4$:         INC     ERRNBR          ;SET ERROR NUMBER TO 5104.
024510  012702  002650                      MOV     #BUF3AS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
024514  012703  000400                      MOV     #256.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
024520  004767  167654                    JSR     PC,DGDMA        ;TRANSMIT THE DATA PATTERN.
024524  103065                           BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.

                                        ;+
                                        ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER PLUS XOFF
                                        ; TO ARRIVE IN THE FIFO.
                                        ;-
024526  005267  157320                        INC     ERRNBR          ;SET ERROR NUMBER TO 5105.
024532  012701  170536                        MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
024536  016702  155464                        MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
024542  004767  172670                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
024546  103054                           BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
024550  012704  000012                      MOV     #10.,R4         ;PASS DELAY OF 10 MILLI SECS.
024554  004767  167502                    JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
```

```
5863
5864                                        ;+
5865                                        ; READ 256 CHARS FROM THE FIFO.  REPORT ERROR IF ANY XOFF'S OR XON'S
5866                                        ; ARE FOUND.
5867                                        ;-
5868 024560  005267  157266                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5106.
5869 024564  012701  000400                    MOV     #256.,R1        ;INITIALISE THE READ COUNTER.
5870 024570  017702  155434          6$:       MOV     @RBUFA,R2       ;READ CHAR FROM THE FIFO.
5871 024574  100041                           BPL     50$             ;GO REPORT ERROR IF FIFO EMPTY.
5872                                        ;+
5873                                        ; CHECK FOR BMP CODE IN THE FIFO.  SAVE ANY FOUND ON THE QUEUE.
5874                                        ;-
5875 024576  012700  170301                    MOV     #170301,R0      ;SET UP BMP BIT MASK.
5876 024602  040200                           BIC     R2,R0           ;TRY TO CLEAR ALL THE BMP BITS.
5877 024604  001002                           BNE     8$              ;SKIP BMPSAV IF NOT A BMP CODE.
5878 024606  004767  171666                    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
5879                                        ;+
5880                                        ; CHECK FOR XOFF AND XON CHARACTERS.
5881                                        ;-
5882 024612  120227  000023          8$:       CMPB    R2,#23          ;IS IT AN XOFF CHARACTER?.
5883 024616  001406                           BEQ     10$             ;YES; GO REPORT ERROR.
5884 024620  120227  000021                    CMPB    R2,#21          ;NO; IS IT AN XON CHARACTER?.
5885 024624  001403                           BEQ     10$             ;YES; GO REPORT ERROR.
5886 024626  005301                           DEC     R1              ;DECREMENT THE READ COUNT.
5887 024630  001357                           BNE     6$              ;LOOP TO READ THE NEXT CHAR.
5888 024632  000413                           BR      12$             ;GO CHECK FOR ANY UNTESTED ACTIVE LINES.
5889
5890 024634  005267  157212          10$:      INC     ERRNBR          ;SET ERROR NUMBER TO 5107.
5891 024640  016701  000042                    MOV     55$,R1          ;PASS THE LINE NUMBER TO BE REPORTED.
5892 024644  012702  006144                    MOV     #EM5103,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
5893 024650                                    ERROR                   ;                >>>>> ERROR <<<<<.
     024650  104460                                                                       TRAP    C$ERROR
5894
5895                                        ;+
5896                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5897                                        ;-
5898 024652  032767  000100  155330            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5899 024660  001413                           BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5900                                                                   ;DURING THE SOFTWARE QUESTIONS.
5901                                        ;+
5902                                        ; CHECK IF ALL ACTIVE LINES HAVE BEEN TESTED.
5903                                        ;-
5904 024662  005267  000020          12$:      INC     55$             ;INCREMENT LINE NUMBER.
5905 024666  016701  000014                    MOV     55$,R1          ;GET NUMBER OF THE NEXT LINE TO TEST.
5906 024672  005705                           TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5907 024674  001252                           BNE     2$              ;LOOP TO CHECK NEXT LINE.
5908 024676  000404                           BR      60$             ;EXIT TEST.
5909
5910 024700  004767  171766          50$:      JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5911 024704  000401                           BR      60$             ;EXIT THIS TEST.
5912 024706  000000          55$:      .WORD   0                       ;STORAGE FOR LINE NUMBER.
5913 024710  005067  155334          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5914
5915 024714                                    ENDTST
     024714                                                                        L10034:
     024714  104401                                                                       TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - IAUTOA -
                                ;+**********************************************************************
                                ;*                      - IAUTO BIT ACTIVE TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT'S IAUTO FUNCTION BEHAVES CORRECTLY
                                ;*      WHEN ACTIVE, IE IAUTO ASSERTED HIGH.
                                ;*      ALL ACTIVE LINES ARE TESTED INDIVIDUALLY BY FILLING THE FIFO, AND
                                ;*      CHECKING FOR THE PRESENCE OF AT LEAST ONE XOFF(ASCII DC3) CHARACTER
                                ;*      AND ONE XON (ASCII DC1) CHARACTER.
                                ;*      ANY BMP CODES THAT ARE FOUND WILL BE PLACED ON THE BMP CODE QUEUE,
                                ;*      TO BE REPORTED LATER.
                                ;*      THE CHARACTERS ARE TRANSMITTED ON ALL ACTIVE LINES, IN INTERNAL
                                ;*      LOOPBACK MODE.
                                ;*
                                ;--**********************************************************************

024716                                  BGNTST
024716                                                                  T8::
024716                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
024716  012700  000240                                                  MOV     #PRI05,R0
024722  104441                                                          TRAP    C$SPRI
        000010                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024724  012767  000010  155322    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (52)
024732  012767  177777  155310    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
024740  012767  000001  157102    MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
024746  012767  012121  157076    MOV     #5201.,ERRNBR  ;SET ERROR NUMBER TO 5201.
024754  012767  006174  157072    MOV     #EM5201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
024762  012767  013136  157066    MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 5201 <<<<<.
                                ;-
024770  004767  167222                  JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
024774  103402                        BCS     .+6
024776  000167  000400                JMP     60$             ;EXIT TEST IF FATAL ERROR FOUND.

                                ;+
                                ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                ;-
025002  004767  167572                  JSR     PC,INDTPX       ;INITIALISE DATA PATTERN.

                                ;+
                                ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
                                ; SET LPR TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                ;-
025006  016705  155206                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
025012  012700  000206                  MOV     #206,R0         ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
025016  004767  172530                  JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGISTER.
025022  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
025026  004767  172550                  JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
025032  012704  000012                  MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
025036  004767  167220                  JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

                                ;+
                                ; SET UP LOOP FOR ALL ACTIVE LINES.
```

```
                                        ; TEST THE STATE OF THE OAUTO BIT PRIOR TO TRANSMITTING THE DATA PATTERN.
                                        ;  IF THE BIT IS CLEAR, THEN REPORT THE ERROR AND SKIP TRANSMITTING
                                        ;  THE DATA PATTERN ON THE SELECTED LINE.
                                        ; TRANSMIT A 224 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                        ; EMPTY THE FIFO, AND COUNT THE XOFF AND AN XON CHARS FOUND.
                                        ;-
025042  005001                                CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
025044  005067  000330                      CLR     55$             ;CLEAR STORAGE FOR LINE NUMBER.
025050  012767  012122  156774  2$:         MOV     #5202.,ERRNBR   ;SET THE ERROR NUMBER TO 5202.
025056  004767  170306                      JSR     PC,PUFIFO       ;PURGE THE FIFO.
025062  103143                            BCC     50$             ;GO REPORT ERROR IF FIFO DID NOT PURGE.
025064  000241                            CLC                     ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
025066  006005                            ROR     R5              ;ROTATE THE BIT MAP INTO THE CARRY BIT.
025070  103131                            BCC     16$             ;BRANCH IF LINE IS INACTIVE.
                                        ;+
                                        ; TEST THE IAUTO BIT ON THE SELECTED ACTIVE LINE.
                                        ; REPORT ERROR IF IT IS CLEAR.
                                        ;  DO NOT TRANSMIT THE DATA PATTERN ON THE SELECTED LINE.
                                        ;-
025072  005267  156754                      INC     ERRNBR          ;SET ERROR NUMBER TO 5203.
025076  010177  155124                      MOV     R1,@CSRA        ;SELECT LINE TO TEST.
025102  032777  000002  155126              BIT     #BIT1,@LNCTRA   ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
025110  001010                            BNE     4$              ;SKIP ERROR IF IAUTO BIT SET.
025112  012702  006227                      MOV     #EM5202,R2      ;PASS THE CORRECT ERROR MESSAGE.
                                                                    ; "IAUTO BIT FOUND CLEAR ON LINE NN"
025116                                      ERROR                   ;                >>>>> ERROR <<<<<.
025116  104460                                                                                  TRAP    C$ERROR
                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
025120  032767  000100  155062                BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
025126  001525                              BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                    ;DURING THE SOFTWARE QUESTIONS.
025130  000511                              BR      16$             ;SKIP TRANSMITTING DATA PATTERN.
                                        ;+
                                        ; TRANSMIT DATA PATTERN TO FILL THE FIFO, 223 CHARS + 32 XOFF'S + XON.
                                        ;-
025132  005267  156714          4$:         INC     ERRNBR          ;SET ERROR NUMBER TO 5204.
025136  012702  002650                      MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
025142  012703  000337                      MOV     #223.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
025146  004767  167226                      JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
025152  103107                              BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
                                        ;+
                                        ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER PLUS XOFF
                                        ; TO ARRIVE IN THE FIFO.
                                        ;-
025154  005267  156672                      INC     ERRNBR          ;SET ERROR NUMBER TO 5205.
025160  012701  170454                      MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
025164  016702  155036                      MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
025170  004767  172242                      JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
025174  103076                              BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
025176  012704  000012                      MOV     #10.,R4         ;PASS DELAY OF 10 MILLI SECS.
025202  004767  167054                      JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
```

```
                                        ;*
                                        ; READ 256 CHARS FROM THE FIFO, COUNT ANY XOFF OR XON CHARS FOUND.
                                        ;*
025206  005003                            CLR     R3              ;CLEAR XOFF COUNTER.
025210  005004                            CLR     R4              ;CLEAR XON COUNTER.
025212  005267  156634                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5206.
025216  012701  000400                    MOV     #256.,R1        ;INITIALISE THE READ COUNTER.
025222  017702  155002            6$:     MOV     @RBUFA,R2       ;READ CHAR FROM THE FIFO.
025226  100061                          BPL     50$             ;GO REPORT ERROR IF FIFO EMPTY.
                                        ;*
                                        ; CHECK FOR BMP CODE IN THE FIFO.  SAVE ANY FOUND ON THE QUEUE.
                                        ;*
025230  012700  170301                    MOV     #170301,R0      ;SET UP BMP BIT MASK.
025234  040200                          BIC     R2,R0           ;TRY TO CLEAR ALL THE BMP BITS.
025236  001002                          BNE     8$              ;SKIP BMPSAV IF NOT A BMP CODE.
025240  004767  171234                    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
                                        ;*
                                        ; CHECK FOR XOFF AND XON CHARACTERS.
                                        ;*
025244  120227  000023            8$:     CMPB    R2,#23          ;IS IT AN XOFF CHARACTER?.
025250  001001                          BNE     10$             ;NO. BRANCH TO SEE IF IT IS AN XON.
025252  005203                            INC     R3              ;COUNT THE XOFF CHAR.
025254  120227  000021            10$:    CMPB    R2,#21          ;IS IT AN XON CHARACTER?.
025260  001001                          BNE     12$             ;NO. SKIP THE NEXT INSTRUCTION.
025262  005204                            INC     R4              ;COUNT THE XON.
025264  005301            12$:    DEC     R1              ;DECREMENT THE READ COUNT.
025266  001412                          BEQ     13$             ;BRANCH IF ALL CHARACTERS READ.
                                        ;*
                                        ; CHECK IF THE FIFO HAS BEEN EMPTIED BELOW THE HALF LEVEL. IF IT
                                        ; HAS DELAY FOR 1MS TO ALLOW THE XON TO BE GENERATED.
                                        ;*
025270  020127  000176                    CMP     R1,#126.        ;IS THE FIFO LEVEL = 126 ?
025274  001352                          BNE     6$              ;LOOP TO READ THE NEXT CHARACTER IF NOT.
025276  010400                            MOV     R4,R0           ;SAVE THE XON COUNT, ALTHOUGH THERE SHOULDN'T
                                                                ;BE ANY.
025300  012704  000001                    MOV     #1,R4           ;SET THE DELAY TO 1MS.
025304  004767  166752                    JSR     PC,DELAY        ;PERFORM THE DELAY.
025310  010004                            MOV     R0,R4           ;RESTORE THE XON COUNT.
025312  000743                          BR      6$              ;LOOP TO READ THE NEXT CHAR.
                                        ;*
                                        ; VERIFY THAN AT LEAST 1 XOFF AND 1 XON WAS FOUND IN THE FIFO.
                                        ; REPORT ERROR IF NONE WERE FOUND.
                                        ;-
025314  005703                    13$:    TST     R3              ;CHECK XOFF COUNT.
025316  001403                          BEQ     14$             ;GO REPORT ERROR IF NONE FOUND.
025320  020427  000001                    CMP     R4,#1           ;CHECK XON COUNT = 1.
025324  001413                          BEQ     16$             ;SKIP THE ERROR REPORT IF ONE XON WAS FOUND.
025326  005267  156520            14$:    INC     ERRNBR          ;SET ERROR NUMBER TO 5207.
025332  016701  000042                    MOV     55$,R1          ;PASS THE LINE NUMBER TO BE REPORTED.
025336  012702  006144                    MOV     #EM5103,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
                                                                ; "IAUTO BIT BAD ON LINE NN".
025342                                    ERROR                   ;                  >>>>> ERROR <<<<<.
025342  104460                                                                        TRAP    C$ERROR
                                        ;*
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
```

```
6083                                    ;*
6084 025344  032767  000100  154636          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
6085 025352  001413                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6086                                                                ;DURING THE SOFTWARE QUESTIONS.
6087                                    ;*
6088                                    ; CHECK IF ALL ACTIVE LINES HAVE BEEN TESTED.
6089                                    ;*
6090 025354  005267  000020        16$:     INC     55$             ;INCREMENT LINE NUMBER.
6091 025360  016701  000014                 MOV     55$,R1          ;GET NUMBER OF THE NEXT LINE TO TEST.
6092 025364  005705                        TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
6093 025366  001230                        BNE     2$              ;LOOP TO CHECK NEXT LINE.
6094 025370  000404                        BR      60$             ;EXIT TEST.
6095
6096 025372  004767  171274        50$:     JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
6097 025376  000401                        BR      60$             ;EXIT THIS TEST.
6098 025400  000000        55$:     .WORD   0               ;STORAGE FOR LINE NUMBER.
6099 025402  005067  154642        60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6100
6101 025406                        ENDTST
     025406                                                                      L10035:
     025406  104401                                                                    TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - FIFDAT -
                                        ;--*********************************************************************
                                        ;*                          - FIFO VALID DATA TEST -
                                        ;*
                                        ;*      THIS TEST VERIFIES THAT THE DUT IS CAPABLE OF HOLDING 256 VALID
                                        ;*      CHARACTERS IN ITS FIFO.
                                        ;*      THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
                                        ;*      INTERNAL LOOPBACK MODE.
                                        ;*      THE DATA FOUND IN THE FIFO IS COMPARED WITH THE EXPECTED DATA, AND ANY
                                        ;*      DISCREPANCIES ARE REPORTED.
                                        ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                        ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                        ;*      REPORTED LATER.
                                        ;*
                                        ;--*********************************************************************

 025410                                         BGNTST
 025410                                                                                 T9::
 025410                                         SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
 025410 012700  000240                                                                          MOV     #PRIO5,R0
 025414 104441                                                                                  TRAP    C$SPRI
        000011                                   TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 025416 012767  000011  154630                   MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (53)
 025424 012767  177777  154616                   MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 025432 012767  000001  156410                   MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 025440 012767  012265  156404                   MOV     #5301.,ERRNBR   ;SET ERROR NUMBER TO 5301.
 025446 012767  006265  156400                   MOV     #EM5301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERROR >>>>> 5301 <<<<<.
                                        ;-
 025454 004767  166536                          JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 025460 103113                                  BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.
                                        ;+
                                        ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                        ; INITIALISE 256 BYTE DATA PATTERN.
                                        ;-
 025462 004767  167002                          JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
 025466 103110                                  BCC     60$             ;EXIT IF NO ACTIVE LINES FOUND.
 025470 004767  167054                          JSR     PC,INDATP       ;INITIALISE THE DATA PATTERN.
                                        ;+
                                        ; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                        ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                        ;-

                                                ;+
                                                ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
                                                ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
                                                ;-
 025474 012700  000204                           MOV    #204,R0         ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
 025500 004767  172046                          JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
 025504 012700  177670                           MOV    #177670,R0      ;PASS THE LPR CONTENTS.
 025510 004767  172066                          JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
 025514 012704  000012                           MOV    #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
 025520 004767  166536                          JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
 025524 012702  002650                           MOV    #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 025530 012703  000400                           MOV    #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
```

```
6157 025534  005267  156312                    INC     ERRNBR          ;SET ERROR NUMBER TO 5302.
6158 025540  004767  166634                    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
6159 025544  103057                            BCC     50$             ;ABORT TEST IF ERROR FOUND DURING DMA TX.
6160                                          ;+
6161                                          ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6162                                          ; THE FIFO.
6163                                          ;-
6164 025546  005267  156300                    INC     ERRNBR          ;SET ERROR NUMBER TO 5303.
6165 025552  010103                            MOV     R1,R3           ;SAVE THE NUMBER OF THE SELECTED ACTIVE LINE.
6166 025554  012701  170536                    MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
6167 025560  016702  154442                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6168 025564  004767  171646                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6169 025570  103045                            BCC     50$             ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
6170 025572  012704  000005                    MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6171 025576  004767  166460                    JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6172                                  ;+
6173                                  ; READ THE FIFO CHECKING FOR DATA CORRUPTION, REPORT ANY ERRORS FOUND.
6174                                  ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
6175                                  ;-
6176 025602  006303                            ASL     R3              ;MULTIPLY BY 2.
6177 025604  005004                            CLR     R4              ;INITIALISE THE EXPECTED DATA.
6178 025606  016705  154416                    MOV     RBUFA,R5        ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
6179 025612  012767  012270  156232  2$:       MOV     #5304.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6180 025620  011502                            MOV     (R5),R2         ;GET THE ACTUAL DATA FROM THE FIFO.
6181 025622  100030                            BPL     50$             ;ABORT THE TEST IF THE FIFO IS EMPTY.
6182                                  ;+
6183                                  ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6184                                  ;  IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6185                                  ;  ABORT THE TEST.
6186                                  ;-
6187 025624  005267  156222                     INC     ERRNBR          ;SET ERROR NUMBER TO 5305.
6188 025630  004767  166142                    JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
6189 025634  103002                            BCC     4$              ;BRANCH IF NOT A BMP CODE.
6190 025636                                    ERROR                   ;                  >>>>> ERROR 5305 <<<<<.
     025636  104460                                                                              TRAP    C$ERROR
6191
6192 025640  000423                            BR      60$             ;ABORT THIS TEST.
6193
6194 025642  005267  156204            4$:      INC     ERRNBR          ;SET ERROR NUMBER TO 5306.
6195 025646  120402                            CMPB    R4,R2           ;COMPARE THE EXPECTED WITH THE ACTUAL DATA.
6196 025650  001412                            BEQ     8$              ;SKIP ERROR REPORT IF DATA IS OK.
6197 025652  012767  012644  156176             MOV     #ER9002,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6198 025660  012701  006321                     MOV     #EM5302,R1      ;PASS THE MESSAGE TO BE REPORTED.
6199                                           ;REPORT THE ERROR "FIFO BAD, DATA FIELD CORRUPTED"
6200 025664                            6$:     ERROR                   ;                  >>>>> ERROR 5306 <<<<<.
     025664  104460                                                                              TRAP    C$ERROR
6201
6202                                  ;+
6203                                  ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6204                                  ;-
6205 025666  032767  000100  154314             BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
6206 025674  001405                            BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6207                                           ;DURING THE SOFTWARE QUESTIONS.
6208
6209 025676  105204                    8$:      INCB    R4              ;INCREMENT THE EXPECTED DATA.
6210 025700  001344                            BNE     2$              ;LOOP IF NOT DONE.
6211 025702  000402                            BR      60$             ;EXIT
```

```
6212
6213 025704  004767  170762          50$:    JSR     PC,TSABRT       ;ABORT THE TEST, REASON SHOWN BY ERROR NUMBER.
6214 025710  005067  154334          60$:     CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6215
6216 025714                                  ENDTST
     025714                                                                  L10036:
     025714  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FI3QLI -
                                ;+*****************************************************************************
                                ;*                        - FIFO 3/4 LEVEL INACTIVE TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
                                ;*      REMAINS INACTIVE WHILE IT CONTAINS 191 CHARACTERS OR LESS.
                                ;*      THE TEST LOOKS FOR AN XOFF (ASCII DC3) CHARACTER IN THE FIFO.
                                ;*      IF ANY XOFF'S ARE FOUND AN ERROR WILL BE REPORTED AND THE TEST ABORTED.
                                ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                ;*      REPORTED LATER.
                                ;*      THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
                                ;*      INTERNAL LOOPBACK MODE.
                                ;
                                ;--*****************************************************************************

                                        BGNTST
                                                                                        T10::
                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
                                                                                        MOV     #PRIO5,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (54)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                         MOV    #1,ERRTYP       ;SET FATAL ERROR TYPE IN ERROR TABLE.
                                         MOV    ^5401..,ERRNBR  ;SET ERROR NUMBER TO 5401.
                                         MOV    #EM5401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                         MOV    #ER0503,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 5401 <<<<<.
                                ;-
                                        JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
                                        BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.
                                ;+
                                ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                ;-
                                        JSR     PC,FINACT       ;FIND THE NUMBER OF THE FIRST ACTIVE LINE.
                                        BCC     60$             ;EXIT IF NO LINES ARE AVAILABLE.

