GS03WD GS3WD/DZ11 LGC DIAG
CZDZGA0

AH-T919A-MC
FICHE 01 OF 01

FEB 1985
digital
Made In USA

```
                        .rem @

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

                        PRODUCT CODE:   AC-T918A-MC

                        PRODUCT NAME:   CZDZGA0 GS3WD/DZ11 LGC DIAG

                        PRODUCT DATE:   JULY 1984

                        MAINTAINER:     CSS ANNECY