                                ;+
                                ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                ;-
                                        JSR     PC,INDTPX       ;INITIALISE THE DATA PATTERN.
                                ;+
                                ; TRANSMIT A 191 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                ;-
                                     ;+
                                     ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RX ON THE SELECTED LINE.
                                     ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
                                     ;
                                        MOV     #206,R0         ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
                                        JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
```

```
6272 026020  012700  177670                MOV     #177670,R0      ;PASS THE LPR CONTENTS.
6273 026024  004767  171552                JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
6274 026030  012704  000012                MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
6275 026034  004767  166222                JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6276 026040  012702  002650                MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
6277 026044  012703  000277                MOV     #191.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
6278 026050  004767  166324                JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
6279 026054  103057                       BCC     50$             ;IF ERROR FOUND DURING DMA THEN ABORT TEST.
6280
6281                                    ;+
6282                                    ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6283                                    ; THE FIFO.
6284                                    ;-
6285 026056  005267  155770                INC     ERRNBR          ;SET ERROR NUMBER TO 5402.
6286 026062  012701  170454                MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6287 026066  016702  154134                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6288 026072  004767  171340                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6289 026076  103046                       BCC     50$             ;IF FIFO EMPTY, REPORT ERROR, ABORT THE TEST.
6290 026100  012704  000005                MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6291 026104  004767  166152                JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6292                                    ;+
6293                                    ; READ THE CONTENTS OF THE FIFO.  IF ANY OF THE FOLLOWING CONDITIONS OCCUR
6294                                    ; REPORT THE ERROR AND ABORT THE TEST;
6295                                    ;  FIFO EMPTY TOO SOON.
6296                                    ;  BMP CODE FOUND.
6297                                    ;  XOFF CODE FOUND.
6298                                    ;  EXTRA (192) CHARACTER FOUND IN FIFO.
6299                                    ;-
6300 026110  005004                       CLR     R4              ;CLEAR THE CHARACTER COUNT.
6301 026112  016705  154112                MOV     RBUFA,R5        ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
6302 026116  012767  012433  155726  2$:   MOV     #5403.,ERRNBR   ;SET ERROR NUMBER TO 5403.
6303 026124  011502                       MOV     (R5),R2         ;GET THE ACTUAL DATA FROM THE FIFO.
6304 026126  100032                      BPL     50$             ;FIFO EMPTY, ABORT TEST.
6305 026130  005204                       INC     R4              ;COUNT THE CHARACTER.
6306                                    ;+
6307                                    ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6308                                    ;  IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6309                                    ;  ABORT THE TEST.
6310                                    ;-
6311 026132  005267  155714                INC     ERRNBR          ;SET ERROR NUMBER TO 5404.
6312 026136  004767  165634                JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
6313 026142  103001                       BCC     4$              ;BRANCH IF NOT A BMP CODE.
6314                                       ;REPORT ERROR "BMP CODE FOUND IN FIFO, TEST INVALIDATED".
6315 026144  000421                       BR      8$              ;REPORT THE ERROR AND ABORT THE TEST.
6316                                    ;+
6317                                    ; CHECK IF THE CHARACTER IS AN XOFF.  REPORT THE ERROR IF ONE IS FOUND.
6318                                    ;-
6319
6320 026146  005267  155700          4$:   INC     ERRNBR          ;SET ERROR NUMBER TO 5405.
6321 026152  122702  000023                CMPB    #23,R2          ;CHECK IF THE READ DATA IS AN XOFF.
6322 026156  001003                       BNE     6$              ;BRANCH IF NOT AN XOFF.
6323 026160  012701  006520                MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
6324                                       ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6325 026164  000411                       BR      8$              ;GO REPORT THE ERROR AND ABORT THE TEST.
6326
6327 026166  005267  155660          6$:   INC     ERRNBR          ;SET ERROR NUMBER TO 5406.
6328 026172  020427  000277                CMP     R4,#191.        ;CHECK IF WE HAVE READ ALL THE CHARACTERS.
```

```
 6329 026176  001347                        BNE     2$              ;LOOP BACK TO GET THE NEXT CHARACTER.
 6330 026200  011502                         MOV     (R5),R2         ;TRY TO READ AN EXTRA CHARACTER FROM THE FIFO.
 6331 026202  100006                        BPL     60$             ;EXIT IF NON FOUND.
 6332 026204  012701  006520                 MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
 6333                                        ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
 6334
 6335 026210                        8$:     ERROR                   ;       >>>>> ERRORS 5304 THRU 5306 <<<<<.
      026210  104460                                                                       TRAP    C$ERROR
 6336 026212  000402                        BR      60$             ;EXIT THE TEST.
 6337
 6338                                                                ;       >>>>> ERRORS 5402 AND 5403 <<<<<.
 6339 026214  004767  170452        30$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
 6340 026220  005067  154024        60$:     CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6341
 6342 026224                                ENDTST
      026224                                                                  L10037:
      026224  104401                                                                   TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FI3QLA -
                                ;++*************************************************************************
                                ;*                      - FIFO 3/4 LEVEL ACTIVE TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
                                ;*      BECOMES ACTIVE WHEN THE FIFO CONTAINS > 192 CHARACTERS.
                                ;*      THE TEST COMPARES THE ACTUAL NUMBER OF XOFF (ASCII DC3)
                                ;*      CHARACTERS THAT ARE FOUND IN THE FIFO WITH THE EXPECTED NUMBER.
                                ;*      AN ERROR WILL BE REPORTED, IF THE COUNTS ARE FOUND TO DIFFER.
                                ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                ;*      REPORTED LATER.
                                ;*      THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
                                ;*      INTERNAL LOOPBACK MODE.
                                ;*
                                ;--*************************************************************************

                                        BGNTST
                                                                                T11::
                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
                                                                                MOV     #PRI05,R0
                                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (55)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #5501.,ERRNBR   ;SET ERROR NUMBER TO 5501.
                                         MOV    #EM5501,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERROR >>>>> 5501 <<<<<.
                                        ;-
026272  004767  165720                          JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
026276  103402                                 BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
026300  000167  000414                         JMP     60$             ;EXIT TEST FATAL ERROR FOUND.
                                        ;+
                                        ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                        ;-
026304  004767  166160                          JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
026310  103402                                 BCS     .+6             ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
026312  000167  000402                         JMP     60$             ;EXIT TEST.
                                        ;+
                                        ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                        ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                        ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                        ;-
026316  004767  166256                          JSR     PC,INDTPX       ;INITIALISE DATA PATTERN.
                                        ;+
                                        ; TRANSMIT A 191 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                        ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                        ;-
                                             ;+
                                             ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
                                             ; TRANSMIT THE FIRST 191 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
                                             ;-
026322  005267  155524                 2$:      INC     ERRNBR          ;SET ERROR NUMBER TO 5502.
```

```
6398 026326  012700  000206                MOV    #206,R0          ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
6399 026332  004767  171214                JSR    PC,WTWLNC        ;INITIALISE THE LINE CONTROL REGISTER.
6400 026336  012700  177670                MOV    #177670,R0       ;PASS THE LPR CONTENTS.
6401 026342  004767  171234                JSR    PC,WTWLPR        ;SET THE LPR CONTENTS TO 38.4K BAUD.
6402 026346  012704  000012                MOV    #10.,R4          ;PASS DELAY TIME OF 10 MILLI SECONDS.
6403 026352  004767  165704                JSR    PC,DELAY         ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6404 026356  010105                        MOV    R1,R5            ;COPY THE LINE NUMBER.
6405 026360  012702  002650                MOV    #BUFBAS,R2       ;PASS THE START OF THE DATA PATTERN TO TX.
6406 026364  012703  000277                MOV    #191.,R3         ;PASS THE LENGTH OF THE DATA PATTERN.
6407 026370  004767  166004                JSR    PC,DODMA         ;TRANSMIT THE DATA PATTERN.
6408 026374  103147                        BCC    50$              ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6409
6410                                    ;+
6411                                    ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6412                                    ; THE FIFO.
6413                                    ;-
6414 026376  005267  155450                INC    ERRNBR           ;SET ERROR NUMBER TO 5503.
6415 026402  012701  170454                MOV    #170454,R1       ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6416 026406  016702  153614                MOV    CSRA,R2          ;PASS THE ADDRESS OF THE CSR.
6417 026412  004767  171020                JSR    PC,WAIBIS        ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6418 026416  103136                        BCC    50$              ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6419 026420  012704  000005                MOV    #5,R4            ;PASS DELAY OF 5 MILLI SECS.
6420 026424  004767  165632                JSR    PC,DELAY         ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6421                                    ;+
6422                                    ; TRANSMIT A NULL CHARACTER WHICH WILL CAUSE AN XOFF TO BE GENERATED.
6423                                    ;-
6424 026430  005267  155416                INC    ERRNBR           ;SET ERROR NUMBER TO 5504.
6425 026434  010501                        MOV    R5,R1            ;PASS THE LINE NUMBER.
6426 026436  012702  002650                MOV    #BUFBAS,R2       ;PASS THE START OF THE DATA PATTERN TO TX.
6427 026442  012703  000001                MOV    #1,R3            ;PASS THE NUMBER OF CHARACTERS TO TX.
6428 026446  004767  165726                JSR    PC,DODMA         ;TX A NULL CHARACTER TO CAUSE AN XOFF.
6429 026452  103120                        BCC    50$              ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6430                                    ;+
6431                                    ; WAIT FOR THE DMA TO COMPLETE AND THE LAST CHAR TO ARRIVE IN THE FIFO
6432                                    ;-
6433 026454  005267  155372                INC    ERRNBR           ;SET ERROR NUMBER TO 5505.
6434 026460  012701  170012                MOV    #170012,R1       ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6435 026464  016702  153536                MOV    CSRA,R2          ;PASS THE ADDRESS OF THE CSR.
6436 026470  004767  170742                JSR    PC,WAIBIS        ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6437 026474  103107                        BCC    50$              ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6438 026476  012704  000005                MOV    #5,R4            ;PASS DELAY OF 5 MILLI SECS.
6439 026502  004767  165554                JSR    PC,DELAY         ;WAIT FOR XOFF TO GET INTO THE FIFO.
6440                                    ;+
6441                                    ; INITIALISE THE 256 BYTE DATA PATTERN TO ALL NULLS.
6442                                    ;-
6443 026506  012702  002650                MOV    #BUFBAS,R2       ;INITIALIZE THE DATA PATTERN TO BE
6444 026512  105022                4$:     CLRB   (R2)+            ;  ALL NULLS.
6445 026514  020227  003250                CMP    R2,#BUFMID       ;
6446 026520  103774                        BLO    4$               ;
6447
6448                                    ;+
6449                                    ; TRANSMIT A FURTHER 31 NULL CHARACTERS WHICH WILL CAUSE 31 XOFF'S TO BE
6450                                    ; GENERATED.
6451                                    ;-
6452 026522  005267  155324                INC    ERRNBR           ;SET ERROR NUMBER TO 5506.
6453 026526  010501                        MOV    R5,R1            ;PASS THE LINE NUMBER.
6454 026530  012702  002650                MOV    #BUFBAS,R2       ;PASS THE START OF THE DATA PATTERN TO TX.
```

```
6455 026534  012703  000037                    MOV     #31.,R3         ;PASS THE LENGTH OF THE DATA PATTERN.
6456 026540  004767  165634                   JSR     PC,DODMA         ;TRANSMIT THE DATA PATTERN.
6457 026544  103063                            BCC     50$              ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6458                                        ;+
6459                                        ; WAIT FOR THE XOFF'S AND THE NULL CHARACTERS  TO BE RECEIVED.
6460                                        ; THERE ARE NOW 255 CHARACTERS IN THE FIFO.
6461                                        ;-
6462 026546  005267  155300                    INC     ERRNBR           ;SET ERROR NUMBER TO 5507.
6463 026552  012701  170454                    MOV     #170454,R1       ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6464 026556  016702  153444                    MOV     CSRA,R2          ;PASS THE ADDRESS OF THE CSR.
6465 026562  004767  170650                   JSR     PC,WAIBIS        ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6466 026566  103052                           BCC     50$              ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6467 026570  012704  000005                    MOV     #5,R4            ;PASS DELAY OF 5 MILLI SECS.
6468 026574  004767  165462                   JSR     PC,DELAY         ;WAIT FOR XOFF TO GET INTO THE FIFO.
6469                                        ;+
6470                                        ; READ THE FIFO UNTIL EMPTY, COUNTING THE NUMBER OF XOFF CHARACTERS
6471                                        ; THAT ARE FOUND.
6472                                        ;-
6473 026600  005004                            CLR     R4               ;CLEAR CHARACTER COUNTER.
6474 026602  005003                            CLR     R3               ;CLEAR THE XOFF FOUND COUNTER.
6475 026604  012701  170001                    MOV     #170001,R1       ;INDICATE TO TEST DATA.VALID BIT, TIME-OUT 1MS.
6476 026610  012767  012604  155234  6$:      MOV     #5508.,ERRNBR    ;SET UP ERROR NUMBER EACH TIME AROUND THE LOOP.
6477 026616  016702  153406                    MOV     RBUFA,R2         ;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6478 026622  004767  170610                   JSR     PC,WAIBIS        ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
6479 026626  103032                           BCC     50$              ;GO REPORT ERROR IF FIFO EMPTY.
6480 026630  005204                            INC     R4               ;COUNT THE CHARACTER.
6481                                    ;+
6482                                    ; CHECK FOR BMP CODES IN THE FIFO, ABORT THE TEST IF ANY ARE FOUND.
6483                                    ;  SAVE THE BMP CODE ON THE QUEUE TO BE REPORTED LATER.
6484                                    ;-
6485 026632  005267  155214                    INC     ERRNBR           ;SET ERROR NUMBER TO 5509.
6486 026636  004767  165134                   JSR     PC,CHKBMP        ;CHECK IF WE HAVE GOT A BMP CODE.
6487 026642  103422                           BCS     12$              ;GO REPORT THE ERROR IF WE FOUND A BMP CODE.
6488                                    ;+
6489                                    ; CHECK FOR XOFF CHARACTER.
6490                                    ;-
6491 026644  122702  000023            8$:      CMPB    #23,R2           ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
6492 026650  001001                           BNE     10$              ;BRANCH IF CHARACTER WAS NOT AN XOFF.
6493 026652  005203                            INC     R3               ;INCREMENT XOFF FOUND COUNT.
6494                                    ;+
6495                                    ; CHECK IF ALL THE CHARACTERS INCLUDING THE XON HAVE BEEN REMOVED.
6496                                    ;-
6497 026654  020427  000400            10$:     CMP     R4,#256.         ;CHECK IF WE HAVE REMOVED ALL THE CHARACTERS.
6498 026660  002753                           BLT     6$               ;GO GET THE NEXT CHAR IF WE HAVE NOT FINISHED.
6499
6500                                    ;+
6501                                    ; CHECK IF THE CORRECT NUMBER OF XOFF'S WERE FOUND IN THE FIFO,
6502                                    ; REPORT ERROR IF COUNT IS INCORRECT.
6503                                    ;-
6504 026662  012767  012606  155162            MOV     #5510.,ERRNBR    ;SET UP THE ERROR NUMBER TO 5510.
6505 026670  022703  000040                    CMP     #32.,R3          ;COMPARE EXPECTED XOFF COUNT WITH ACTUAL COUNT.
6506 026674  001411                           BEQ     60$              ;EXIT TEST IF SUCCESS.
6507 026676  012767  011762  155152            MOV     #ERO503,ERRBLK   ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6508 026704  012701  006520                    MOV     #EM5402,R1       ;PASS THE MESSAGE TO BE REPORTED.
6509                                            REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6510 026710                            12$:     ERROR                    ;                  >>>>> ERROR <<<<<.
     026710  104460                                                                        TRAP    C$ERROR
```

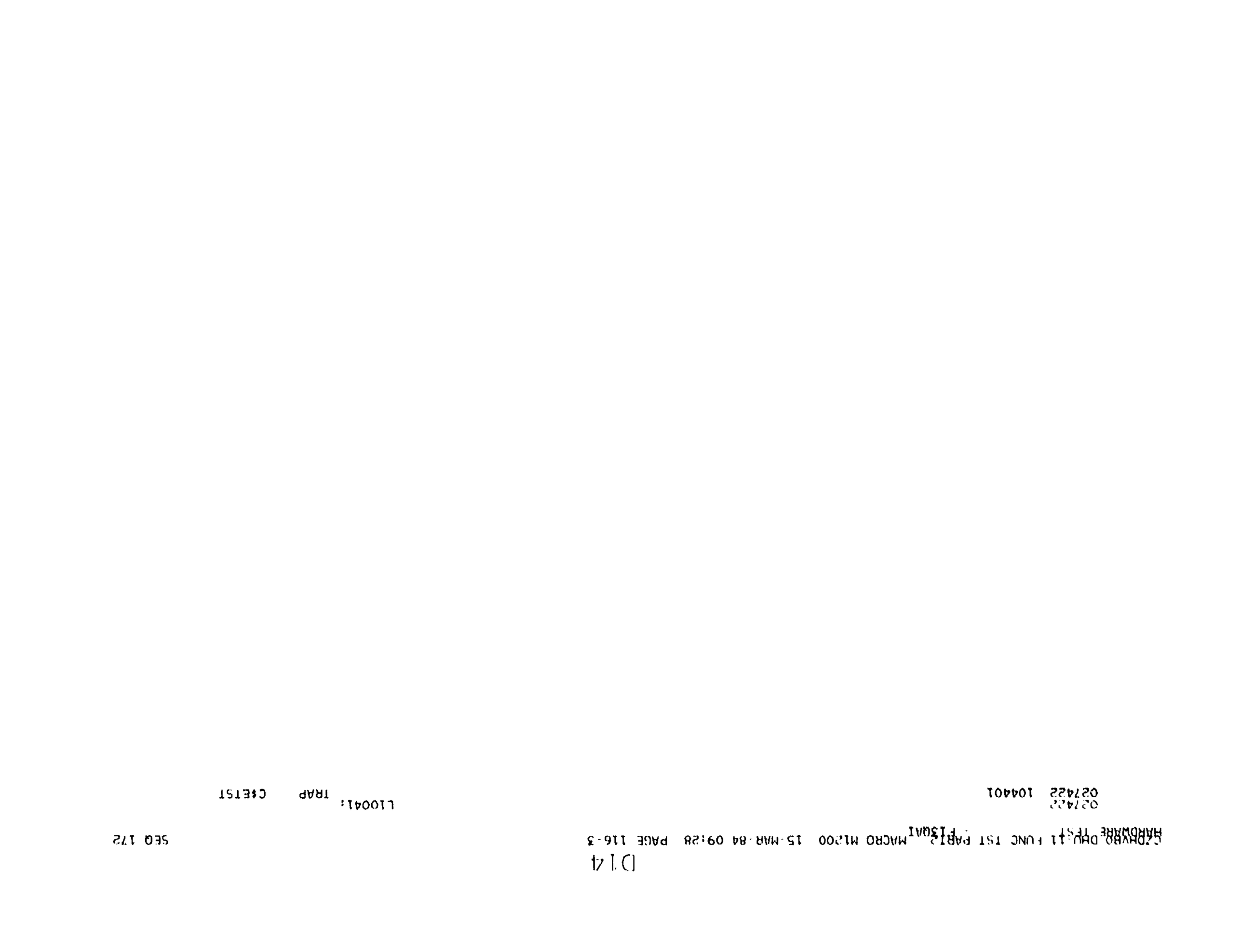
```
 6511 026712  000402                          BR      60$             ;ABORT THE TEST.
 6512
 6513 026714  004767  167752          50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED.  ERROR # SHOWS REASON.
 6514 026720  005067  153324          60$:     CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6515
 6516 026724                          ENDTST
      026724                                                                  L10040:
      026724  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FI3QAI -
                                ;+**********************************************************************
                                ;*                  - FIFO 3/4 ALARM LEVEL ACTIVE/INACTIVE TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
                                ;*      BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.
                                ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                ;*      REPORTED LATER.
                                ;*      THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
                                ;*      INTERNAL LOOPBACK MODE.
                                ;*
                                ;-**********************************************************************

026726                                  BGNTST
026726                                                                  T12::
026726                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
026726  012700  000240                                                  MOV     #PRIO5,R0
026732  104441                                                          TRAP    C$SPRI
000014                                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026734  012767  000014  153312           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (56)
026742  012767  177777  153300           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
026750  012767  000001  155072           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026756  012767  012741  155066           MOV     #5601.,ERRNBR  ;SET ERROR NUMBER TO 5601.
026764  012767  006625  155062           MOV     #EM5601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 5601 <<<<<.
                                ;-
026772  004767  165270                    JSR     PC,CLNRST      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
026776  103402                          BCS     2$             ;SKIP EXITING TEST A SUCCESSFUL RESET.
027000  000167  000412                  JMP     60$            ;EXIT THIS TEST.
027004                          2$:
                                ;+
                                ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                ;-
027004  004767  165460                  JSR     PC,FINACT      ;FIND AN ACTIVE LINE.
027010  103402                          BCS     .+6            ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
027012  000167  000400                  JMP     60$            ;EXIT TEST.
                                ;+
                                ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                ;-
027016  004767  165556                  JSR     PC,INDTPX      ;INITIALISE THE DATA PATTERN.
                                ;+
                                ; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                ;-
                                    ;+
                                    ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
                                    ; TRANSMIT THE FIRST 191 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
                                    ;-
027022  005267  155024                    INC     ERRNBR         ;SET ERROR NUMBER TO 5602.
027026  012700  000206                    MOV     #206,R0        ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
027032  004767  170514                    JSR     PC,WTWLNC      ;INITILAISE THE LINE CONTROL REGISTER.
```

```
6572 027036  012700  177670                MOV     #177670,R0      ;PASS THE LPR CONTENTS.
6573 027042  004767  170534                JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
6574 027046  012704  000012                MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
6575 027052  004767  165204                JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6576 027056  010105                        MOV     R1,R5           ;COPY THE LINE NUMBER.
6577 027060  012702  002650                MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
6578 027064  012703  000277                MOV     #191.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
6579 027070  004767  165304                JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
6580 027074  103146                        BCC     50$             ;EXIT IF ERROR FOUND DURING DMA TX.
6581                                    ;+
6582                                    ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6583                                    ; THE FIFO.
6584                                    ;-
6585 027076  005267  154750                INC     ERRNBR          ;SET ERROR NUMBER TO 5603.
6586 027102  012701  170454                MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6587 027106  016702  153114                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6588 027112  004767  170320                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6589 027116  103135                        BCC     50$             ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
6590 027120  012704  000005                MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6591 027124  004767  165132                JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6592
6593                                    ;+
6594                                    ; TRANSMIT A NULL CHARACTER WHICH WILL CAUSE AN XOFF TO BE GENERATED.
6595                                    ;-
6596 027130  005267  154716                INC     ERRNBR          ;SET ERROR NUMBER TO 5604.
6597 027134  010501                        MOV     R5,R1           ;PASS THE LINE NUMBER.
6598 027136  012702  002650                MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
6599 027142  012703  000001                MOV     #1,R3           ;PASS THE NUMBER OF
6600 027146  004767  165226                JSR     PC,DODMA        ;TX A NULL CHARACTER TO CAUSE AN XOFF.
6601 027152  103117                        BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6602                                    ;+
6603                                    ; WAIT FOR THE XOFF TO BE RECEIVED BEFORE CONTINUING THE TEST.
6604                                    ;-
6605 027154  005267  154672                INC     ERRNBR          ;SET ERROR NUMBER TO 5605.
6606 027160  012701  170012                MOV     #170012,R1      ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6607 027164  016702  153036                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6608 027170  004767  170242                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6609 027174  103106                        BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6610 027176  012704  000005                MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6611 027202  004767  165054                JSR     PC,DELAY        ;WAIT FOR XOFF TO GET INTO THE FIFO.
6612
6613 027206  010577  153014                MOV     R5,@CSRA        ;SELECT THE LINE READY FOR TRANSMISSION.
6614                                    ;+
6615                                    ; READ THREE CHARACTERS, TRANSMIT ONE CHARACTER UNTIL THE FIRST 192 CHARACTERS
6616                                    ; HAVE BEEN READ FROM THE FIFO, IE UNTIL THE HALF LEVEL IS REACHED.
6617                                    ;  THEN READ THE FIFO UNTIL EMPTY.
6618                                    ; COUNT ALL XOFF'S THAT ARE DETECTED.
6619                                    ;-
6620 027212  005005                        CLR     R5              ;CLEAR THE TX FLAG.
6621 027214  005004                        CLR     R4              ;CLEAR THE CHARACTER COUNTER.
6622 027216  012703  000300                MOV     #192.,R3        ;SET UP READ COUNTER FOR THE FIRST 192 CHARS.
6623
6624 027222  012700  000003        4$:     MOV     #3,R0           ;SET READ COUNTER.
6625 027226  012701  170005        6$:     MOV     #170005,R1      ;INDICATE TO TEST DATA_VALID BIT, TIME OUT 5MS.
6626 027232  016702  152772                MOV     RBUFA,R2        ;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6627 027236  004767  170174                JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME OUT.
6628 027242  103046                        BCC     14$             ;EXIT LOOP IF TIME-OUT, FIFO EMPTY.
```

```
6629 027244  005300                        DEC     R0              ;DECREMENT READ COUNTER.
6630 027246  005303                        DEC     R3              ;DECREMENT CHAR COUNTER.
6631 027250  003002                        BGT     8$              ;SKIP DISBL'G TX IF FIRST 192 CHARS NOT READ.
6632 027252  052705  100000                BIS     #BIT15,R5       ;DISABLE ANY FURTHER TRANSMISSIONS.
6633                                ;*
6634                                ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6635                                ;  IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6636                                ;  ABORT THE TEST.
6637                                ;*
6638 027256  012767  012746  154566  8$:    MOV     #5606.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6639 027264  004767  164506                JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
6640 027270  103446                        BCS     16$             ;GO REPORT ERROR AND ABORT TEST IF BMP FOUND.
6641                                ;*
6642                                ; CHECK FOR XOFF CHARACTER.  IF ONE IS FOUND, COUNT IT.
6643                                ; TRANSMIT A NULL CHARACTER UNTIL THE FIRST 192 CHARS HAVE BEEN READ.
6644                                ;*
6645 027272  122702  000023          10$:    CMPB    #23,R2          ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
6646 027276  001001                        BNE     12$             ;BRANCH IF CHARACTER WAS NOT AN XOFF.
6647 027300  005204                        INC     R4              ;INCREMENT THE XOFF CHAR FOUND COUNTER.
6648
6649 027302  005700                  12$:  TST     R0              ;CHECK READ COUNT, TO SEE IF A CHAR CAN BE TX.
6650 027304  001350                        BNE     6$              ;BRANCH IF 3 CHARS HAVE NOT YET BEEN READ.
6651 027306  005705                        TST     R5              ;CHECK THE TRANSMISSION ENABLED FLAG.
6652 027310  100744                        BMI     4$              ;SKIP TRANSMITTING A CHARACTER IF TX DISABLED.
6653 027312  112777  000000  152714        MOVB    #0,@#DATA       ;TX A NULL CHARACTER.
6654 027320  010446                        MOV     R4,-(SP)        ;SAVE THE XOFF COUNT ON THE STACK.
6655                                ;*
6656                                ; WAIT FOR THE CHARACTER TO BE RECEIVED BEFORE CONTINUING THE TEST.
6657                                ;*
6658 027322  005267  154524                INC     ERRNBR          ;SET ERROR NUMBER TO 5607.
6659 027326  012701  170012                MOV     #170012,R1      ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6660 027332  016702  152670                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6661 027336  004767  170074                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6662 027342  103023                        BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6663 027344  012704  000005                MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6664 027350  004767  164706                JSR     PC,DELAY        ;WAIT FOR XOFF TO GET INTO THE FIFO.
6665 027354  012604                        MOV     (SP)+,R4        ;RESTORE THE XOFF COUNT.
6666 027356  000721                        BR      4$              ;GO RESET THE READ COUNT AND GET NEXT CHAR.
6667
6668                                ;*
6669                                ; CHECK IF THE CORRECT NUMBER OF XOFF'S WERE FOUND IN THE FIFO
6670                                ; REPORT ERROR IF COUNT IS INCORRECT.
6671                                ;*
6672 027360  012767  012750  154464  14$:    MOV     #5608.,ERRNBR   ;SET ERROR NUMBER TO 5608.
6673 027366  020427  000077                CMP     R4,#63.         ;COMPARE THE EXPECTED AND ACTUAL XOFF COUNTS.
6674 027372  001411                        BEQ     60$             ;EXIT TEST IF SUCCESS.
6675 027374  012767  011762  154454        MOV     #ER0503,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6676 027402  012701  006520                MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
6677                                ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6678 027406                          16$:  ERROR                   ;                  >>>>> ERROR <<<<<
     027406  104460                                                                        TRAP    C$ERROR
6679 027410  000402                        BR      60$             ;EXIT THIS TEST.
6680
6681 027412  004767  167254          50$:  JSR     PC,TSABRT       ;REPORT TEST ABORTED.  ERROR # INDICATES FAULT.
6682 027416  005067  152626          60$:  CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6683
6684 027422                                ENDTST
```



```
027422                                                          L10041:
027422  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - FIHAVL -
                                        ;+**********************************************************************
                                        ;*                  - FIFO HALF LEVEL ACTIVE/INACTIVE TEST -
                                        ;*
                                        ;*      THIS TEST CHECKS THAT THE DUT'S FIFO HALF LEVEL ALARM SYSTEM
                                        ;*      BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.
                                        ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                        ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                        ;*      REPORTED LATER.
                                        ;*      THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
                                        ;*      INTERNAL LOOPBACK MODE.
                                        ;*
                                        ;-**********************************************************************

027424                                          BGNTST
027424                                                                          T13::
027424                                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
027424  012700  000240                                                                  MOV     #PRIO5,R0
027430  104441                                                                        TRAP    C$SPRI
        000015                                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027432  012767  000015  152614                  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.                (57)
027440  012767  177777  152602                  MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
027446  012767  000001  154374                  MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027454  012767  013105  154370                  MOV     #5701.,ERRNBR   ;SET ERROR NUMBER TO 5701.
027462  012767  006702  154364                  MOV     #EM5701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
027470  012767  011762  154360                  MOV     #ER0503,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERROR >>>>> 5701 <<<<<.
                                        ;-
027476  004767  164514                          JSR     PC,CLNRST       ;RESET THE DHU 11, REPORT ANY ERRORS FOUND.
027502  103402                                 BCS     2$              ;SKIP EXITING TEST A SUCCESSFUL RESET.
027504  000167  000364                         JMP     60$             ;EXIT THIS TEST.
027510                                  2$:
                                        ;+
                                        ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                        ;-
027510  004767  164754                          JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
027514  103167                                 BCC     60$             ;EXIT IF NO ACTIVE LINES AVAILABLE.
                                        ;+
                                        ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                        ;  ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
                                        ;  NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
                                        ;-
027516  004767  165056                          JSR     PC,INDTPX       ;INITIALISE THE DATA PATTERN.
                                        ;+
                                        ; FILL THE FIFO AND THE UART'S 3 CHAR BUFFER BY TRANSMITTING 225 CHARS
                                        ; (IE 225 + 34 XOFF'S). TRANSMIT DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                        ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                        ;-
                                            ;+
                                            ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
                                            ; TRANSMIT THE 225 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
                                            ;-
027522  005267  154324                              INC     ERRNBR          ;SET ERROR NUMBER TO 5702.
027526  004767  167014                          JSR     PC,SETPAR       ;SET UP PARAMETERS FOR TRANSMISSION.
```

```
6740 027532  012700  000341                    MOV     #225.,R0        ;PASS LENGTH OF DATA PATTERN.
6741 027536  004767  167242                    JSR     PC,TXDATP       ;TRANSMIT DATA PATTERN.
6742 027542  103152                            BCC     50$             ;EXIT IF ERROR FOUND DURING TX.
6743 027544  010105                            MOV     R1,R5           ;COPY THE LINE NUMBER.
6744                                        ;+
6745                                        ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6746                                        ; THE FIFO.
6747                                        ;-
6748 027546  005267  154300                    INC     ERRNBR          ;SET ERROR NUMBER TO 5703.
6749 027552  004767  167734                    JSR     PC,WAITTX       ;WAIT FOR TRANSMISSION TO COMPLETE.
6750 027556  103144                            BCC     50$             ;GO REPORT ERROR IF TX FAILED TO COMPLETE.
6751                                    ;+
6752                                    ; READ THE FIRST 130 CHARACTERS FROM THE FIFO, IF ANY XON'S ARE FOUND
6753                                    ; REPORT THE ERROR. IF ANY BMP CODES ARE FOUND THEN SAVE THEM ON THE QUEUE
6754                                    ; AND ABORT THE TEST.
6755                                    ;-
6756 027560  005267  154266                    INC     ERRNBR          ;SET ERROR NUMBER TO 5704.
6757 027564  012700  000202                    MOV     #130.,R0        ;PASS THE NUMBER OF CHARS TO READ.
6758 027570  004767  166060                    JSR     PC,READBX       ;READ THE FIRST 130 CHARS FROM THE FIFO.
6759 027574  103135                            BCC     50$             ;GO REPORT ERROR IF BMP CODE FOUND.
6760 027576  005267  154250                    INC     ERRNBR          ;SET ERROR NUMBER TO 5705.
6761 027602  005701                            TST     R1              ;CHECK IF AN XON WAS FOUND.
6762 027604  001125                            BNE     30$             ;GO REPORT ERROR IF AN XON WAS FOUND.
6763
6764                                        ;+
6765                                        ; TRANSMIT A NULL CHARACTER (WHICH CAUSES AN XOFF TO BE GENERATED).
6766                                        ;-
6767 027606  010577  152414                    MOV     R5,@CSRA        ;SELECT THE LINE READY FOR TRANSMISSION.
6768 027612  112777  000000  152414            MOVB    #0,@TDATA       ;TRANSMIT A NULL CHARACTER.
6769 027620  005267  154226                    INC     ERRNBR          ;SET ERROR NUMBER TO 5706.
6770 027624  004767  167662                    JSR     PC,WAITTX       ;WAIT FOR TX TO COMPLETE.
6771 027630  103117                            BCC     50$             ;GO REPORT ERROR IF TX DID NOT COMPLETE.
6772                                    ;+
6773                                    ; READ THREE CHARACTERS, TO CAUSE THE XON TO BE GENERATED.
6774                                    ;-
6775 027632  005267  154214                    INC     ERRNBR          ;SET ERROR NUMBER TO 5707.
6776 027636  012700  000003                    MOV     #3,R0           ;SET THE READ COUNT TO 3.
6777 027642  004767  166006                    JSR     PC,READBX       ;READ 3 CHARACTERS FROM THE FIFO.
6778 027646  103110                            BCC     50$             ;GO REPORT ERROR IF FIFO EMPTY.
6779 027650  005267  154176                    INC     ERRNBR          ;SET ERROR NUMBER TO 5708.
6780 027654  005701                            TST     R1              ;CHECK IF AN XON WAS FOUND.
6781 027656  001102                            BNE     40$             ;GO REPORT ERROR IF AN XON WAS FOUND.
6782                                    ;+
6783                                    ; TRANSMIT 62 CHARACTERS TO BRACKET THE XON AND FILL THE FIFO WITH 191 CHARS.
6784                                    ;-
6785 027660  012700  000076                    MOV     #62.,R0         ;PASS LENGTH OF DATA PATTERN.
6786 027664  010501                            MOV     R5,R1           ;PASS THE LINE NUMBER.
6787 027666  005267  154160                    INC     ERRNBR          ;SET ERROR NUMBER TO 5709.
6788 027672  004767  167106                    JSR     PC,TXDATP       ;TRANSMIT DATA PATTERN.
6789 027676  103074                            BCC     50$             ;EXIT IF ERROR FOUND DURING TX.
6790
6791                                    ;+
6792                                    ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6793                                    ; THE FIFO.
6794                                    ;-
6795 027700  005267  154146                    INC     ERRNBR          ;SET ERROR NUMBER TO 5710.
6796 027704  004767  167602                    JSR     PC,WAITTX       ;WAIT FOR TX TO COMPLETE.
```

```
6797 027710  103067                    BCC     50$             ;GO REPORT ERROR IF TX FAILED TO COMPLETE.
6798                                ;+
6799                                ; READ THE FIRST 126 CHARACTERS.
6800                                ; READ THE NEXT 4 CHARACTERS AND CHECK IF THEY ARE IN THE FOLLOWING ORDER
6801                                ; NULL, XOFF, XON, NULL.
6802                                ;-
6803 027712  005267  154134            INC     ERRNBR          ;SET ERROR NUMBER TO 5711.
6804 027716  012700  000176            MOV     #126.,R0        ;SET UP READ COUNTER.
6805 027722  004767  165726            JSR     PC,READBX       ;READ THE FIRST 126 CHARS.
6806 027726  103060                    BCC     50$             ;GO REPORT THE ERROR IF FIFO EMPTY.
6807 027730  005267  154116            INC     ERRNBR          ;SET ERROR NUMBER TO 5712.
6808 027734  005701                    TST     R1              ;CHECK IF AN XON WAS FOUND.
6809 027736  001052                    BNE     40$             ;GO REPORT ERROR IF AN XON WAS FOUND.
6810 027740  005267  154106            INC     ERRNBR          ;SET ERROR NUMBER TO 5713.
6811 027744  012701  006520            MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
6812 027750  016703  152254            MOV     RBUFA,R3        ;GET THE RECEIVER BUFFER ADDRESS.
6813 027754  011302                    MOV     (R3),R2         ;READ THE NULL CHARACTER FROM THE FIFO.
6814 027756  120227  000000            CMPB    R2,#000         ;CHECK IF IT IS A NULL CHARACTER.
6815 027762  001040                    BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6816 027764  005267  154062            INC     ERRNBR          ;SET ERROR NUMBER TO 5714.
6817 027770  011302                    MOV     (R3),R2         ;READ THE XOFF FROM THE FIFO.
6818 027772  120227  000023            CMPB    R2,#23          ;CHECK IF THE READ CHAR IS AN XOFF.
6819 027776  001032                    BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6820 030000  011302                    MOV     (R3),R2         ;READ THE XON FROM THE FIFO.
6821 030002  005267  154044            INC     ERRNBR          ;SET ERROR NUMBER TO 5715.
6822 030006  120227  000021            CMPB    R2,#21          ;CHECK IF THE READ CHARACTER IS AN XON.
6823 030012  001024                    BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6824 030014  005267  154032            INC     ERRNBR          ;SET ERROR NUMBER TO 5716.
6825 030020  011302                    MOV     (R3),R2         ;READ THE NULL CHARACTER FROM THE FIFO.
6826 030022  120227  000000            CMPB    R2,#000         ;CHECK IF IT IS A NULL CHARACTER.
6827 030026  001016                    BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6828
6829                                ;+
6830                                ; READ THE REMAINING CHARACTERS FROM THE FIFO.
6831                                ;-
6832 030030  012700  000075        6$:     MOV     #61.,R0         ;SET UP READ COUNTER.
6833 030034  005267  154012            INC     ERRNBR          ;SET ERROR NUMBER TO 5717.
6834 030040  004767  165610            JSR     PC,READBX       ;READ THE FIRST 61 CHARS.
6835 030044  103011                    BCC     50$             ;GO REPORT THE ERROR IF FIFO EMPTY.
6836 030046  005267  154000            INC     ERRNBR          ;SET ERROR NUMBER TO 5718.
6837 030052  005701                    TST     R1              ;CHECK IF AN XON WAS FOUND.
6838 030054  001003                    BNE     40$             ;GO REPORT ERROR IF AN XON WAS FOUND.
6839 030056  000406                    BR      60$             ;EXIT THE TEST.
6840 030060  012701  006520        30$:    MOV     #EM5402,R1      ;SET UP THE MESSAGE
6841                                                            ; "FIFO ALARM SIGNAL DEFECTIVE".
6842 030064                        40$:    ERROR                   ;                >>>>> ERROR <<<<<
     030064  104460                                                                     TRAP    C$ERROR
6843 030066  000402                    BR      60$             ;EXIT THE TEST.
6844
6845 030070  004767  166576        50$:    JSR     PC,TSTABRT      ;REPORT TEST ABORTED.  ERROR # INDICATES FAULT.
6846 030074  005067  152150        60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6847
6848 030100                            ENDTST
     030100                                                                 L10042:
     030100  104401                                                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXTIMER -
                                ;++ ******************************************************************************
                                ;*                          - RXTIMER REG TEST -
                                ;*      THIS TEST VERIFIES THAT THE RXTIMER DELAYS ANY RX-INTS BY THE
                                ;*      REQUESTED AMOUNT AND THAT WHEN THE RXFIFO IS MORE THAN 3/4 FULL
                                ;*      THE RXTIMER VALUE IS IGNORED AND AN INTERRUPT OCCURS IMMEDIATELY.
                                ;*       DUE TO THE DIFFERANCES IN LTC HANDLING OF DIFFERENT VERSIONS OF
                                ;*      THE DRS AND LTC AVAILABILITY ON DIFFERENT PDP-11 MACHINES THE
                                ;*      RX-INT CAN ONLY BE TIMED TO WITHIN  +/- 20%  OF THE RXTIMER VALUE.
                                ;*       THIS TEST IS PERFORMED IN INTERNAL LOOPBACK ON THE FIRST ACTIVE LINE.
                                ;*
                                ;-- ******************************************************************************

030102                                  BGNTST
030102                                                                          T14::
030102                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
030102  012700  000240                                                        MOV     #PRIO5,R0
030106  104441                                                                 TRAP    C$SPRI
        000016                           TNUM = TNUM + 1        ;INCREMENT ASSEMBLY TIME TEST COUNTER
030110  012767  000016  152136           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.
030116  012767  177777  152124           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
030124  012767  000001  153716           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030132  012767  013251  153712           MOV     #5801.,ERRNBR  ;SET THE ERROR NUMBER TO 5801.
030140  012767  006757  153706           MOV     #EM5801,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030146  005067  152100                   CLR     EXOERR         ;CLEAR THE "EXIT ON ERROR" FLAG.
                                ;+
                                ; RESET THE DEVICE AND LEAVE THE SELFTEST CODES IN THE FIFO.
                                ;-
030152  004767  166124                  JSR     PC,RESETT       ;RESET THE DUT.
030156  103402                          BCS     .+6             ;CONTINUE IF FIFO PURGED
030160  000167  000732                  JMP     60$             ;REPORT THE RESET FAILURE.
                                ;+
                                ; SET UP THE INTERRUPT SERVICE ROUTINE THAT DECTECTS THE RX-INT.
                                ;-
030164                                  SETVEC  RXVECA,#RXDECT,#PRIO6
030164  012746  000300                                                        MOV     #PRIO6,-(SP)
030170  012746  017702                                                        MOV     #RXDECT,-(SP)
030174  016746  152014                                                        MOV     RXVECA,-(SP)
030200  012746  000003                                                        MOV     #3,-(SP)
030204  104437                                                              TRAP    C$SVEC
030206  062706  000010                                                        ADD     #10,SP
030212                                  SETPRI  #PRIO4          ;ALLOW DEVICE INTERRUPTS.
030212  012700  000200                                                        MOV     #PRIO4,R0
030216  104441                                                              TRAP    C$SPRI
030220  005067  152042                   CLR     RXINTC         ;CLEAR THE RX-INT COUNT.
                                ;+
                                ; ENABLE RX-INTS AND WAIT FOR ONE TO OCCUR.
                                ;-
030224  004767  166224                  JSR     PC,RXIE1        ;ENABLE RX-INTS.
030230  012704  000005                   MOV     #5,R4          ;SET THE DELAY OF 5 MILLI SECS.
030234  004767  164022                  JSR     PC,DELAY        ;DELAY WHILE THE INT OCCURS.
030240  004767  166150                  JSR     PC,RXIE0        ;DISABLE RX INTS.
030244  005267  153602                   INC     ERRNBR         ;SET THE ERROR NUMBER TO 5802.
030250  005767  152012                   TST     RXINTC         ;TEST IF AN INTERRUPT OCCURED.
030254  001002                          BNE     .+6             ;CONTINUE IF AN INTERRUPT OCCURED.
030256  000167  000630                  JMP     50$             ;REPORT THE ERROR IF NO INTERRUPT.
                                ;+
```

```
                                        ; SET INTERNAL LOOPBACK ON THE FIRST ACTIVE LINE AND ENABLE RECIEVERS.  SET UP
                                        ; THE LPR'S TO 38.4K BAUD, 8 BITS/CHARA, ODD PARITY, 2 STOP BITS.
                                        ;-
030262  004767  164202                          JSR     PC,FINACT       ;FIND AN ACTIVE LINE FOR THIS TEST.
030266  103402                                 BCS     .+6             ;CONTINUE IF A LINE HAS BEEN FOUND.
030270  000167  000622                         JMP     60$             ;EXIT THE TEST IF NO LINES ACTIVE.
030274  012700  000204                          MOV     #204,R0         ;PASS PARAMETER FOR INTERNAL LOOPBACK,
                                                                        ;ENABLE RECIEVERS.
030300  004767  167246                         JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGS.
030304  012700  177670                          MOV     #177670,R0      ;PASS THE LPR CONTENTS.
030310  004767  167266                         JSR     PC,WTWLPR       ;SET THE LPR'S TO 38.4K BAUD.
030314  012704  000012                          MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECS.
030320  004767  163736                         JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
                                        ;+
                                        ; SET UP THE LOOP TO TEST THE RXTIMER WITH DELAYS OF 15,31,63,127 AND 255 MS.
                                        ; DMA 191 CHARACTERS INTO THE FIFO AND THEN ENABLE INTERRUPTS, VERIFY THAT
                                        ; THE INTERRUPT OCCURS WITHIN +/- 20% OF THE RXTIMER VALUE.
                                        ;-
030324  010167  000616                          MOV     R1,70$          ;SAVE THE LINE NUMBER.
030330  012705  000020                          MOV     #16.,R5         ;SET THE FIRST (RXTIMER VALUE + 1).
030334  012767  013253  153510  2$:            MOV     #5803.,ERRNBR   ;SET THE ERROR NUMBER TO 5803.
030342  004767  165022                         JSR     PC,PUFIFO       ;PURGE THE RXFIFO.
030346  103402                                 BCS     .+6             ;CONTINUE IF SUCCESSFUL.
030350  000167  000536                         JMP     50$             ;REPORT THE ERROR IF FIFO FAILED TO PURGE.
030354  016701  000566                          MOV     70$,R1          ;PASS THE LINE NUMBER.
030360  012703  000277                          MOV     #191.,R3        ;PASS THE NUMBER OF CHARS TO DMA.
030364  005267  153462                          INC     ERRNBR          ;SET THE ERROR NUMBER TO 5804.
030370  004767  163726                         JSR     PC,DMABUF       ;PERFORM THE DMA FROM ADDR #BUFBAS.THIS SUBR
                                                                        ;PRODUCES ERRORS >>>>> 5804 THRU 5805 <<<<<.
030374  103402                                 BCS     .+6             ;CONTINUE IF SUCCESSFUL.
030376  000167  000510                         JMP     50$             ;REPORT THE ERROR IF ONE OCCURED.
                                        ;+
                                        ; CALCULATE THE TIME-OUT VALUE FOR THE RX-INT, SET UP THE RXTIMER, AND
                                        ; WAIT FOR THE RX-INT.
                                        ;-
030402  010501                          MOV     R5,R1           ;COPY THE RXTIMER VALUE + 1.
030404  006301                          ASL     R1              ; MULTIPLY BY 4 TO OBTAIN,
030406  006301                          ASL     R1              ;THE TIME-OUT FOR THE RX-INT.
030410  105077  151612                  CLRB    @CSRA           ;CLEAR THE IND.ADDR.REG BITS OF THE CSR READY,
                                                                ;FOR THE WRITE TO THE RXTIMER REG.
030414  010500                          MOV     R5,R0           ;COPY THE RXTIMER VALUE +1.
030416  005300                          DEC     R0              ;GET THE RXTIMER VALUE
030420  110077  151604                  MOVB    R0,@RXTMA       ;LOAD THE RXTIMER REG.
030424  012704  000002                  MOV     #2,R4           ;SET DELAY OF 2 MS.
030430  004767  163626                 JSR     PC,DELAY        ;DELAY TO ALLOW THE RXTIMER VALUE TO UPDATE.
030434  012702  000001                  MOV     #BIT0,R2        ;INDICATE TO TEST BIT0.
030440  010203                          MOV     R2,R3           ;INDICATE TO TEST FOR A "1".
030442  012704  002266                  MOV     #RXINTC,R4      ;PASS ADDR OF WORD TO TEST.
030446  005067  151614                  CLR     RXINTC          ;CLEAR THE RX INT COUNT.
030452  012767  000100  151576          MOV     #BIT6,IESTAT    ;SET THE RX-INT-ENBL BIT IN IESTAT.
030460  016777  151572  151540          MOV     IESTAT,@CSRA    ;ENABLE RX INTS.
030466  004767  164236                 JSR     PC,MSLGET       ;WAIT FOR THE INT TO OCCUR.
030472  103415                          BCS     4$              ;AVOID ERROR REPORT IF THE INTERRUPT OCCURED.
                                        ;+
                                        ; REPORT THE TIME-OUT ERROR. >>>>> 5806 <<<<<.
                                        ;-
030474  010502                          MOV     R5,R2           ;PASS THE RXTIMER VALUE TO,
```

```
6953 030476  005302                        DEC     R2              ;THE ERROR REPORTING ROUTINE.
6954 030500  012701  007265                MOV     #EM5805,R1      ;PASS THE MESSAGE.
6955                                                                ; "RXTIMER BAD, TIME-OUT OCCURED WAITING FOR
6956                                                                ;  THE RX-INT".
6957
6958 030504                                ERRDF   5806,EM5801,ER5801 ;REPORT ERROR  5806.
     030504  104455                                                                    TRAP    C$ERDF
     030506  013256                                                                    .WORD   5806
     030510  006757                                                                    .WORD   EM5801
     030512  012112                                                                    .WORD   ER5801
6959
6960 030514  032767  000100  151466          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
6961 030522  001575                        BEQ     60$             ;EXIT THE TEST IF IT HASN'T.
6962 030524  000454                        BR      6$              ;BRANCH TO TEST ANOTHER RXTIMER VALUE.
6963                                ;+
6964                                ; CHECK THAT THE INTERRUPT OCCURED WITHIN  +/- 20%  OF THE RXTIMER VALUE.
6965                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 5807 <<<<<.
6966                                ;-
6967 030526  012767  013257  153316  4$:     MOV     #5807.,ERRNBR   ;SET THE ERROR NUMBER TO 5807.
6968 030534  010502                        MOV     R5,R2           ;PASS THE RXTIMER VALUE TO,
6969 030536  005302                        DEC     R2              ;THE "CHECK TIME" SUBR.
6970 030540  004767  163302                JSR     PC,CKRXTM       ;CHECK THE TIME TAKEN AND REPORT ANY ERROR.
6971 030544  005767  151502                TST     EXOERR          ;TEST THE "EXIT ON ERROR" FLAG.
6972 030550  001162                        BNE     60$             ;EXIT IF SET.
6973 030552  004767  165636                JSR     PC,RXIE0        ;DISABLE RX-INTS.
6974                                ;+
6975                                ; DMA ANOTHER CHARACTER TO FILL THE FIFO TO THE 75% LEVEL, AND CHECK THAT THE
6976                                ; RX-INT OCCURS IMMEDIATELY.
6977                                ;-
6978 030556  016701  000364                MOV     70$,R1          ;PASS THE LINE NUMBER.
6979 030562  012703  000001                MOV     #1,R3           ;PASS THE NUMBER OF CHARS TO DMA.
6980 030566  005267  153260                INC     ERRNBR          ;SET THE ERROR NUMBER TO 5808.
6981 030572  004767  163524                JSR     PC,DMABUF       ;PERFORM THE DMA FROM ADDR #BUFBAS,THIS SUBR
6982                                                                ;PRODUCES ERRORS >>>>> 5808 THRU 5809 <<<<<.
6983 030576  103145                        BCC     50$             ;REPORT THE ERROR IF ONE OCCURED.
6984 030600  005067  151462                CLR     RXINTC          ;CLEAR THE RX-INT COUNT.
6985 030604  012701  000005                MOV     #5,R1           ;SET THE TIME-OUT TO 5 MS.
6986 030610  012702  002266                MOV     #RXINTC,R2      ;PASS ADDR OF WORD TO TEST.
6987 030614  004767  165634                JSR     PC,RXIE1        ;ENABLE INTERRUPTS.
6988 030620  004767  166612                JSR     PC,WAIBIS       ;WAIT FOR THE INT TO OCCUR.
6989 030624  103414                        BCS     6$              ;AVOID THE ERROR IF AN INTERRUPT OCCURED.
6990                                ;+
6991                                ; REPORT THE ERROR, RX-INT DID NOT OCCUR IMMEDIATLEY.>>>>> 5810 <<<<<.
6992                                ;-
6993 030626  010502                        MOV     R5,R2           ;PASS THE RXTIMER VALUE.
6994 030630  005302                        DEC     R2
6995 030632  012701  007077                MOV     #EM5803,R1      ;PASS THE MESSAGE.
6996                                                                ; "RXTIMER BAD, RX-INT DID NOT OCCUR
6997                                                                ;  IMMEDIATELY WHEN RXFIFO 3/4 FULL".
6998
6999 030636                                ERRDF   5810,EM5801,ER5801 ; REPORT ERROR 5810.
     030636  104455                                                                    TRAP    C$ERDF
     030640  013262                                                                    .WORD   5810
     030642  006757                                                                    .WORD   EM5801
     030644  012112                                                                    .WORD   ER5801
7000
7001 030646  032767  000100  151334          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
```

```
7002 030654  001520                        BEQ     60$             ;EXIT THE TEST IF IT HASN'T.
7003                                ;+
7004                                ; SELECT ANOTHER VALUE FOR THE RXTIMER OR IF ALL VALUES HAVE BEEN TESTED THEN
7005                                ; TEST THE RXTIMER WITH INDEFINATE DELAY SET.
7006                                ;-
7007 030656  004767  165532        6$:     JSR     PC,RXIE0        ;DISABLE INTERRUPTS.
7008 030662  006305                         ASL     R5              ;MULTIPLY (RXTIMER VALUE + 1) BY 2.
7009 030664  020527  000400                 CMP     R5,#256.        ;HAVE ALL VALUES BEEN TESTED ?
7010 030670  003621                        BLE     2$              ;BRANCH AND TEST ANOTHER VALUE IF NOT.
7011                                ;+
7012                                ; VERIFY THAT WHEN RXTIMER VALUE IS 0 THE INTERRUPT IS DELAYED INDEFINITELY,
7013                                ; UNLESS THE RXFIFO IS 75% FULL OR MORE.
7014                                ;-
7015 030672  012767  013263  153152         MOV     #5811.,ERRNBR   ;SET THE ERROR NUMBER TO 5811.
7016 030700  004767  164464                JSR     PC,PUFIFO       ;PURGE THE RXFIFO.
7017 030704  103102                        BCC     50$             ;REPORT THE ERROR IF THE FIFO FAILED TO PURGE.
7018 030706  016701  000234                 MOV     70$,R1          ;PASS THE LINE NUMBER.
7019 030712  012703  000277                 MOV     #191.,R3        ;PASS THE NUMBER OF CHARS TO DMA.
7020 030716  005267  153130                 INC     ERRNBR          ;SET THE ERROR NUMBER TO 5812.
7021 030722  004767  163374                JSR     PC,DMABUF       ;PERFORM THE DMA FROM ADDR #BUFBAS,THIS SUBR
7022                                                                ;PRODUCES ERRORS >>>>> 5812 THRU 5813 <<<<<.
7023 030726  103071                        BCC     50$             ;REPORT THE ERROR IF ONE OCCURED.
7024 030730  012701  001750                 MOV     #1750,R1        ;INDICATE TO TEST BITO WITH TIME OUT OF 1 SEC.
7025 030734  012702  002266                 MOV     #RXINTC,R2      ;PASS THE ADDR OF THE WORD TO TEST.
7026 030740  005067  151322                 CLR     RXINTC          ;CLEAR THE RX-INT COUNT.
7027 030744  105077  151256                 CLRB    @CSRA           ;CLEAR THE IND.ADDR.REG BITS OF THE CSR READY,
7028                                                                ;FOR THE WRITE TO THE RXTIMER REG.
7029 030750  105077  151254                 CLRB    @RXTMA          ;SET THE VALUE 0 IN THE RXTIMER.
7030 030754  012704  000002                 MOV     #2,R4           ;SET THE DELAY OF 2 MS.
7031 030760  004767  163276                JSR     PC,DELAY        ;DELAY TO ALLOW THE RXTIMER VALUE TO UPDATE.
7032 030764  004767  165464                JSR     PC,RXIE1        ;ENABLE RX-INTS.
7033 030770  004767  166442                JSR     PC,WAIBIS       ;WAIT FOR THE INTERRUPT TO OCCUR.
7034 030774  103007                        BCC     8$              ;AVOID THE ERROR IF NO INTERRUPT.
7035                                ;+
7036                                ; REPORT THE ERROR, RX-INT OCCURED WITH RXTIMER VALUE ZERO.>>>>> 5814 <<<<<.
7037                                ;-
7038 030776  012701  007201                 MOV     #EM5804,R1      ;PASS THE MESSAGE,
7039                                                                ; "RXTIMER BAD, RX-INT OCCURED WITH RXTIMER
7040                                                                ;  VALUE ZERO".
7041
7042 031002                                ERRDF   5814,EM5801,ER0503 ; REPORT ERROR  5814.
     031002  104455                                                                    TRAP    C$ERDF
     031004  013266                                                                    .WORD   5814
     031006  006757                                                                    .WORD   EM5801
     031010  011762                                                                    .WORD   ER0503
7043
7044 031012  000441                        BR      60$             ;EXIT THE TEST.
7045                                ;+
7046                                ; VERIFY THAT WHEN THE FIFO IS 75% FULL THE INTERRUPT OCURS IMMEDIATELY.
7047                                ;-
7048 031014  004767  165374        8$:     JSR     PC,RXIE0        ;DISABLE RX-INTS.
7049 031020  012767  013267  153024         MOV     #5815.,ERRNBR   ;SET THE ERROR NUMBER TO 5815.
7050 031026  016701  000114                 MOV     70$,R1          ;PASS THE LINE NUMBER.
7051 031032  012703  000001                 MOV     #1,R3           ;PASS THE NUMBER OF CHARS TO DMA.
7052 031036  004767  163260                JSR     PC,DMABUF       ;PERFORM THE DMA FROM ADDR #BUFBAS,THIS SUBR
7053                                                                ;PRODUCES ERRORS >>>>> 5815 THRU 5816 <<<<<.
7054 031042  103023                        BCC     50$             ;REPORT THE ERROR IF ONE OCCURED.
```

```
 7055 031044  005067  151216                  CLR     RXINTC          ;CLEAR THE RX-INT COUNT.
 7056 031050  012701  000005                  MOV     #5,R1           ;SET THE TIME-OUT TO 5 MS.
 7057 031054  012702  002266                  MOV     #RXINTC,R2      ;PASS ADDR OF WORD TO TEST.
 7058 031060  004767  165370                  JSR     PC,RXIE1        ;ENABLE INTERRUPTS.
 7059 031064  004767  166346                  JSR     PC,WAIBIS       ;WAIT FOR THE INT TO OCCUR.
 7060 031070  103412                          BCS     60$             ;EXIT THE TEST IF AN INTERRUPT OCCURED.
 7061                                ;+
 7062                                ; REPORT THE ERROR, RX-INT DID NOT OCCUR IMMEDIATLEY.>>>>> 5817 <<<<<.
 7063                                ;-
 7064 031072  005002                          CLR     R2              ;PASS THE RXTIMER VALUE.
 7065 031074  012701  007077                  MOV     #EM5803,R1      ;PASS THE MESSAGE.
 7066
 7067 031100                                  ERRDF   5817,EM5801,ER5801 ; REPORT ERROR  5817.
      031100  104455                                                                  TRAP    C$ERDF
      031102  013271                                                                  .WORD   5817
      031104  006757                                                                  .WORD   EM5801
      031106  012112                                                                  .WORD   ER5801
 7068
 7069 031110  000402                          BR      60$             ;EXIT THE TEST.
 7070
 7071 031112  004767  165554          50$:    JSR     PC,TSABRT       ;REPORT NON-RELATED TEST ERROR.
 7072 031116                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      031116  012700  000340                                                          MOV     #PRI07,R0
      031122  104441                                                                  TRAP    C$SPRI
 7073 031124  004767  165264                  JSR     PC,RXIE0        ;DISABLE DEVICE RX-INTS.
 7074 031130                                  CLRVEC  RXVECA          ;CLEAR DOWN THE RX VECTOR.
      031130  016700  151060                                                          MOV     RXVECA,R0
      031134  104436                                                                  TRAP    C$CVEC
 7075 031136  005067  151106                  CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7076 031142                                  EXIT TST
      031142  104432                                                                  TRAP    C$EXIT
      031144  000004                                                                  .WORD   L10043-.
 7077 031146  000000          70$:    .WORD   0               ;LOCAL STORAGE FOR LINE NUMBER USED IN THE TEST.
 7078 031150                                  ENDTST
      031150                                                              L10043:
      031150  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - TXACTF -
                                ;++ ************************************************************************
                                ;*                      - TX ACTION FIFO TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT'S TX-ACTION FIFO CAN CORRECTLY
                                ;*      HOLD 16 TX-ACTIONS.  ONE CHARACTER IS TRANSMIITED ON EACH LINE
                                ;*      USING DMA, THE TX-ACTIONS ARE THEN READ FROM THE FIFO, VERIFING
                                ;*      THAT THEY ARE IN THE CORRECT ORDER AND THAT THERE ARE 16 OF THEM.
                                ;*      THE TEST ALSO VERIFIES THAT THE DUT WILL NOT SEND TX-INTS AFTER
                                ;*      THE TX_ACTION FIFO HAS BEEN EMPTIED.
                                ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK ON ALL LINES.
                                ;*
                                ;-- ************************************************************************

031152                                  BGNTST
031152                                                                  T15::

031152                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
031152  012700  000240                                                  MOV     #PRI05,R0
031156  104441                                                          TRAP    C$SPRI

        000017                           TNUM == TNUM + 1       ;INCREMENT ASSEMBLY TIME TEST COUNTER
031160  012767  000017  151066           MOV     #TNUM,@TNUM    ;SET UP THE TEST NUMBER.
031166  012767  177777  151054           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
031174  012767  000001  152646           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
031202  012767  013415  152642           MOV     #5901.,ERRNBR  ;SET THE ERROR NUMBER TO 5901.
031210  012767  007346  152636           MOV     #EM5901,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
031216  012767  012170  152632           MOV     #ER5901,ERRBLK ;SET THE ERROR REPORTING ROUTINE.

                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE ANY STATUS CODES IN THE FIFO.
                                ; CLEAR THE RX AND TX ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 5901 <<<<<.
                                ;-
031224  004767  162766                    JSR     PC,CLNRST      ;RESET THE DHU-11 REPORT ANY ERRORS FOUND.
031230  103402                          BCS     .+6            ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
031232  000167  000474                  JMP     60$            ;EXIT THE TEST IF FATAL ERROR FOUND.

                                ;+
                                ; SET INTERNAL LOOPBACK O  ALL LINES AND ENABLE RECIEVERS.  SET UP THE
                                ; LPR'S TO 38.4K BAUD, 8 BITS/CHARA, ODD PARITY, 2 STOP BITS.
                                ;-
031236  012705  177777                    MOV     #MAPLNS,R5     ;INDICATE TO SET UP ALL LINES.
031242  012700  00020                     MOV     #204,R0        ;PASS PARAMETER FOR INTERNAL LOOPBACK,
                                                                 ;ENABLE RECIEVERS.
031246  004767  166300                  JSR     PC,WTWLNC      ;INITIALISE THE LINE CONTROL REGS.
031252  012700  177670                    MOV     #177670,R0     ;PASS THE LPR CONTENTS.
031256  004767  166320                  JSR     PC,WTWLPR      ;SET THE LPR'S TO 38.4K BAUD.
031262  012704  000012                    MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI SECS.
031266  004767  162770                  JSR     PC,DELAY       ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

                                ;+
                                ; INITIATE A DMA ON EACH LINE AND WAIT FOR ALL DMA'S TO COMPLEATE.
                                ;-
031272  005001                          CLR     R1             ;PASS THE FIRST LINE NUMBER.
031274  012702  002650                    MOV     #BUFBAS,R2     ;PASS THE START OF THE DATA PATTERN TO TX.
031300  012703  000001                    MOV     #1,R3          ;PASS THE LENGTH OF THE DATA PATTERN.
```

```
7134 031350  012704  000005                    MOV    #5,R4          ;PASS THE DELAY TIME OF 5 MILLI SECS.
7135 031354  005267  152536                    INC    ERRNBR         ;SET THE ERROR NUMBER TO 5902.
7136 031360  004767  163060          2$:       JSR    PC,DODMA       ;TRANSMIT THE DATA PATTERN.
7137 031364  103402                           BCS    .+6            ;CONTINUE IF SUCCESSFUL.
7138 031366  000167  000400                    JMP    50$            ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
7139                                        ;+
7140                                        ; WAIT FOR THE DMA TO COMPLEATE BEFORE INITIATING ANOTHER.
7141                                        ;-
7142 031372  004767  162730                    JSR    PC,DELAY       ;WAIT 5 MILLI SECS FOR THE DMA TO COMPLEATE.
7143 031376  005201                            INC    R1             ;INCREMENT THE LINE NUMBER.
7144 031400  022701  000020                    CMP    #16.,R1        ;BRANCH TO INITIATE ANOTHER DMA IF
7145 031404  001365                           BNE    2$             ;NOT ALL LINES SERVED.
7146                                        ;+
7147                                        ; READ THE TX-ACTIONS FROM THE TX_ACTION FIFO AND VERIFY THAT THEY OCCURED
7148                                        ; IN THE CORRECT ORDER.
7149                                        ;-
7150 031406  005267  152504                    INC    ERRNBR         ;SET THE ERROR NUMBER TO 5903.
7151 031412  012703  007402                    MOV    #EM5902,R3     ;SET THE ERROR MESSAGE TO,
7152                                                                  ; "TX-ACTION RECIEVED FROM THE WRONG LINE".
7153 031416  005001                            CLR    R1             ;CLEAR THE LINE NUMBER.
7154 031420  017702  150646          4$:       MOV    @CSRA,R2       ;READ THE CSR.
7155 031424  100150                           BPL    14$            ;BRANCH TO REPORT ERROR >>5904<<, IF NO TX-ACT.
7156 031426  000302                            SWAB   R2             ;PUT THE TX LINE NUMBER IN THE LOW BYTE.
7157 031430  042702  177760                    BIC    #177760,R2     ;CLEAR THE CLUTTER FROM THE LINE NUMBER.
7158 031434  020201                            CMP    R2,R1          ;COMPARE THE ACTUAL LINE NUMBER OF THE
7159                                                                  ;TX-ACTION WITH THE EXPECTED NUMBER.
7160 031436  001405                           BEQ    6$             ;SKIP THE ERROR REPORT IF CORRECT.
7161 031440                                    ERROR                 ;REPORT THE ERROR  >>>>> 5903 <<<<<.
     031440  104460                                                                    TRAP    C$ERROR
7162 031442  032767  000100  150604            BIT    #BIT06,OPTION  ;EXIT IF EXTENDED ERROR REPORTING HAS NOT
7163 031450  001552                           BEQ    60$            ;BEEN REQUESTED.
7164 031452  005201                    6$:     INC    R1             ;INCREMENT THE EXPECTED LINE NUMBER.
7165 031454  022701  000020                    CMP    #16.,R1        ;BRANCH AND READ ANOTHER TX-ACTION IF,
7166 031460  001357                           BNE    4$             ;NOT ALL LINES HAVE BEEN SERVICED.
7167
7168                                        ;+
7169                                        ; 16 TX-ACTIONS HAVE BEEN READ, THE TX-ACTION BIT SHOULD NOW BE CLEAR.
7170                                        ; CHECK THAT IT IS CLEAR, IF IT ISN'T THEN COUNT THE NUMBER OF EXTRA
7171                                        ; TX-ACTIONS RECIEVED AND REPORT THE ERROR.
7172                                        ;-
7173 031462  005777  150604                    TST    @CSRA          ;READ THE CSR.
7174 031466  100024                           BPL    10$            ;BRANCH IF THE TX-ACTION BIT IS CLEAR, TO
7175                                                                  ;TEST THE TX-INTERRUPTS.
7176 031470  012767  013421  152420            MOV    #5905.,ERRNBR  ;SET THE ERROR NUMBER TO 5905.
7177 031476  012702  000021                    MOV    #17.,R2        ;SET R2 TO BE THE NUMBER OF TX-ACTIONS FOUND.
7178 031502  005777  150564          8$:       TST    @CSRA          ;READ THE CSR.
7179 031506  100123                           BPL    16$            ;BRANCH AND REPORT ERROR IF THE TX-ACTION FIFO,
7180                                                                  ;FINALLY CLEARED.
7181 031510  005202                            INC    R2             ;INCREMENT THE NUMBER OF TX-ACTIONS FOUND.
7182 031512  022702  000145                    CMP    #101.,R2       ;IF LESS THAN 100 TX-ACTIONS HAVE BEEN READ.
7183 031516  001371                           BNE    8$             ;THEN CONTINUE READING THE CSR.
7184                                                                  ;OTHERWISE REPORT THE ERROR.
7185                                        ;+
7186                                        ; REPORT THE ERROR "TX-ACTION FIFO WOULD NOT EMPTY"
7187                                        ;-
7188 031520  005267  152372                    INC    ERRNBR         ;SET THE ERROR NUMBER TO 5906.
7189 031524  012701  007570                    MOV    #EM5904,R1     ;SET THE ERROR MESSAGE TO,
```

```
7190                                                    ;"TX ACTION FIFO WOULD NOT EMPTY"
7191 031464  012767  011762  152364          MOV     #ER0503,ERRBLK  ;SET UP THE ERROR REPORTING ROUTINE.
7192 031472  000513                          BR      18$             ;GO REPORT THE ERROR.
7193                                 ;*
7194                                 ; NOW VERIFY THAT NO TX INTS OCCUR AFTER THE TX ACTION FIFO HAS BEEN EMPTIED,
7195                                 ; OF TX ACTIONS, I.E. NO INTERRUPTS OCCUR WITH THE TX ACTION BIT CLEAR.
7196                                 ;-
7197
7198 031474  005001                 10$:    CLR     R1              ;PASS THE NUMBER OF THE FIRST LINE
7199
7200                                 ;*
7201                                 ; INITIATE A DMA ON ALL LINES AND WAIT FOR ALL DMA'S TO COMPLETE.
7202                                 ;-
7203 031476  012767  013423  152346          MOV     #5907.,ERRNBR   ;SET THE ERROR NUMBER TO 5907.
7204 031504  012702  002650                  MOV     #BUFBAS,R2      ;PASS THE START OF THE DMA PATTERN TO TX.
7205 031510  012703  000001                  MOV     #1,R3           ;PASS THE LENGTH OF THE DATA PATTERN.
7206 031514  004767  162660          12$:    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERNS.
7207 031520  103102                         BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
7208 031522  005201                          INC     R1              ;INCREMENT THE LINE NUMBER.
7209 031524  022701  000020                  CMP     #16.,R1         ;BRANCH TO INITIATE ANOTHER DMA IF,
7210 031530  001371                         BNE     12$             ;ALL LINES NOT SERVED.
7211 031532  012704  000144                  MOV     #100.,R4        ;SET THE DELAY OF 100 MILLI SECS.
7212 031536  004767  162520                  JSR     PC,DELAY        ;WAIT FOR THE DMA'S TO COMPLETE.
7213                                 ;*
7214                                 ; SET UP THE INTERRUPT SERVICE ROUTINE THAT WILL READ THE TX ACTION FIFO
7215                                 ; UNTIL EMPTY AND CHECK FOR ANY SUBSEQUENT INTERRUPTS WITH NO TX ACTION.
7216                                 ; ENABLE TX INTERRUPTS.
7217                                 ;-
7218 031542  005067  150524                  CLR     TXINTC          ;CLEAR THE TX INT COUNTER.
7219 031546  005067  150522                  CLR     TXINTF          ;CLEAR THE TX INT FLAGS.
7220
7221 031552                                  SETVEC  TXVECA,#TXAINT,#PRI06
      031552  012746  000300                                                  MOV     #PRI06,-(SP)
      031556  012746  017734                                                  MOV     #TXAINT,-(SP)
      031562  016746  150430                                                  MOV     TXVECA,-(SP)
      031566  012746  000003                                                  MOV     #3,-(SP)
      031572  104437                                                          TRAP    C$SVEC
      031574  062706  000010                                                  ADD     #10,SP
7222 031600                                  SETPRI  #PRI04          ;ALLOW DEVICE INTERRUPTS.
      031600  012700  000200                                                  MOV     #PRI04,R0
      031604  104441                                                          TRAP    C$SPRI
7223 031606  004767  165444                  JSR     PC,TXIE1        ;ENABLE TX INTERRUPTS.
7224                                 ;*
7225                                 ; WAIT FOR THE INTERRUPTS TO OCCUR
7226                                 ;-
7227 031612  012704  000005                  MOV     #5,R4           ;SET THE DELAY FOR 5 MILLI SECS.
7228 031616  004767  162440                  JSR     PC,DELAY        ;DELAY FOR 5 MS.
7229                                 ;*
7230                                 ; DISABLE INTERRUPTS AND CLEAR DOWN THE INTERRUPT SERVICE ROUTINE.
7231                                 ;-
7232 031622                                  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      031622  012700  000340                                                  MOV     #PRI07,R0
      031626  104441                                                          TRAP    C$SPRI
7233 031630  004767  165362                  JSR     PC,TXIE0        ;DISABLE OUT TX INTERRUPTS
7234 031634                                  CLRVEC  TXVECA          ;CLEAR THE TX INT VECTOR
      031634  016700  150356                                                  MOV     TXVECA,R0
      031640  104436                                                          TRAP    C$CVEC
```

```
                                        ;*
                                        ; VERIFY THAT A TX INTERRUPT OCCURED, AND THAT NO TX INTERRUPT OCCURED WHEN
                                        ; THE TX_ACTION FIFO WAS EMPTY.
                                        ;-
031642  005267  152204                        INC     ERRNBR          ;SET THE ERROR NUMBER TO 5908.
031646  005767  150420                        TST     TXINTC          ;READ THE INTERRUPT COUNT.
031652  001425                              BEQ     50$             ;BRANCH TO REPORT ERROR IF NO INTS OCCURED.
031654  005267  152172                        INC     ERRNBR          ;SET THE ERROR NUMBER TO 5909.
031660  005767  150410                        TST     TXINTF          ;READ THE TX INTERRUPT FLAGS.
031664  100022                              BPL     60$             ;EXIT WITH SUCCESS IF NO INTERRUPT OCCURED
                                                                    ;WITHOUT A CORRESPONDING TX_ACTION.
031666  012701  007654                        MOV     #EM5905,R1      ;SET UP THE ERROR MESSAGE.
                                                                    ; "TX INT OCCURED AFTER TX FIFO HAD
                                                                    ;  BEEN EMPTIED".
031672  012767  011762  152156                MOV     #ER0503,ERRBLK  ;SET THE ERROR ROUTINE IN THE ERROR TABLE.
031700  000410                              BR      18$             ;GO AND REPORT THE ERROR.
                                        ;*
                                        ; ERROR REPORTING.
                                        ;-
031702  010102                        14$:  MOV     R1,R2           ;PASS THE ACTUAL NUMBER OF TX ACTIONS RECIEVED,
                                                                    ;TO THE ERROR ROUTINE.
031704  012767  013420  152140              MOV     #5904.,ERRNBR   ;SET THE ERROR NUMBER TO 5904.
031712  012701  000020                  16$:  MOV     #16.,R1         ;PASS THE EXPECTED NUMBER OF TX-ACTIONS.
031716  012703  007476                        MOV     #EM5903,R3      ;SET THE ERROR MESSAGE TO,
                                                                    ; "INCORRECT NUMBER OF TX-ACTIONS FOUND".
031722                                  18$:  ERROR                   ;REPORT THE ERROR.
        031722  104460                                                      TRAP    C$ERROR
031724  000402                              BR      60$             ;EXIT THE TEST.

031726  004767  164740                  50$:  JSR     PC,TSABRT       ;REPORT NON RELATED TEST ERROR.
031732  005067  150312                  60$:  CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.

031736                                  ENDTST
        031736                                                              L10044:
        031736  104401                                                      TRAP    C$ETST
```

```
7269                                    .SBTTL  HARDWARE TEST           - TXFIFO -
7270                                    ;** ************************************************************************
7271                                    ;*                              - TXFIFO TEST -
7272                                    ;*      THIS TEST IS USED TO VERIFY THAT THE DUT'S TRANSMIT FIFO'S CAN HOLD
7273                                    ;*      64 CHARACTERS AND THAT ONLY ONE TX INTERRUPT OCCURS FOR ALL 64
7274                                    ;*      CHARACTERS.  THE TEST ALSO EXERSISES THE BYTE SWAPPER BY USING
7275                                    ;*      ALTERNATE WORD AND BYTE WRITES TO THE TX FIFO, AS WELL AS CHECKING THAT
7276                                    ;*      THE FIFOSIZE REGISTER REPORTS THE NUMBER OF CHARACTERS IN THE FIFO
7277                                    ;*      CORRECTLY.  ANY BMP CODES FOUND ARE SAVED ON THE QUE TO BE REPORTED
7278                                    ;*      LATER.
7279                                    ;*      THE TEST IS PERFORMED IN INTERNAL LOOPBACK ON ALL ACTIVE LINES.
7280                                    ;** ************************************************************************
7281
7282 031740                                    BGNTST
    031740                                                                          T16::
7283
7284 031740                                    SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
    031740  012700  000240                                                          MOV     #PRIO5,R0
    031744  104441                                                                  TRAP    C$SPRI
7285            000020                          TNUM = TNUM + 1         ;INCREMENT ASSEMBLY TIME TEST COUNTER
7286 031746  012767  000020  150300            MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.
7287 031754  012767  177777  150266            MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
7288 031762  012767  000001  152060            MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7289 031770  012767  013561  152054            MOV     #6001.,ERRNBR   ;SET THE ERROR NUMBER TO 6001.
7290 031776  012767  007754  152050            MOV     #EM6001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
7291 032004  005067  150602                    CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
7292 032010  005067  150236                    CLR     EXOERR          ;CLEAR THE "EXIT ON ERROR" FLAG.
7293 032014  012700  002614                    MOV     #ERCNTB,R0      ;SET UP THE START ADDRESS FOR THE BLOCK.
7294 032020  004767  162214                    JSR     PC,CLR16W       ;CLEAR THE BLOCK OF 16 WORDS.
7295                                    ;+
7296                                    ; RESET THE DUT TO A KNOWN STATE, REMOVE ANY STATUS CODES IN THE FIFO.
7297                                    ; CLEAR THE RX AND TX ENABLE BITS IN THE CSR.
7298                                    ; THIS SUBROUTINE REPORTS ERROR >>>>> 6001 <<<<<.
7299                                    ;-
7300 032024  004767  162166                    JSR     PC,CLNRST       ;RESET THE DHU-11 REPORT ANY ERRORS FOUND.
7301 032030  103402                            BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
7302 032032  000167  000722                    JMP     62$             ;EXIT THE TEST IF FATAL ERROR FOUND.
7303                                    ;+
7304                                    ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES AND ENABLE RECIEVERS.  SET UP THE
7305                                    ; LPR'S TO 38.4K BAUD, 8 BITS/CHARA, ODD PARITY, 2 STOP BITS.  DISABLE
7306                                    ; TRANSMITTERS ON ALL LINES.
7307                                    ;-
7308 032036  012705  177777                    MOV     #MAPLNS,R5      ;INDICATE TO DISABLE ALL LINES.
7309 032042  004767  164760                    JSR     PC,TXDSBL       ;DISABLE TX ON ALL LINES.
7310 032046  016705  150146                    MOV     ACTLNS,R5       ;INDICATE TO SET UP ACTIVE LINES ONLY.
7311 032052  012700  000204                    MOV     #204,R0         ;PASS PARAMETER FOR INTERNAL LOOPBACK,
7312                                                                    ;ENABLE RECIEVERS.
7313 032056  004767  165470                    JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGS.
7314 032062  012700  177670                    MOV     #177670,R0      ;PASS THE LPR CONTENTS.
7315 032066  004767  165510                    JSR     PC,WTWLPR       ;SET THE LPR'S TO 38.4K BAUD.
7316 032072  012704  000012                    MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECS.
7317 032076  004767  162160                    JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7318                                    ;+
7319                                    ; SET UP THE TX INTERRUPT SERVICE ROUTINE AND VECTOR.  THE ROUTINE COUNTS
7320                                    ; THE NUMBER OF INTERRUPTS AND CHECKS FOR ANY INTERRUPTS OCCURING WITH
7321                                    ; NO TX-ACTION.
7322                                    ;
```

```
 7323 032102                            SETVEC  TXVECA,#TXAINT,#PRI06
      032102  012746  000300                                                MOV     #PRI06,-(SP)
      032106  012746  017734                                                MOV     #TXAINT,-(SP)
      032112  016746  150100                                                MOV     TXVECA,-(SP)
      032116  012746  000003                                                MOV     #3,-(SP)
      032122  104437                                                        TRAP    C$SVEC
      032124  062706  000010                                                ADD     #10,SP
 7324 032130  004767  165122                    JSR     PC,TXIE1        ;SET THE TX INT ENABLE BIT IN THE CSR.
 7325                                   ;+
 7326                                   ; WRITE 64 CHARACTERS TO ALL TXFIFO'S USING ALTERNATE WORD/BYTE WRITES
 7327                                   ; TO EXERSISE THE BYTE SWAPPER.  AFTER THE FIRST 9 CHARACTERS HAVE BEEN
 7328                                   ; WRITTEN CHECK THAT THE FIFOSIZE REGISTER SHOWS THE CORRECT NUMBER OF
 7329                                   ; FREE BYTES IN THE FIFO.
 7330                                   ;-
 7331 032134  016705  150060                    MOV     ACTLNS,R5       ;SET UP THE ACTIVE LINE BIT MAP.
 7332 032140  005001                            CLR     R1              ;SET UP THE FIRST LINE NUMBER
 7333
 7334 032142  000241                    2$:     CLC                     ;CLEAR THE CARRY BIT READY FOR THE ROTATION.
 7335 032144  006005                            ROR     R5              ;ROTATE THE ACTIVE LINE BIT MAP INTO THE CARRY.
 7336 032146  103067                            BCC     10$             ;AVOID TESTING THIS LINE IF ITS INACTIVE.
 7337 032150  110177  150052                    MOVB    R1,@CSRA        ;LOAD THE LINE NUMBER OF THE OUT INTO THE CSR.
 7338 032154  010103                            MOV     R1,R3           ;INITIALISE THE DATA PATTERN FOR THIS LINE BY,
 7339 032156  000303                            SWAB    R3              ;PUTTING THE LINE NUMBER IN THE HIBYTE AND
 7340                                                                   ;CLEARING THE LOBYTE.
 7341                                   ;+
 7342                                   ; LOAD 9 CHARACTERS INTO THE TXFIFO AND CHECK THE FIFOSIZE REGISTER.
 7343                                   ;-
 7344 032160  012700  000003                    MOV     #3,R0           ;LOOP COUNT.
 7345 032164  010377  150044            4$:     MOV     R3,@#DATA       ;MOVE A WORD OF DATA INTO THE FIFO.
 7346 032170  105203                            INCB    R3              ;INCREMENT THE LOBYTE OF THE DATA PATTERN.
 7347 032172  110377  150036                    MOVB    R3,@#DATA       ;MOVE A BYTE OF DATA INTO THE FIFO.
 7348 032176  062703  000401                    ADD     #401,R3         ;INCREMENT THE HIGH AND LOW BYTE OF THE DATA.
 7349 032202  005300                            DEC     R0              ;DECREMENT THE LOOP COUNT.
 7350 032204  001367                            BNE     4$              ;BRANCH IF NOT ALL 9 CHARACTERS WRITTEN.
 7351 032206  005002                            CLR     R2              ;CLEAR THE UPPER AND LOWER BYTE OF R2.
 7352 032210  117702  150020                    MOVB    @#SLSA,R2       ;READ THE FIFOSIZE REGISTER.
 7353 032214  122702  000067                    CMPB    #55.,R2         ;COMPARE THE EXPECTED SIZE WITH THE ACTUAL.
 7354 032220  001425                            BEQ     6$              ;AVOID THE ERROR REPORT IF THE SIZE IS CORRECT.
 7355                                   ;+
 7356                                   ; REPORT THE ERROR, INCORRECT VALUE. >>>>> 6002 <<<<<.
 7357                                   ;-
 7358 032222  010104                            MOV     R1,R4           ;PASS THE LINE NUMBER TO THE ERROR ROUTINE.
 7359 032224  012701  000067                    MOV     #55.,R1         ;PASS THE EXPECTED FIFO SIZE.
 7360 032230  010346                            MOV     R3,-(SP)        ;SAVE THE DATA PATTERN.
 7361 032232  012767  013562  151612            MOV     #6002.,ERRNBR   ;SET THE ERROR NUMBER TO 6002.
 7362 032240  012703  010000                    MOV     #EM6002,R3      ;PASS THE MESSAGE,
 7363                                                                   ; "INCORRECT VALUE IN FIFOSIZE REGISTER".
 7364 032244  012767  012276  151604            MOV     #ER6001,ERRBLK  ;SET THE ERROR REPORTING ROUTINE.
 7365 032252                                    ERROR
      032252  104460                                                        TRAP    C$ERROR
 7366 032254  012603                            MOV     (SP)+,R3        ;RESTORE THE PATTERN.
 7367 032256  010401                            MOV     R4,R1           ;RESTORE THE LINE NUMBER.
 7368 032260  032767  000100  147722            BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
 7369 032266  001002                            BNE     6$              ;CONTINUE IF IT HAS.
 7370 032270  000167  000444                    JMP     60$             ;EXIT THE TEST IF IT HASN'T.
 7371                                   ;+
 7372                                   ; CONTINUE FILLING UP THE FIFO UNTIL 64 CHARACTERS HAVE BEEN LOADED.
```

```
7373                                    ;-
7374 032274  012700  000022            6$:     MOV     #18.,R0         ;SET UP THE OUTER LOOP.
7375 032300  010377  147730            8$:     MOV     R3,@#FDATA      ;MOVE A WORD OF DATA INTO THE FIFO.
7376 032304  105203                             INCB    R3              ;INCREMENT THE LOBYTE OF THE DATA PATTERN.
7377 032306  110377  147722                     MOVB    R3,@#FDATA      ;MOVE A BYTE OF DATA INTO THE FIFO.
7378 032312  062703  000401                     ADD     #401,R3         ;INCREMENT THE HIGH AND LOW BYTE OF THE DATA.
7379 032316  005300                             DEC     R0              ;DECREMENT THE LOOP COUNT.
7380 032320  001367                            BNE     8$              ;BRANCH IF NOT ALL 54 CHARACTERS WRITTEN.
7381 032322  110377  147706                     MOVB    R3,@#FDATA      ;LOAD THE LAST CHARACTER.
7382                                    ;+
7383                                    ; LOOP UNTIL THE TXFIFO'S ON ALL ACTIVE LINES HAVE BEEN FILLED.
7384                                    ;-
7385 032326  005705                    10$:     TST     R5              ;HAVE ALL FIFOS BEEN FILLED.
7386 032330  001402                            BEQ     12$             ;YES, THEN GO AND TRANSMIT THE CHARACTERS.
7387 032332  005201                             INC     R1              ;OTHERWISE SELECT THE NEXT LINE.
7388 032334  000702                            BR      2$              ;AND BRANCH TO TEST IF ITS ACTIVE.
7389                                    ;+
7390                                    ; ENABLE TRANSMISSION ON EACH LINE IN TURN AND WAIT FOR 100 MS TO ALLOW
7391                                    ; ANY INTERRUPTS TO OCCUR BEFORE DISABLING TRANSMISSION.  VERIFY THAT ONLY
7392                                    ; INTERRUPT OCCURED AND CHECK THAT THE CHARACTERS WERE TRANSMITTED UNCORUPTED
7393                                    ; AND ON THE CORRECT LINE.
7394                                    ;-
7395 032336  016705  147656            12$:     MOV     ACTLNS,R5       ;SET UP THE ACTIVE LINE BIT MAP.
7396 032342  005001                             CLR     R1              ;SET UP THE FIRST LINE NUMBER.
7397
7398 032344  000241                    14$:     CLC                     ;CLEAR THE CARRY BIT READY FOR THE ROTATION.
7399 032346  006005                             ROR     R5              ;ROTATE THE ACTIVE LINE BIT MAP INTO THE CARRY.
7400 032350  103157                            BCC     24$             ;AVOID TESTING THIS LINE IF ITS INACTIVE.
7401 032352  110177  147650                     MOVB    R1,@CSRA        ;LOAD THE LINE NUMBER OF THE OUT INTO THE CSR.
7402 032356  010103                             MOV     R1,R3           ;INITIALISE THE "EXPECTED" DATA PATTERN FOR
7403 032360  000303                             SWAB    R3              ;THIS LINE.
7404 032362  012767  013563  151462             MOV     #6003.,ERRNBR   ;SET THE ERROR NUMBER TO 6003.
7405 032370  004767  162774                     JSR     PC,PUFIFO       ;PURGE THE RXFIFO.
7406 032374  103151                            BCC     50$             ;REPORT THE ERROR IF THE FIFO WOULD NOT PURGE.
7407 032376  005067  147670                     CLR     TXINTC          ;CLEAR THE INTERRUPT COUNT.
7408 032402  005067  147666                     CLR     TXINTF          ;CLEAR THE INTERRUPT FLAGS.
7409 032406  052777  100000  147626             BIS     #BIT15,@TXAD2A  ;ENABLE TRANSMISSION ON THIS LINE.
7410 032414                            SETPRI  #PRI04                  ;ALLOW DEVICE INTERRUPTS.
     032414  012700  000200                                                     MOV     #PRI04,R0
     032420  104441                                                             TRAP    C$SPRI
7411 032422  012704  000144                     MOV     #100.,R4        ;SET THE DELAY.
7412 032426  004767  161630                     JSR     PC,DELAY        ;DELAY FOR 100 MILLI SECS.
7413 032432                            SETPRI  #PRI05
     032432  012700  000240                                                     MOV     #PRI05,R0
     032436  104441                                                             TRAP    C$SPRI
7414 032440  042777  100000  147574             BIC     #BIT15,@TXAD2A  ;DISABLE TRANSMISSION ON THIS LINE.
7415                                    ;+
7416                                    ; VERIFY THAT ONLY ONE INTERRUPT OCCURED AND THAT IT WAS ACCOMPANIED BY
7417                                    ; A CORRESPONDING TX ACTION.
7418                                    ;-
7419 032446  005267  151400                     INC     ERRNBR          ;SET THE ERROR NUMBER TO 6004.
7420 032452  005767  147614                     TST     TXINTC          ;HAS AN INTERRUPT OCCURED ?
7421 032456  001520                            BEQ     50$             ;REPORT THE ERROR IF NO INTERRUPT OCCURED.
7422 032460  005267  151366                     INC     ERRNBR          ;SET THE ERROR NUMBER TO 6005.
7423 032464  005767  147604                     TST     TXINTF          ;HAS AN INTERRUPT OCCURED WITHOUT A TX ACTION ?
7424 032470  100513                            BMI     50$             ;REPORT THE ERROR IF IT HAS.
7425 032472  022767  000001  147572             CMP     #1,TXINTC       ;DID ONLY ONE INT OCCUR ?
```

```
7426 032500  001424                        BEQ     16$             ;AVOID THE ERROR IF ONLY ONE INTERRUPT.
7427                                ;+
7428                                ; REPORT THE ERROR, MORE THAN ONE INTERRUPT. >>>>> 6006 <<<<<.
7429                                ;-
7430 032502  016702  147564                 MOV     TXINTC,R2       ;PASS THE ACTUAL NUMBER OF INTERRUPTS.
7431 032506  010104                        MOV     R1,R4           ;PASS THE LINE NUMBER.
7432 032510  012701  000001                MOV     #1,R1           ;PASS THE EXPECTED NUMBER OF INTS.
7433 032514  010346                        MOV     R3,-(SP)        ;SAVE THE DATA PATTERN.
7434 032516  012703  010040                MOV     #EM6003,R3      ;PASS THE MESSAGE,
7435                                                               ; "MORE THAN ONE TX-INT OCCURED FROM A FULL
7436                                                               ;  TX FIFO".
7437 032522  005267  151324                INC     ERRNBR          ;SET THE ERROR NUMBER TO 6006.
7438 032526  012767  012276  151322        MOV     #ER6001,ERRBLK  ;SET UP THE ERROR BLOCK.
7439 032534                               ERROR
     032534  104460                                                                TRAP    C$ERROR
7440 032536  012603                        MOV     (SP)+,R3        ;RESTORE THE DATA PATTERN.
7441 032540  010401                        MOV     R4,R1           ;RESTORE THE LINE NUMBER.
7442 032542  032767  000100  147440        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
7443 032550  001473                       BEQ     60$             ;EXIT THE TEST IF IT HAS.
7444
7445
7446                                ;+
7447                                ; READ THE CHARACTERS FROM THE RXFIFO AND VERIFY THEY ARE CORRECT AND WERE
7448                                ; RECIEVED ON THE CORRECT LINE.
7449                                ; THIS SUBROUTINE REPORTS ERRORS, >>>>> 6007 THRU 6008 <<<<<.
7450                                ;-
7451 032552  012767  013567  151272 16$:    MOV     #6007.,ERRNBR   ;SET UP THE ERROR NUMBER TO 6007.
7452 032560  005000                        CLR     R0              ;INITIALISE THE NUMBER OF CHARS READ COUNT.
7453 032562  012704  000025                MOV     #21.,R4         ;SET UP THE OUTER LOOP COUNT.
7454 032566  012702  000003         18$:    MOV     #3,R2           ;SET UP THE INNER LOOP COUNT.
7455 032572  004767  163140         20$:    JSR     PC,REPDER       ;READ A CHARACTER FROM THE RXFIFO, VERIFY THAT
7456                                                               ;IT IS CORRECT AND CAME FROM THE UUT.  REPORT
7457                                                               ;ANY ERRORS. >>>>> 6007 THRU 6008 <<<<<.
7458 032576  103022                       BCC     22$             ;BRANCH TO REPORT THE ERROR IF THE FIFO EMPTY.
7459 032600  005767  147446                TST     EXOERR          ;TEST THE "EXIT ON ERROR" FLAG.
7460 032604  001055                       BNE     60$             ;EXIT THE TEST IF SET, I.E.  AN ERROR OCCURED,
7461                                                               ;AND NO EXTENDED ERROR REPORTING WAS REQUESTED.
7462 032606  005200                        INC     R0              ;INCREMENT THE READ CHAR COUNT.
7463 032610  105203                        INCB    R3              ;INCREMENT THE LOBYTE OF THE DATA PATTERN.
7464 032612  000303                        SWAB    R3              ;SWAP BYTES TO PLACE EXPECTED CHAR IN LOBYTE.
7465 032614  005302                        DEC     R2              ;DECREMENT THE INNER LOOP COUNTER.
7466 032616  001365                       BNE     20$             ;BRANCH TO READ ANOTHER CHAR IF 3 CHARS HAVE
7467                                                               ;NOT BEEN READ.
7468 032620  000303                        SWAB    R3              ;RESTORE THE DATA BYTES TO THE CORRECT POSITION
7469 032622  005304                        DEC     R4              ;DECREMENT THE OUTER LOOP COUNTER.
7470 032624  001360                       BNE     18$             ;BRANCH TO READ ANOTHER 3 CHARACTERS IF NOT
7471                                                               ;ALL 63 HAVE BEEN READ.
7472 032626  004767  163104                JSR     PC,REPDER       ;READ AND CHECK THE LAST CHARACTER.
7473 032632  103004                       BCC     22$             ;BRANCH TO REPORT THE ERROR IF THE FIFO EMPTY.
7474 032634  005767  147412                TST     EXOERR          ;TEST THE "EXIT ON ERROR" FLAG.
7475 032640  001037                       BNE     60$             ;EXIT THE TEST IF SET.
7476 032642  000422                       BR      24$             ;OTHERWISE GO AND TEST ANOTHER LINE.
7477                                ;+
7478                                ; REPORT THE ERROR, NOT ALL CHARACTERS TRANSMITTED. >>>>> 6009 <<<<<.
7479                                ;
7480 032644  012767  013571  151200 22$:    MOV     #6009.,ERRNBR   ;SET THE ERROR NUMBER TO 6009.
7481 032652  012703  010243                MOV     #EM6006,R3      ;PASS THE MESSAGE,
```

```
                                                                ; "TX FIFO BAD, TOO FEW CHARS RECIEVED"
032656  010104                          MOV     R1,R4           ;PASS THE LINE NUMBER OF THE UUT.
032660  012701  000100                  MOV     #64.,R1         ;PASS THE EXPECTED NUMBER OF CHARACTERS.
032664  010002                          MOV     R0,R2           ;PASS THE ACTUAL NUMBER RECIEVED.
032666  012767  012276  151162          MOV     #ER6001,ERRBLK  ;SET UP THE ERROR REPORTING ROUTINE.
032674                                  ERROR
032674  104460                                                                  TRAP    C$ERROR
032676  010401                          MOV     R4,R1           ;RESTORE THE LINE NUMBER.
032700  032767  000100  147302          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
032706  001414                          BEQ     60$             ;EXIT THIS TEST IF IT HAS.
                        ;+
                        ; IF ALL ACTIVE LINES HAVE BEEN TESTED THEN EXIT THE TEST, OTHERWISE BRANCH TO
                        ; TEST ANOTHER LINE.
                        ;-
032710  005705          24$:    TST     R5              ;HAVE ALL LINES BEEN TESTED ?
032712  001405                  BEQ     52$             ;YES, THEN GO AND REPORT ERROR SUMMARIES IF
                                                        ;NECESSARY.
032714  005201                  INC     R1              ;OTHERWISE SELECT THE NEXT LINE.
032716  000612                  BR      14$             ;AND BRANCH TO TEST IF IT IS ACTIVE.

032720  004767  163746          50$:    JSR     PC,TSABRT       ;REPORT NON-RELATED TEST ERROR.
032724  000405                  BR      60$             ;ABORT THE TEST.
032726  012767  013572  151116  52$:    MOV     #6010.,ERRNBR   ;SET THE ERROR NUMBER TO 6010.
032734  004767  163314          JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF NECESSARY.
032740                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
032740  012700  000340                                                          MOV     #PRI07,R0
032744  104441                                                                  TRAP    C$SPRI
032746  004767  164244                  JSR     PC,TXIEO        ;CLEAR THE TX INT ENABLE BIT IN THE DEVICE CSR.
032752                                  CLRVEC  TXVECA          ;CLEAR DOWN THE TX INT VECTOR.
032752  016700  147240                                                          MOV     TXVECA,R0
032756  104436                                                                  TRAP    C$CVEC
032760  005067  147264          62$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
032764                                  ENDTST
032764                                                                  L10045:
032764  104401                                                                  TRAP    C$ETST
```

```
                                   .SBTTL  HARDWARE TEST           - BREAKB -
                                   ;++*****************************************************************************
                                   ;*                  - BREAK GENERATION TEST -
                                   ;*      THIS TEST VERIFIES THAT ALL SERIAL TRANSMIT LINES CAN GENERATE A BREAK
                                   ;*      BY SETTING THE BRK BIT IN THE ASSOCIATED LNCTRL REGISTER.
                                   ;*      USE OF THE INTERNAL LOOPBACK FEATURE OF THE DUARTS IS MADE TO MINIMISE
                                   ;*      ANY EXTERNAL EFFECTS CAUSED ON THE SERIAL LINES BY THIS TEST
                                   ;*      FRAMING ERROR DETECTION IS USED TO INDICATE THE PRESENCE OF A BREAK,
                                   ;*      BY SETTING THE APPROPRIATE BIT IN THE RBUF REGISTER.
                                   ;--*****************************************************************************

 032766                                    BGNTST
      032766                                                                              T17:

 032766  012767  177777  147254             MOV     #-1,CTRLCF        ;INDICATE THAT WE ARE IN A TEST.
         000021                             TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 032774  012767  000021  147252             MOV     #TNUM,TSTNUM      ;SET UP THE TEST NUMBER.       (64)
 033002  012767  000001  151040             MOV     #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 033010  012767  014401  151034             MOV     #6401.,ERRNBR     ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 033016  012767  010307  151030             MOV     #EM6401,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
                                   ;+
                                   ; 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 >>>>> 6401 <<<<<.
                                   ;-
 033024  004767  161166                     JSR     PC,CLNRST         ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 033030  103165                             BCC     60$               ;EXIT TEST IF FATAL ERROR FOUND.
                                   ;+
                                   ; SET UP DEVICE UNDER TEST (DUT) TO:
                                   ;     DISABLE TRANSMISSION AND RECEPTION INTERRUPTS.
                                   ;     DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME TO CLEAR ANY BREAKS.
                                   ;-
 033032  012705  177777                     MOV     #MAPLNS,R5        ;PASS ACTIVE LINE BIT MAP.
 033036  012700  000200                     MOV     #200,R0           ;PASS INTERNAL LOOPBACK MODE.
 033042  004767  164504                     JSR     PC,WTWLNC         ;SELECT INTERNAL LOOPBACK,DISABLE DMA.
 033046  012704  000012                     MOV     #10.,R4           ;PASS DELAY TIME OF 10 MILLI SECONDS.
 033052  004767  161204                     JSR     PC,DELAY          ;DELAY TO ALLOW ANY BREAKS TO BE CLEARED.
                                   ;+
                                   ; SET UP TRANSMISSION AN RECEPTION PARAMETERS FOR ALL LINES.
                                   ; 9600 BAUD,8 CHAR,1 STOPBIT,NO PARITY.
                                   ;-
 033056  012700  156430                     MOV     #156430,R0        ;SET UP BAUD RATE,ETC.
 033062  004767  164514                     JSR     PC,WTWLPR         ;SET COMMUNICATION PARAMETERS ON ALL LINES.
                                   ;+
                                   ; ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
                                   ;-
 033066  016705  147126                     MOV     ACTLNS,R5         ;PASS ACTIVE LINE BIT MAP.
 033072  004767  164024                     JSR     PC,TXENBL         ;ENABLE TRANSMISSIONS ON ALL LINES.
                                   ;+
                                   ; PURGE THE FIFO OF ANY UNWANTED CHARACTERS.
                                   ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 6402 THRU 6404 <<<<<.
                                   ;-
 033076  005267  150750                     INC     ERRNBR            ;SET ERROR NUMBER TO 6402.
 033102  004767  162344                     JSR     PC,PUFIFR         ;PURGE FIFO.
 033106  103136                             BCC     60$               ;ABORT TEST IF FIFO WILL NOT CLEAR.
                                   ;+
                                   ; VERIFY BREAK GENERATION ON INDIVIDUAL LINES.
```

```
                                        ;  CLEAR BREAKS ON ALL LINES.
                                        ;  DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME FOR ANY BREAKS TO BE CLEARED.
                                        ;  SELECT LINE,SET BREAK BIT IN LNCTRL REGISTER.
                                        ;  TEST FOR A CHARACTER IN THE FIFO WITH FRAME ERROR.
                                        ;-
033110  005002                  2$:       CLR     R2                ;CLEAR LINE COUNTER.
033112  012703  000001                    MOV     #1,R3             ;SET UP ACTIVE LINE BIT MASK.
033116  030367  147076            4$:     BIT     R3,ACTLNS         ;CHECK IF THIS LINE IS ACTIVE.
033122  001440                           BEQ     8$                ;GO SELECT NEXT LINE IF THIS ONE IS INACTIVE.
033124  012700  000200                    MOV     #200,R0           ;SET UP PARAMETER TO CLEAR BREAK BITS.
033130  004767  164416                    JSR     PC,WTWLNC         ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
033134  012704  000012                    MOV     #10.,R4           ;PASS DELAY TIME OF 10 MILLI SECONDS.
033140  004767  161116                    JSR     PC,DELAY          ;DELAY TO ALLOW BREAKS TO BE CLEARED.
                                        ;+
                                        ; SET BREAK BIT ON SELECTED LINE.
                                        ;  SET UP PARAMETERS TO TEST FOR THE FRAME ERROR BIT SET IN RBUF.
                                        ;  TIME-OUT = 5 MILLI SECONDS.
                                        ;  CALL ROUTINE TO CHECK FOR CONDITION FOUND.
                                        ;-
033144  010305                  6$:       MOV     R3,R5             ;COPY ACTIVE LINE BIT MASK.
033146  012700  000214                    MOV     #214,R0           ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
033152  004767  164374                    JSR     PC,WTWLNC         ;SET BREAK ON SELECTED LINE.
                                        ;+
                                        ; DELAY FOR 5 MS TO ALLOW TIME FOR BREAK TO BE GENERATED AND RECEIVED.
                                        ; VERIFY RECEPTION OF A CHARACTER WITH FRAME ERROR BIT SET.
                                        ;-
033156  012704  000005                    MOV     #5.,R4            ;SET DELAY VALUE TO 5 MILLI SECS.
033162  004767  161074                    JSR     PC,DELAY          ;ALLOW TIME FOR CHARACTER RECEPTION.
033166  017700  147036                    MOV     @RBUFA,R0         ;GET CHARACTER FROM RBUF REGISTER.
033172  032700  020000                    BIT     #BIT13,R0         ;CHECK FOR FRAME ERROR BIT.
033176  001012                           BNE     8$                ;SKIP ERROR REPORT IF SET.
033200  012701  010344                    MOV     #EM6402,R1        ;SELECT MESSAGE TO BE PRINTED.
                                      ;REPORT ERROR"BREAK NOT RECEIVED ON LINE #NN"
033204                                    ERRDF   6405,EM6401,ER6401 ;            >>>>> ERROR #6405 <<<<<
033204  104455                                                                    TRAP    C$ERDF
033206  014405                                                                    .WORD   640
033210  010307                                                                    .WORD   EM6401
033212  012400                                                                    .WORD   ER6401
                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
033214  032767  000100  146766            BIT     #BIT06,OPTION     ;EXIT WITH TEST FAILURE MESSAGE IF
033222  001470                           BEQ     60$               ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                  ;DURING THE SOFTWARE QUESTIONS.
033224  006303                    8$:     ASL     R3                ;SHIFT BIT MASK FOR NEXT LINE.
033226  005202                            INC     R2                ;NEXT LINE
033230  020227  000020                    CMP     R2,#NUMLNS        ;CHECK FOR MAX LINE COUNT.
033234  001330                           BNE     4$                ;IF <>,LOOP TO CHECK NEXT LINE
                                        ;+
                                        ; VERIFY BREAK GENERATION ON ALL LINES SIMULTANEOUSLY.
                                        ;  CLEAR BREAKS ON ALL LINES.
                                        ;  DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME FOR ANY BREAKS TO BE CLEARED.
                                        ;  PURGE THE FIFO.
                                        ;  SET BREAK BIT IN LNCTRL REGISTERS ON ALL ACTIVE LINES.
                                        ;  TEST FOR CHARACTERS IN THE FIFO WITH FRAME ERROR.
```

```
7620                                    ;-
7621 033236  012705  177777                     MOV     #MAPLNS,R5      ;SET UP LINE TO CLEAR BREAKS ON.
7622 033242  012700  000200                     MOV     #200,R0         ;SET UP PARAMETER TO CLEAR BREAK BITS.
7623 033246  004767  164300                    JSR     PC,WTWLNC       ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
7624 033252  012704  000012                     MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
7625 033256  004767  161000                    JSR     PC,DELAY        ;DELAY TO ALLOW BREAKS TO BE CLEARED.
7626                                    ;+
7627                                    ; PURGE THE FIFO OF UNWANTED CHARACTERS.
7628                                    ;-
7629 033262  004767  162102                     JSR     PC,PUFIFO       ;PURGE FIFO.
7630 033266  103044                             BCC     50$             ;GO REPORT ERROR IF FAILED TO CLEAN_OUT FIFO.
7631                                    ;+
7632                                    ; SET UP PARAMETERS FOR SETTING THE BREAK BIT ON ALL ACTIVE LINES.
7633                                    ; THEN CALL ROUTINE TO DO IT.
7634                                    ;-
7635 033270  016705  146724            10$:     MOV     ACTLNS,R5       ;SET UP ACTIVE LINE BIT MASK.
7636 033274  012700  000214                     MOV     #214,R0         ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
7637 033300  004767  164246                    JSR     PC,WTWLNC       ;SET BREAK ON SELECTED LINES.
7638                                    ;+
7639                                    ;  DELAY FOR 10 MILLI SECONDS,TO ALLOW TIME FOR RECEPTION.
7640                                    ;  TEST FOR CHARACTERS IN FIFO WITH FRAME ERROR BIT SET.
7641                                    ;-
7642 033304  012704  000012                     MOV     #10.,R4         ;SET DELAY VALUE TO 10 MILLI SECS.
7643 033310  004767  160746                    JSR     PC,DELAY        ;ALLOW TIME FOR CHARACTER RECEPTION.
7644 033314  010502                             MOV     R5,R2           ;COPY ACTIVE LINE BIT MAP.
7645 033316  004767  161354                    JSR     PC,MAPCNT       ;COUNT THE NUMBER OF LINES AVAILABLE.
7646 033322  017701  146702            12$:     MOV     @RBUFA,R1       ;GET CHARACTER FROM RBUF REGISTER.
7647 033326  100011                            BPL     14$             ;BRANCH IF DATA_VALID NOT SET.
7648 033330  032701  020000                     BIT     #BIT13,R1       ;CHECK FOR FRAME ERROR BIT.
7649 033334  001406                             BEQ     14$             ;DO NOT CLR FLG FOR THIS LINE IF FRAME BIT CLR.
7650 033336  000301                             SWAB    R1              ;GET LINE NUMBER IN LOW BYTE.
7651 033340  042701  177400                     BIC     #177400,R1      ;CLEAR EVERTHING BUT THE LINE NUMBER.
7652 033344  004767  161300                    JSR     PC,LINBIT       ;CALC BIT MASK FROM LINE NUMBER.
7653 033350  040005                             BIC     R0,R5           ;CLEAR LINE FLAG.
7654 033352  005302                    14$:     DEC     R2              ;DECREMENT THE LINE NUMBER COUNTER.
7655 033354  001362                            BNE     12$             ;LOOP TO GET THE NEXT CHARACTER.
7656 033356  005705                             TST     R5              ;CHECK IF ANY BREAKS NOT RECEIVED.
7657 033360  001411                            BEQ     60$             ;EXIT TEST IF ALL CLEAR.
7658 033362  012701  010344                     MOV     #EM6402,R1      ;SELECT MESSAGE TO BE PRINTED.
7659                                    ;REPORT ERROR"BREAK NOT RECEIVED ON LINE #NN".
7660 033366                                     ERRDF   6406,EM6401,ER6401;             >>>>> ERROR #6407 <<<<<.
     033366  104455                                                                             TRAP    C$ERDF
     033370  014406                                                                             .WORD   6406
     033372  010307                                                                             .WORD   EM6401
     033374  012400                                                                             .WORD   ER6401
7661 033376  000402                            BR      60$             ;EXIT THE TEST.
7662
7663 033400  004767  163266            50$:     JSR     PC,TSABRT       ;ABORT THE TEST.
7664 033404  005067  146640            60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7665 033410                                     ENDTST
     033410                                                                         L10046:
     033410  104401                                                                     TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - NORERR -
                                ;++*************************************************************************
                                ;*                      - NO OVERRUN ERROR TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT WILL NOT REPORT DATA OVERRUN
                                ;*      ERRORS WHEN THEY DO NOT OCCUR.
                                ;*      THIS TEST PUTS 256 CHARACTERS IN THE DUT FIFO PLUS 4 IN EACH ACTIVE
                                ;*      UART AND VERIFIES THAT NO OVERRUN ERRORS ARE REPORTED.
                                ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                ;*      REPORTED LATER.
                                ;*
                                ;--*************************************************************************

                                        BGNTST
                                                                                T18::
                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
                                                                                        MOV     #PRIO5,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (66)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #6601.,ERRNBR   ;SET ERROR NUMBER TO 6601.
                                         MOV    #EM6601,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 6601 <<<<<.
                                ;-
                                        JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
                                        BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
                                        JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.
                                ;*
                                ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                ;-
                                        JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
                                        BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
                                        JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.
                                        JSR     PC,INDATP       ;INITIALISE DATA PATTERN.
                                ;*
                                ; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                ;-
                                     ;*
                                     ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
                                     ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
                                     ;-
                                         INC     ERRNBR         ;SET THE ERROR REPORT NUMBER TO 6602.
                                         MOV     #204,R0        ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
                                        JSR     PC,WTWLNC       ;INITIALAISE THE LINE CONTROL REGISTER.
                                         MOV    #177670,R0     ;PASS THE LPR CONTENTS.
                                        JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
                                         MOV    #10.,R4        ;PASS DELAY TIME OF 10 MILLI SECONDS.
                                        JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
                                         MOV    #BUFBAS,R2     ;PASS THE START OF THE DATA PATTERN TO TX.
```

```
7721 033546  012703  000400                    MOV     #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
7722 033552  004767  160622                    JSR     PC,DODMA          ;TRANSMIT THE DATA PATTERN.
7723 033556  103157                            BCC     50$               ;EXIT IF ERROR FOUND DURING DMA TX.
7724                                          ;+
7725                                          ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
7726                                          ; THE FIFO.
7727                                          ;-
7728 033560  005267  150266                    INC     ERRNBR            ;SET ERROR NUMBER TO 6603.
7729 033564  012701  170536                    MOV     #170536,R1        ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
7730 033570  016702  146432                    MOV     CSRA,R2           ;PASS THE ADDRESS OF THE CSR.
7731 033574  004767  163636                    JSR     PC,WAIBIS         ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
7732 033600  103146                            BCC     50$               ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
7733 033602  012704  000005                    MOV     #5,R4             ;PASS DELAY OF 5 MILLI SECS.
7734 033606  004767  160450                    JSR     PC,DELAY          ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7735
7736                                          ;+
7737                                          ; TRANSMIT 4 CHARACTERS ON EACH ACTIVE LINE.
7738                                          ;-
7739 033612  016705  146402                    MOV     ACTLNS,R5         ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
7740 033616  012700  000204                    MOV     #204,R0           ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
7741 033622  004767  163724                    JSR     PC,WTWLNC         ;INITILAISE THE LINE CONTROL REGISTER.
7742 033626  012700  177670                    MOV     #177670,R0        ;PASS THE LPR CONTENTS.
7743 033632  004767  163744                    JSR     PC,WTWLPR         ;SET THE LPR CONTENTS TO 38.4K BAUD.
7744 033636  012704  000012                    MOV     #10.,R4           ;PASS DELAY TIME OF 10 MILLI SECONDS.
7745 033642  004767  160414                    JSR     PC,DELAY          ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7746
7747 033646  012702  002650                    MOV     #BUFBAS,R2        ;PASS THE START OF THE DATA PATTERN TO TX.
7748 033652  012703  000004                    MOV     #4,R3             ;PASS THE LENGTH OF THE DATA PATTERN.
7749 033656  005001                            CLR     R1                ;CLEAR THE LINE COUNTER.
7750 033660  005267  150166                    INC     ERRNBR            ;SET ERROR NUMBER TO 6604.
7751 033664  010100                    2$:     MOV     R1,R0
7752 033666  006300                            ASL     R0                ;CALCULATE THE LINE OFFSET FROM THE LINE #.
7753 033670  036067  002336  146322            BIT     BITTBL(R0),ACTLNS ;TEST FOR THIS LINE BEING ACTIVE.
7754 033676  001403                            BEQ     4$                ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
7755 033700  004767  160474                    JSR     PC,DODMA          ;TRANSMIT THE 5 CHAR DATA PATTERN.
7756 033704  103104                            BCC     50$               ;ABORT IF ERROR FOUND DURING DMA TX.
7757 033706  005201                    4$:     INC     R1                ;INCREMENT THE LINE COUNTER.
7758 033710  020127  000020                    CMP     R1,#NUMLNS        ;TEST FOR ALL POSSIBLE LINES HANDLED
7759 033714  002763                            BLT     2$                ;LOOP IF NOT ALL LINES HANDLED.
7760
7761 033716  005267  150130                    INC     ERRNBR            ;SET ERROR NUMBER TO 6605.
7762 033722  012701  170040                    MOV     #170040,R1        ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
7763 033726  016702  146274                    MOV     CSRA,R2           ;PASS THE ADDRESS OF THE CSR.
7764 033732  004767  163500                    JSR     PC,WAIBIS         ;WAIT FOR A DMA TO COMPLETE, TX_ACTION SET.
7765 033736  103067                            BCC     50$               ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
7766 033740  012704  000005                    MOV     #5,R4             ;PASS DELAY OF 5 MILLI SECS.
7767 033744  004767  160312                    JSR     PC,DELAY          ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7768                                          ;+
7769                                          ; READ THE FIFO CHECKING FOR OVERRUN ERRORS.  REPORT ERRORS IF FOUND.
7770                                          ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
7771                                          ;-
7772 033750  016702  146244                    MOV     ACTLNS,R2
7773 033754  004767  160716                    JSR     PC,MAPCNT         ;GET THE NUMBER OF ACTIVE LINES.
7774 033760  006302                            ASL     R2
7775 033762  006302                            ASL     R2                ;MULTIPLY NUMBER OF ACTIVE LINES BY 4.
7776 033764  012705  000400                    MOV     #256.,R5
7777 033770  060205                            ADD     R2,R5             ;CALCULATE NUMBER OF CHARACTERS TO RX.
```

```
7778 033772  005004                          CLR     R4              ;CLEAR THE CHARACTER COUNTER.
7779 033774  012767  014716  150050  6$:     MOV     #6606.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7780 034002  017702  146222                  MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE FIFO.
7781 034006  100036                        BPL     10$             ;EXIT THE READ LOOP IF THE FIFO IS EMPTY.
7782                                ;+
7783                                ; CHECK IF THE READ CHARACTER IS A BMP CODE.
7784                                ;  IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
7785                                ;  ABORT THE TEST.
7786                                ;-
7787 034010  004767  157762                  JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
7788 034014  103002                          BCC     8$              ;BRANCH IF NOT A BMP CODE.
7789 034016                                  ERROR                   ;                >>>>> ERROR #6606 <<<<<.
     034016  104460                                                                      TRAP    C$ERROR
7790 034020  000440                          BR      60$             ;EXIT THIS TEST.
7791
7792 034022  005267  150024          8$:      INC     ERRNBR          ;SET ERROR NUMBER TO 6607.
7793 034026  005204                          INC     R4              ;COUNT THIS CHARACTER.
7794 034030  020405                        CMP     R4,R5           ;COMPARE # OF CHARS WITH MAX # OF CHARS.
7795 034032  003031                          BGT     50$             ;ABORT TEST IF TOO MANY VALID CHARS READ.
7796 034034  032702  040000                BIT     #BIT14,R2       ;TEST THE OVERRUN BIT OF THE READ CHAR.
7797 034040  001755                          BEQ     6$              ;LOOP TO READ THE NEXT CHAR IF NO ERROR.
7798 034042  005267  150004                  INC     ERRNBR          ;SET ERROR NUMBER TO 6608.
7799 034046  012767  012506  150002          MOV     #ER7801,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
7800 034054  012701  010442                  MOV     #EM6602,R1      ;PASS THE MESSAGE TO BE REPORTED.
7801 034060  010203                          MOV     R2,R3
7802 034062  000303                          SWAB    R3
7803 034064  042703  177760                  BIC     #177760,R3      ;GET FAILING LINE NUMBER.
7804                                ;REPORT "OVERRUN ERROR REPORTED WHEN NONE FORCED, ON LINE NN ..."
7805 034070                                  ERROR                   ;                >>>>> ERROR  #6608 <<<<<.
     034070  104460                                                                      TRAP    C$ERROR
7806
7807                                ;+
7808                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7809                                ;-
7810 034072  032767  000100  146110          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7811 034100  001410                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7812                                                                ;DURING THE SOFTWARE QUESTIONS.
7813
7814 034102  000734                          BR      6$              ;LOOP TO READ THE NEXT CHAR.
7815
7816 034104  012767  014721  147740  10$:     MOV     #6609.,ERRNBR   ;SET ERROR NUMBER TO 6609.
7817 034112  020405                          CMP     R4,R5           ;COMPARE NUMBER OF CHARS READ WITH EXPECTED.
7818 034114  001402                          BEQ     60$             ;EXIT TEST WITHOUT ABORT IF CORRECT # OF CHARS.
7819
7820 034116  004767  162550          50$:     JSR     PC,TSABRT       ;ABORT THE TEST, NON-RELATED TEST ERROR FOUND.
7821 034122  005067  146122          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7822 034126                                  ENDTST
     034126                                                                          L10047:
     034126  104401                                                                      TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - ORERR
                                ;+**********************************************************************
                                ;*                        - OVERRUN ERROR TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DUT WILL REPORT DATA OVERRUN ERRORS WHEN
                                ;*      THEY OCCUR.
                                ;*      THIS TEST PUTS 256 CHARACTERS IN THE DUT FIFO PLUS 5 IN EACH ACTIVE
                                ;*      UART AND VERIFIES THAT OVERRUN ERRORS ARE REPORTED ON ALL ACTIVE LINES.
                                ;*      ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
                                ;*      HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
                                ;*      REPORTED LATER.
                                ;*
                                ;-**********************************************************************

034130                                  BGNTST
034130                                                                  T19::
034130                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
034130  012700  000240                                                          MOV     #PRIO5,R0
034134  104441                                                                  TRAP    C$SPRI
                000023                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
034136  012767  000023  146110           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (67)
034144  012767  177777  146076           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
034152  012767  000001  147670           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
034160  012767  015055  147664           MOV     #6701.,ERRNBR  ;SET ERROR NUMBER TO 6701.
034166  012767  010514  147660           MOV     #EM6701,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 6701 <<<<<.
                                ;-
034174  004767  160016                   JSR     PC,CLNRST      ;RESET THE DHU 11, REPORT ANY ERRORS FOUND.
034200  103402                           BCS     .+6            ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
034202  000167  000660                   JMP     60$            ;EXIT THE TEST, FATAL ERROR WAS FOUND.
                                ;+
                                ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                                ; INITIALIZE THE 256 BYTE DATA PATTERN.
                                ;-
034206  004767  160256                   JSR     PC,FINACT      ;FIND AN ACTIVE LINE.
034212  103402                           BCS     .+6            ;IF ACTIVE LINE IS FOUND, DON'T ABORT TEST.
034214  000167  000646                   JMP     60$            ;ABORT THE TEST, NO ACTIVE LINES WERE FOUND.
034220  004767  160324                   JSR     PC,INDATP      ;INITIALISE DATA PATTERN.
                                ;+
                                ; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
                                ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
                                ;-
                                     ;+
                                     ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
                                     ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
                                     ;-
034224  005267  147622                   INC     ERRNBR         ;SET ERROR NUMBER TO 6702.
034230  012700  000204                   MOV     #204,R0        ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
034234  004767  163312                   JSR     PC,WTWLNC      ;INITILAISE THE LINE CONTROL REGISTER.
034240  012700  177670                   MOV     #177670,R0     ;PASS THE LPR CONTENTS.
034244  004767  163332                   JSR     PC,WTWLPR      ;SET THE LPR CONTENTS TO 38.4K BAUD.
034250  012704  000012                   MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI SECONDS.
034254  004767  160002                   JSR     PC,DELAY       ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
034260  012702  002650                   MOV     #BUFBAS,R2     ;PASS THE START OF THE DATA PATTERN TO TX.
```

```
7878 034264  012703  000400                    MOV     #BUFMID BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
7879 034270  004767  160104                    JSR     PC,DODMA          ;TRANSMIT THE DATA PATTERN.
7880 034274  103402                            BCS     .+6               ;IF NO ERROR FOUND DURING DMA TX, DON'T ABORT.
7881 034276  000167  000560                    JMP     50$               ;ABORT TEST, ERROR FOUND DURING DMA TX.
7882                                        ;+
7883                                        ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
7884                                        ; THE FIFO.
7885                                        ;-
7886 034302  005267  147544                    INC     ERRNBR            ;SET ERROR NUMBER TO 6703.
7887 034306  012701  170536                    MOV     #170536,R1        ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
7888 034312  016702  145710                    MOV     CSRA,R2           ;PASS THE ADDRESS OF THE CSR.
7889 034316  004767  163114                    JSR     PC,WAIBIS         ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
7890 034322  103402                            BCS     .+6               ;IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
7891 034324  000167  000532                    JMP     50$               ;ABORT TEST, TIME-OUT ON DMA COMPLETION.
7892 034330  012704  000005                    MOV     #5,R4             ;PASS DELAY OF 5 MILLI SECS.
7893 034334  004767  157722                    JSR     PC,DELAY          ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7894                                        ;+
7895                                        ; TRANSMIT 5 CHARACTERS ON EACH ACTIVE LINE.
7896                                        ;-
7897 034340  016705  145654                    MOV     ACTLNS,R5         ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
7898 034344  012700  000204                    MOV     #204,R0           ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
7899 034350  004767  163176                    JSR     PC,WTWLNC         ;INITILAISE THE LINE CONTROL REGISTER.
7900 034354  012700  177670                    MOV     #177670,R0        ;PASS THE LPR CONTENTS.
7901 034360  004767  163216                    JSR     PC,WTWLPR         ;SET THE LPR CONTENTS TO 38.4K BAUD.
7902 034364  012704  000012                    MOV     #10.,R4           ;PASS DELAY TIME OF 10 MILLI SECONDS.
7903 034370  004767  157666                    JSR     PC,DELAY          ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7904
7905 034374  012702  002650                    MOV     #BUFBAS,R2        ;PASS THE START OF THE DATA PATTERN TO TX.
7906 034400  012703  000005                    MOV     #5,R3             ;PASS THE LENGTH OF THE DATA PATTERN.
7907 034404  005001                            CLR     R1                ;CLEAR THE LINE COUNTER.
7908 034406  005267  147440                    INC     ERRNBR            ;SET ERROR NUMBER TO 6704.
7909 034412  010100                    2$:     MOV     R1,R0
7910 034414  006300                            ASL     R0                ;CALCULATE LINE OFFSET FROM THE LINE 0.
7911 034416  036067  002336  145574            BIT     BITTBL(R0),ACTLNS ;TEST FOR THIS LINE BEING ACTIVE.
7912 034424  001405                            BEQ     4$                ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
7913 034426  004767  157746                    JSR     PC,DODMA          ;TRANSMIT THE 5 CHAR DATA PATTERN.
7914 034432  103402                            BCS     .+6               ;IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
7915 034434  000167  000422                    JMP     50$               ;ABORT TEST, TIME-OUT ON DMA COMPLETION.
7916 034440  005201                    4$:     INC     R1                ;INCREMENT THE LINE NUMBER COUNTER.
7917 034442  020127  000020                    CMP     R1,#NUMLNS        ;TEST FOR ALL POSSIBLE LINES HANDLED
7918 034446  002761                            BLT     2$                ;LOOP IF NOT ALL LINES HANDLED.
7919
7920 034450  005267  147376                    INC     ERRNBR            ;SET ERROR NUMBER TO 6705.
7921 034454  012701  170040                    MOV     #170040,R1        ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
7922 034460  016702  145542                    MOV     CSRA,R2           ;PASS THE ADDRESS OF THE CSR.
7923 034464  004767  162746                    JSR     PC,WAIBIS         ;WAIT FOR A DMA TO COMPLETE, TX ACTION SET.
7924 034470  103174                            BCC     50$               ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
7925 034472  012704  000005                    MOV     #5,R4             ;PASS DELAY OF 5 MILLI SECS.
7926 034476  004767  157560                    JSR     PC,DELAY          ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7927                                        ;+
7928                                        ; READ 256 CHARS FROM THE FIFO CHECKING FOR BMP CODES.
7929                                        ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
7930                                        ;-
7931 034502  012704  000400                    MOV     #256.,R4          ;SET UP THE CHARACTER COUNTER.
7932 034506  012767  015062  147336    6$:     MOV     #6706.,ERRNBR     ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7933 034514  017702  145510                    MOV     @RBUFA,R2         ;READ A CHARACTER FROM THE FIFO.
7934 034520  100160                            BPL     50$               ;ABORT THE TEST IF DATA.VALID IS CLEAR.
```

```
7935 034522  005267  147324                    INC     ERRNBR          ;SET ERROR NUMBER TO 6707.
7936 034526  004767  157244                    JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
7937 034532  103551                            BCS     24$             ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
7938 034534  005304                            DEC     R4              ;COUNT THIS CHARACTER.
7939 034536  001363                            BNE     6$              ;LOOP IF NOT 256 CHARS READ FROM FIFO.
7940                                ;+
7941                                ; READ THE REMAINING AND VERIFY 1 OVERRUN PLUS 1 CHAR FROM EACH LINE.
7942                                ;-
7943 034540  005004                            CLR     R4              ;CLEAR THE OVERRUN ERROR FLAGS.
7944 034542  012700  003710                    MOV     #RXCNTB,R0
7945 034546  004767  157466                    JSR     PC,CLR16W       ;CLEAR RX CHAR COUNT TABLE.
7946 034552  012767  015064  147272  8$:       MOV     #6708.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7947 034560  017702  145444                    MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE FIFO.
7948 034564  100047                            BPL     14$             ;GO ANALYZE THE RESULTS IF ALL CHARS READ.
7949 034566  004767  157204                    JSR     PC,CHKBMP       ;CHECK IF CHAR IS A BMP CODE.
7950 034572  103531                            BCS     24$             ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
7951 034574  005267  147252                    INC     ERRNBR          ;SET ERROR NUMBER TO 6709.
7952 034600  010200                            MOV     R2,R0
7953 034602  000300                            SWAB    R0
7954 034604  042700  177760                    BIC     #177760,R0      ;CALCULATE THE LINE NUMBER OF THE CHAR.
7955 034610  006300                            ASL     R0              ;FORM WORD TABLE OFFSET FOR TABLE ACCESS.
7956 034612  042702  007400                    BIC     #7400,R2        ;REMOVE LINE NUMBER FROM THE READ CHAR.
7957 034616  036067  002336  145374            BIT     BITTBL(R0),ACTLNS       ;TEST FOR ACTIVE LINE.
7958 034624  001516                            BEQ     50$             ;ABORT TEST IF FOR INACTIVE LINE.
7959 034626  005267  147220                    INC     ERRNBR          ;SET ERROR NUMBER TO 6710.
7960 034632  005760  003710                    TST     RXCNTB(R0)      ;CHECK THE RX CHAR COUNTER FOR THIS LINE.
7961 034636  001006                            BNE     10$             ;IS THIS FIRST CHAR ON LINE?
7962 034640  020227  140000                    CMP     R2,#140000      ;YES, TEST FOR NULL CHAR WITH OVERRUN.
7963 034644  001414                            BEQ     12$             ;IS CHAR A NULL?
7964 034646  056004  002336                    BIS     BITTBL(R0),R4   ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
7965 034652  000411                            BR      12$             ;GO COUNT THE CHAR AND CONTINUE.
7966 034654  026027  003710  000004  10$:      CMP     RXCNTB(R0),#4
7967 034662  002077                            BGE     50$             ;5TH CHAR ON THIS LINE?  YES, ABORT.
7968 034664  032702  040000                    BIT     #BIT14,R2       ;NO, CHECK OVERRUN BIT.
7969 034670  001402                            BEQ     12$             ;IS OVERRUN BIT CLEAR?  YES, GO COUNT CHAR.
7970 034672  056004  002336                    BIS     BITTBL(R0),R4   ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
7971 034676  005260  003710            12$:    INC     RXCNTB(R0)      ;COUNT THIS CHARACTER.
7972 034702  000723                            BR      8$              ;LOOP UNTIL ALL CHARS ARE READ FROM FIFO.
7973                                ;+
7974                                ; TEST FOR ABORT CONDITIONS.  ONLY NONE ABORT CONDITIONS ARE:
7975                                ;     1) 2 CHARS RXED ON A LINE AND NO OVERRUN ERROR BIT FAILURE DETECTED.
7976                                ;     2) 2 TO 4 CHARS RXED ON A LINE AND AN OVERRUN BIT FAILURE DETECTED.
7977                                ;
7978 034704  005001                14$:    CLR     R1              ;INITIALIZE LINE LOOP, CLEAR LINE OFFSET.
7979 034706  012767  015067  147136  16$:    MOV     #6711.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7980 034714  036167  002336  145276            BIT     BITTBL(R1),ACTLNS
7981 034722  001415                            BEQ     18$             ;LINE ACTIVE?  NO, NEXT LINE.
7982 034724  026127  003710  000002            CMP     RXCNTB(R1),#2   ;YES.
7983 034732  002453                            BLT     50$             ;FEWER THAN 2 CHARS RXED?  YES, ABORT.
7984 034734  036104  002336                    BIT     BITTBL(R1),R4   ;NO.
7985 034740  001006                            BNE     18$             ;OVERRUN BIT ERROR FLAG SET?  YES, NEXT LINE.
7986 034742  005267  147104                    INC     ERRNBR          ;SET LINE NUMBER TO 6712.
7987 034746  026127  003710  000002            CMP     RXCNTB(R1),#2
7988 034754  001042                            BNE     50$             ;NOT 2 CHARS RXED?  YES, ABORT.  NO, NEXT LINE.
7989 034756  062701  000002            18$:    ADD     #2,R1           ;SET LINE OFFSET TO THE NEXT LINE.
7990 034762  020127  000040                    CMP     R1,#NUMLNS*2
7991 034766  002747                            BLT     16$             ;ALL LINES DONE?  NO, LOOP.  YES, CONTINUE.
```

```
7992                                    ;*
7993                                    ; CHECK FOR OVERRUN ERROR BIT FAILURES, PRINT ERROR MESSAGE IF FOUND.
7994                                    ;-
7995 034770  012767  015071  147054            MOV     #6715.,ERRNBR   ;SET UP ERROR NUMBER.
7996 034776  005001                            CLR     R1              ;INITIALIZE LOOP.  CLEAR LINE OFFSET.
7997 035000  010102                    20$:    MOV     R1,R2           ;COPY THE LINE OFFSET.
7998 035002  036104  002336                    BIT     BITTBL(R1),R4   ;OVERRUN BIT FAILURE FLAGS ARE IN R4.
7999 035006  001415                            BEQ     22$             ;ERROR FLAG CLEAR?  YES, NEXT LINE.
8000 035010  010103                            MOV     R1,R3
8001 035012  006203                            ASR     R3              ;CALCULATE LINE NUMBER FROM LINE OFFSET.
8002 035014  012767  012506  147034            MOV     #ER7801,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
8003 035022  012701  010546                    MOV     #EM6702,R1      ;PASS THE MESSAGE TO BE REPORTED.
8004                                ;REPORT "OVERRUN ERROR NOT REPORTED CORRECTLY WHEN FORCED, ON LINE NN ..."
8005 035026                                    ERROR                   ;               >>>>> ERROR  #6713 <<<<<.
     035026  104460                                                                                 TRAP    C$ERROR
8006
8007                                    ;*
8008                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
8009                                    ;-
8010 035030  032767  000100  145152            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
8011 035036  001413                            BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
8012                                                                   ;DURING THE SOFTWARE QUESTIONS.
8013
8014 035040  010201                            MOV     R2,R1           ;RESTORE THE LINE OFFSET THAT WAS DESTROYED.
8015 035042  046104  002336            22$:    BIC     BITTBL(R1),R4   ;CLEAR THE LINE ERROR FLAG WE JUST HANDLED.
8016 035046  001407                            BEQ     60$             ;ALL FAILURE BITS HANDLED?  YES, EXIT TEST.
8017 035050  062701  000002                    ADD     #2,R1           ;NO, INCREMENT THE LINE OFFSET.
8018 035054  000751                            BR      20$             ;LOOP TO HANDLE THE NEXT LINE.
8019
8020 035056                            24$:    ;REPORT "BMP CODE FOUND IN FIFO, TEST INVAILDATED."
8021 035056                                    ERROR                   ;               >>>>> ERROR <<<<<.
     035056  104460                                                                                 TRAP    C$ERROR
8022 035060  000402                            BR      60$             ;EXIT THIS TEST.
8023
8024 035062  004767  161604            50$:    JSR     PC,TSABRT       ;ABORT THE TEST.  ERROR # INDICATES FAULT TYPE.
8025 035066  005067  145156            60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8026
8027 035072                                    .ENDTST
     035072                                                                            L10050:
     035072  104401                                                                            TRAP    C$ETST
```

```
8029                                   .SBTTL  HARDWARE TEST           - REPBMP -
8030                                   ;** *******************************************************************
8031                                   ;*                      - REPORT ANY BMP CODES IN THE QUEUE -
8032                                   ;*      THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
8033                                   ;*      IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
8034                                   ;*      QUEUE.
8035                                   ;*      IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
8036                                   ;*      ERROR REPORTS.
8037                                   ;*
8038                                   ;-- *******************************************************************
8039 035074                                 BGNTST
     035074                                                                           T20::
8040        000024                           TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER,
8041 035074 012767 000024 145152            MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
8042 035102 012767 177777 145140            MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
8043 035110 016702 145274                   MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER.
8044 035114 012703 002412                   MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
8045 035120 020203                          CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
8046 035122 001411                          BEQ     60$             ;EXIT NO CODES IN THE QUEUE.
8047                                   ;*
8048                                   ; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
8049                                   ;*
8050                                        ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
8051
8052 035124 012701 011354                   MOV     #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
8053 035130                                 ERRDF   9301,EM9301,ER9301 ;           >>>>> ERROR #9301 <<<<<.
     035130 104455                                                                          TRAP    C$ERDF
     035132 022125                                                                          .WORD   9301
     035134 011200                                                                          .WORD   EM9301
     035136 013176                                                                          .WORD   ER9301
8054
8055 035140 012767 002412 145242            MOV     #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.
8056
8057 035146 005067 145076           60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8058 035152                                 ENDTST
     035152                                                                        L10051:
     035152 104401                                                                          TRAP    C$ETST
```

```
8067
8068
8069                                    .SBTTL  HARDWARE PARAMETER CODING SECTION
8070
8071
8072
8073                                    ;++
8074                                    ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
8075                                    ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8076                                    ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8077                                    ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8078                                    ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8079                                    ; WITH THE OPERATOR.
8080                                    ;--
8081
8082 035154                                    BGNHRD
     035154  000022                                                                 .WORD L10052-L$HARD/2
     035156                                                                L$HARD::
8083
8093                                    ;DEVICE CSR ADDRESS QUESTION:
8094 035156                                    GPRMA   HWPTQ1,0,0,160000,177776,YES
     035156  000031                                                                .WORD   T$CODE
     035160  035222                                                                .WORD   HWPTQ1
     035162  160000                                                                .WORD   T$LOLIM
     035164  177776                                                                .WORD   T$HILIM
8095                                    ;DEVICE INTERRUPT VECTOR QUESTION:
8096 035166                                    GPRMA   HWPTQ2,2,0,40,776,YES
     035166  001051                                                                .WORD   T$CODE
     035170  035240                                                                .WORD   HWPTQ2
     035172  000040                                                                .WORD   T$LOLIM
     035174  000776                                                                .WORD   T$HILIM
8097                                    ;ACTIVE LINES BIT MAP QUESTION:
8098 035176                                    GPRMD   HWPTQ3,4,0,MAPLNS,0,177777,YES
     035176  002032                                                                .WORD   T$CODE
     035200  035273                                                                .WORD   HWPTQ3
     035202  177777                                                                .WORD   MAPLNS
     035204  000000                                                                .WORD   T$LOLIM
     035206  177777                                                                .WORD   T$HILIM
8099                                    ;TYPE OF LOOPBACK QUESTION:
8100 035210                                    GPRMD   HWPTQ4,6,0,377,1,2,YES
     035210  003032                                                                .WORD   T$CODE
     035212  035321                                                                .WORD   HWPTQ4
     035214  000377                                                                .WORD   377
     035216  000001                                                                .WORD   T$LOLIM
     035220  000002                                                                .WORD   T$HILIM
8101
8102
8103 035222                                    ENDHRD
                                                                                   .EVEN
     035222                                                                L10052:
8104
8111
8112 035222  103  123  122                    HWPTQ1: .ASCIZ  /CSR ADDRESS: /
     035225  040  101  104
     035230  104  122  105
     035233  123  123  072
     035236  040  000
```

HARDWARE PARAMETER CODING SECTION

```
 8113 035240  111  116  124        HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
      035243  105  122  122
      035246  125  120  124
      035251  040  126  105
      035254  103  124  117
      035257  122  040  101
      035262  104  104  122
      035265  105  123  123
      035270  072  040  000
 8114 035273  101  103  124        HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
      035276  111  126  105
      035301  040  114  111
      035304  116  105  040
      035307  102  111  124
      035312  040  115  101
      035315  120  072  040
      035320  000
 8115 035321  124  131  120        HWPTQ4: .ASCIZ  /TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277):/
      035324  105  040  117
      035327  106  040  114
      035332  117  117  120
      035335  102  101  103
      035340  113  040  050
      035343  061  075  111
      035346  116  124  105
      035351  122  116  101
      035354  114  054  040
      035357  062  075  110
      035362  063  060  062
      035365  071  040  117
      035370  122  040  110
      035373  063  062  067
      035376  067  051  072
      035401  000
 8116
 8117                             .EVEN
```

```
8126
8127
8128                                    .SBTTL  SOFTWARE PARAMETER CODING SECTION
8129
8130                                    ;++
8131                                    ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
8132                                    ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8133                                    ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8134                                    ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8135                                    ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8136                                    ; WITH THE OPERATOR.
8137                                    ;--
8138
8139 035402                                    BGNSFT
     035402  000014                                                                .WORD L10053-L$SOFT/2
     035404                                                                  L$SOFT::
8140
8149                                    ;UNIT NUMBER PRINTOUT QUESTION:
8150 035404                                    GPRML   SWPTQ1,0,20,YES
     035404  000130                                                            .WORD   T$CODE
     035406  035434                                                            .WORD   SWPTQ1
     035410  000020                                                            .WORD   20
8151                                    ;EXTENDED ERROR REPORTING QUESTION:
8152 035412                                    GPRML   SWPTQ2,0,100,YES
     035412  000130                                                            .WORD   T$CODE
     035414  035510                                                            .WORD   SWPTQ2
     035416  000100                                                            .WORD   100
8153                                    ;+
8154                                    ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
8155                                    ;-
8156 035420                                    XFERF   ENDD
     035420  006044                                                            .WORD   T$CODE
8157
8158                                    ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
8159 035422                                    GPRMD   SWPTQ3,2,D,177777,0,177777,YES
     035422  001052                                                            .WORD   T$CODE
     035424  035543                                                            .WORD   SWPTQ3
     035426  177777                                                            .WORD   177777
     035430  000000                                                            .WORD   T$LOLIM
     035432  177777                                                            .WORD   T$HILIM
8160
8161                                            .EVEN
8162
8163 035434                                ENDD:   ENDSFT
                                                                                .EVEN
     035434                                                                 L10053:
8164
8165
8172 035434  122  105  120                       SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED:
     035437  117  122  124
     035442  040  125  116
     035445  111  124  040
     035450  116  125  115
     035453  102  105  122
     035456  040  101  123
     035461  040  105  101
     035464  103  110  040
```

SOFTWARE PARAMETER CODING SECTION

```
                035467    125     116     111
                035472    124     040     111
                035475    123     040     124
                035500    105     123     124
                035503    105     104     072
                035506    040     000
8173            035510    105     130     124         SWPTQ2: .ASCIZ  /EXTENDED ERROR REPORTING: /
                035513    105     116     104
                035516    105     104     040
                035521    105     122     122
                035524    117     122     040
                035527    122     105     120
                035532    117     122     124
                035535    111     116     107
                035540    072     040     000
8174            035543    116     125     115         SWPTQ3: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
                035546    102     105     122
                035551    040     117     106
                035554    040     111     116
                035557    104     111     126
                035562    111     104     125
                035565    101     114     040
                035570    104     101     124
                035573    101     040     105
                035576    122     122     117
                035601    122     123     040
                035604    124     117     040
                035607    122     105     120
                035612    117     122     124
                035615    040     117     116
                035620    040     101     040
                035623    114     111     116
                035626    105     072     040
                035631    000
8175                                            .EVEN
```

```
                                
                                
035632                          $PATCH::
035632                                  .BLKW   24
                                
                                
                                
                                
                                
                                
035702                                  LASTAD
                                                                .EVEN
035702  000000                                                  .WORD   0
035704  000000                                                  .WORD   0
035706                          L$LAST::
035706                                  ENDMOD
                                
                                
                                
                                
                                
                                
000001                                  .END
```

SYMBOL TABLE

```
ACTLNS  002220 G    CTRLCF  002250 G    DMABUF  014322 G    EM5804  007201 G    FDATO - 000006 G
ADR   = 000020 G    C$AU   = 000052     DODMA   014400 G     EM5805  007265 G    FINACT  014470 G
ADRPTR  014200 G    C$AUTO= 000061      DRADRT  002226 G     EM5901  007346 G    FSLSA   002234 G
ALTFLD  013370 G    C$BRK  = 000022     DROP    020730       EM5902  007402 G    FSLSO = 000006 G
ASLNTL  013442 G    C$BSEG= 000004      EDROP   021006       EM5903  007476 G    F$AU   = 000015
ASSEMB= 000010      C$BSUB= 000002      EF.CON= 000036 G     EM5904  007570 G    F$AUTO= 000020
BCOUNT  002322 G    C$CEFG= 000045      EF.NEW= 000035 G     EM5905  007654 G    F$BGN  = 000040
BITTBL  002336 G    C$CLCK= 000062      EF.PWR= 000034 G     EM6001  007754 G    F$CLEA= 000007
BIT0   = 000001 G    C$CLEA= 000012     EF.RES= 000037 G     EM6002  010000 G    F$DU   = 000016
BIT00  = 000001 G    C$CLOS= 000035     EF.STA= 000040 G     EM6003  010040 G    F$END  = 000041
BIT01  = 000002 G    C$CLP1= 000006     EF0503  004156 G     EM6004  010120 G    F$HARD= 000004
BIT02  = 000004 G    C$CVEC= 000036     EF1601  004163 G     EM6005  010165 G    F$HW   = 000013
BIT03  = 000010 G    C$DCLN= 000044     EF5801  004215 G     EM6006  010243 G    F$INIT= 000006
BIT04  = 000020 G    C$DODU= 000051     EF5901  004270 G     EM6401  010307 G    F$JMP  = 000050
BIT05  = 000040 G    C$DRPT= 000024     EF5902  004320 G     EM6402  010344 G    F$MOD  = 000000
BIT06  = 000100 G    C$DU   = 000053    EF6401  004350 G     EM6601  010405 G    F$MSG  = 000011
BIT07  = 000200 G    C$EDIT= 000003     EF7801  004417 G     EM6602  010442 G    F$PROT= 000021
BIT08    000 ^ G     C$ERDF= 000055     EF9001  004455 G     EM6701  010514 G    F$PWR  = 000017
BIT09  = 0010  G     C$ERHR= 000056     EF9002  004537 G     EM6702  010546 G    F$RPT  = 000012
BIT1   = 000002 G    C$ERRO= 000060     EF9003  004611 G     EM9009  010623 G    F$SEG  = 000003
BIT10  = 002000 G    C$ERSF= 000054     EF9004  004640 G     EM9010  010647 G    F$SOFT= 000005
BIT11  = 004000 G    C$ERSO= 000057     EF9005  004670 G     EM9014  010673 G    F$SRV  = 000010
BIT12  = 010000 G    C$ESCA= 000010     EF9006  004721 G     EM9017  010767 G    F$SUB  = 000002
BIT13  = 020000 G    C$ESEG= 000005     EF9010  004745 G     EM9026  011100 G    F$SW   = 000014
BIT14  = 040000 G    C$ESUB= 000003     EF9019  005034 G     EM9104  011124 G    F$TEST= 000001
BIT15  = 100000 G    C$ETST= 000001     EF9301  005053 G     EM9301  011200 G    GETPRM  020446
BIT2   = 000004 G    C$EXIT= 000032     EF9302  005131 G     EM9302  011257 G    GPRSOB  002376 G
BIT3   = 000010 G    C$GETB= 000026     EM0101  015124 G     EM9303  011307 G    G$CNTO= 000200
BIT4   = 000020 G    C$GETW= 000027     EM0102  015210 G     EM9304  011354 G    G$DELM= 000372
BIT5   = 000040 G    C$GMAN= 000043     EM0103  005262 G     ENDD    035434       G$DISP= 000003
BIT6   = 000100 G    C$GPHR= 000042     EM1601  005320 G     ENDETB  003650 G     G$EXCP= 000400
BIT7   = 000200 G    C$GPLO= 000030     EM4001  005403 G     ENDIT   020650       G$HILI= 000002
BIT8   = 000400 G    C$GPRI= 000040     EM4002  005435 G     ERCNTB  002614 G       .OLI= 000001
BIT9   = 001000 G    C$INIT= 000011     EM4101  005471 G     ERLTBL  002650 G     G$NO   = 000000
BMPCQB  002412 G     C$INLP= 000020     EM4102  005523 G     ERRBLK  004056 G     G$OFFS= 000400
BMPCQE  002612 G     C$MANI= 000050     EM4103  005557 G     ERRMSG  004054 G     G$OFSI= 000376
BMPCQP  002410 G     C$MEM  = 000031    EM4201  005643 G     ERRNBR  004052 G     G$PRMA= 000001
BOE    = 000400 G    C$MSG  = 000023    EM4202  005675 G     ERRTYP  004050 G     G$PRMD= 000002
BRLEVL  002223 G     C$OPEN= 000034     EM4901  005717 G     ERSMRF  002612 G     G$PRML= 000000
BUFBAS  002650 G     C$PNTB= 000014     EM4902  005760 G     ER0101  011430 G     G$RADA= 000140
BUFEND  003650 G     C$PNTF= 000017     EM5001  006012 G     ER0503  011762 G     G$RADB= 000000
BUFMID  003250 G     C$PNTS= 000016     EM5101  006051 G     ER1603  012020 G     G$RADD= 000040
BUFPTR  002246 G     C$PNTX= 000015     EM5102  006106 G     ER5801  012112 G     G$RADL= 000120
BUF3QT  003450 G     C$QIO  = 000377    EM5103  006144 G     ER5901  012170 G     G$RADO= 000020
CALMSL  013552 G     C$RDBU= 000007     EM5201  006174 G     ER6001  012276 G     G$XFER= 000004
CHKBMP  013776 G     C$REFG= 000047     EM5202  006227 G     ER6401  012400 G     G$YES  = 000010
CKRXTM  014046 G     C$RESE= 000033     EM5301  006265 G     ER7801  012506 G     HELP   = 000000
CKTRAP  014166 G     C$REVI= 000003     EM5302  006321 G     ER9001  012544 G     HOE    = 100000 G
CLKBRL  002306 G     C$RFLA= 000021     EM5303  006401 G     ER9002  012644 G     HWPTQ1  035222
CLKCSR  002304 G     C$RPT  = 000025    EM5401  006452 G     ER9004  013022 G     HWPTQ2  035240
CLKHRZ  002312 G     C$SEFG= 000046     EM5402  006520 G     ER9101  013136 G     HWPTQ3  035273
CLKINT  017632 G     C$SPRI= 000041     EM5501  006561 G     ER9301  013176 G     HWPTQ4  035321
CLKVEC  002310 G     C$SVEC= 000037     EM5601  006625 G     EVL    = 000004 G     IBE    = 010000 G
CLRRST  014216 G     C$TPRI= 000013     EM5701  006702 G     EXDERR  002252 G     IDU    = 000040 G
CLR16W  014240 G     DELAY   014262 G   EM5801  006757 G     E$END  = 002100       IFF    = 020000 G
CLRA    002226 G     DFPTBL  002176 G   EM5802  007003 G     E$LOAD= 000035       IFSTAT  002256 G
CSRQ   = 000000 G    DIAGMC= 000000     EM5803  007077 G     FDATA   002234 G     INDATP  014550 G
```

SYMBOL TABLE

```
INDTPX  014600 G        L$EXP4  002064 G        L10034  024714          PRIO6 = 000300 G        TXBFCA  002244 G
ISR     = 000100 G      L$EXP5  002066 G        L10035  025406          PRIO7 = 000340 G        TXBFCO= 000016 G
IXE     = 004000 G      L$HARD  035156 G        L10036  025714          PRTLPR  015306 G        TXDATP  017004 G
I$AU    = 000041        L$HIME  002120 G        L10037  026224          PUFIFO  015370 G        TXDSBL  017026 G
I$AUTO= 000041          L$HPCP  002016 G        L10040  026724          PUFIFR  015452 G        TXENBL  017122 G
I$CLN = 000041          L$HPTP  002022 G        L10041  027422          RBUFA   002230 G        TXIEO   017216 G
I$DU  = 000041          L$HW    002176 G        L10042  030100          RBUFO = 000002 G        TXIE1   017256 G
I$HRD = 000041          L$ICP   002104 G        L10043  031150          READBX  015654 G        TXINTC  002272 G
I$INIT= 000041          L$INIT  020044 G        L10044  031736          REPDER  015736 G        TXINTF  002274 G
I$MOD = 000041          L$LADP  002026 G        L10045  032764          REPSMR  016254 G        TXRLNB  004010 G
I$MSG = 000041          L$LAST  035706 G        L10046  033410          RESETT  016302 G        TXRLNE  004030 G
I$PROT= 000040          L$LOAD  002100 G        L10047  034126          RXBDTX= 000030 G        TXRXLB  003750 G
I$PTAB= 000041          L$LUN   002074 G        L10050  035072          RXBETX= 000020 G        TXRXLE  004010 G
I$PWR = 000041          L$MREV  002050 G        L10051  035152          RXBFUL= 000100 G        TXVECA  002216 G
I$RPT = 000041          L$NAME  002000 G        L10052  035222          RXCNTB  003710 G        T$ARGC= 000002
I$SEG = 000041          L$PRIO  002042 G        L10053  035434          RXDECT  017702 G        T$CODE= 001052
I$SETU= 000041          L$PROT  020036 G        MAPCNT  014676 G        RXIEO   016414 G        T$ERRN= 022125
I$SFT = 000041          L$PRT   002112 G        MAPLNS= 177777 G        RXIE1   016454 G        T$EXCP= 000000
I$SRV = 000041          L$REPP  002062 G        MFUNIT  005231 G        RXINTC  002266 G        T$FLAG= 000040
I$SUB = 000041          L$REV   002010 G        MMENAB  002334 G        RXINTF  002270 G        T$GMAN= 000000
I$TST = 000041          L$RPT   020030 G        MMPRES  002332 G        RXTIMO= 000002 G        T$HILI= 177777
J$JMP = 000167          L$SOFT  035404 G        MMSRO   002330 G        RXTMA   002230 G        T$LAST= 000001
LGRP1M  002260 G        L$SPC   002056 G        MSG1    011546 G        RXVECA  002214 G        T$LOLI= 000000
LGRP2M  002262 G        L$SPCP  002020 G        MSG2    011624 G        ROSLOT= 000002 G        T$LSYM= 010000
LINBIT  014650 G        L$SPTP  002024 G        MSG3    011703 G        R1SLOT= 000004 G        T$LTNO= 000024
LNCTRA  002236 G        L$STA   002030 G        MSLCNT  002326 G        R2SLOT= 000006 G        T$NEST= 177777
LNCTRO= 000010 G        L$SW    002210 G        MSLGET  014730 G        R3SLOT= 000010 G        T$NSO = 000000
LOE     = 040000 G      L$TEST  002114 G        MSLOOP  015044 G        R4SLOT= 000012 G        T$NS1 = 000005
LOPBCK  002222 G        L$TIML  002014 G        MSTICK  002324 G        R5SLOT= 000014 G        T$PTNU= 000000
LOT     = 000010 G      L$UNIT  002012 G        NDERPT  002212 G        SAVBMP  016500 G        T$SAVL= 177777
LPCSLT= 000036 G        L10000  002206          NEWPAS  020426          SETPAR  016546 G        T$SEGL= 177777
LPRA    002232 G        L10001  002214          NEWRES  020420          SFPTBL  002210 G        T$SUBN= 000000
LPRO    = 000004 G      L10002  011544          NEWSTA  020110          SKPSTS  016614 G        T$TAGL= 177777
L$ACO   002110 G        L10003  012016          NUMLNS= 000020 G        STGTRB  004030 G        T$TAGN= 010054
L$APT   002036 G        L10004  012110          OOPS    015060 G        SVCGBL= 000000          T$TEMP= 000000
L$AU    021014 G        L10005  012166          OPTION  002210 G        SVCINS= 000001          T$TEST= 000024
L$AUT   002070 G        L10006  012274          O$APTS= 000000          SVCSUB= 000001          T$TSTM= 177777
L$AUTO  020664 G        L10007  012376          O$AU  = 000000          SVCTAG= 000001          T$TSTS= 000001
L$CCP   002106 G        L10010  012504          O$BGNR= 000001          SVCTST= 000001          T$$AU  = 010025
L$CLEA  020666 G        L10011  012542          O$BGNS= 000001          SWPTQ1  035434          T$$AUT= 010022
L$CO    002032 G        L10012  012642          O$DU  = 000001          SWPTQ2  035510          T$$CLE= 010023
L$DEPO  002011 G        L10013  013020          O$ERRT= 000001          SWPTQ3  035543          T$$DU  = 010024
L$DESC  004130 G        L10014  013134          O$GNSW= 000001          S$LSYM= 010000          T$$HAR= 010052
L$DESP  002076 G        L10015  013174          O$POIN= 000001          TIMER1  002314 G        T$$HW  = 010000
L$DEVP  002060 G        L10016  013366          O$SETU= 000000          TIMER2  002316 G        T$$INT= 010021
L$DISP  002124 G        L10017  020034          PASCNT  002264 G        TIMER3  002320 G        T$$MSG= 010016
L$DLY   002116 G        L10021  020662          PCSLOT= 000016 G        TNUM  = 000024 G        T$$PRO= 010020
L$DTP   002040 G        L10022  020664          PNT   = 001000 G        TP4FLG  002300 G        T$$RPT= 010017
L$DTYP  002034 G        L10023  020702          PREGRT  004102 G        TP4RTN  017712 G        T$$SOF= 010053
L$DU    020704 G        L10024  021012          PREG05  004060          TP4VEC  002276 G        T$$SW  = 010001
L$DUT   002072 G        L10025  021020          PRI   = 002000 G        TSABRT  016672 G        T$$TES= 010051
L$DVTY  004120 G        L10026  021302          PRIO0 = 000000 G        TSTNUM  002254 G        T1      021022 G
L$EF    002052 G        L10027  021706          PRIO1 = 000040 G        TXAD1A  002240 G        T10     025716 G
L$ENVI  002044 G        L10030  022344          PRIO2 = 000100 G        TXAD10= 000012 G        T11     026226 G
L$ERRT  004050 G        L10031  022672          PRIO3 = 000140 G        TXAD2A  002242 G        T12     026726 G
L$ETP   002102 G        L10032  023472          PRIO4 = 000200 G        TXAD20= 000014 G        T13     027424 G
L$EXP1  002046 G        L10033  024272          PRIO5 = 000240 G        TXAINT  017734 G        T14     030102 G
```

SYMBOL TABLE

T15	031152 G	T20	035074 G	T8	024716 G	WAIBIS	017436 G	X$ALWA=	000000
T16	031740 G	T3	021710 G	T9	025410 G	WAITTX	017512 G	X$FALS=	000040
T17	032766 G	T4	022346 G	UAM =	000200 G	WORD1	002302 G	X$OFFS=	000400
T18	033412 G	T5	022674 G	UNITN	002224 G	WTWLNC	017552 G	X$TRUE=	000020
T19	034130 G	T6	023474 G	UNSDIV	017302 G	WTWLPR	017602 G	$PATCH	035632 G
T2	021304 G	T7	024274 G						

. ABS. [illegible] 000
[illegible] 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28661 WORDS (112 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:05:03
CZDHVB0,BIN,CZDHVB0,LST/-SP=SVC34R/ML,CZDHVB0,P11
.

SYMBOL TABLE B1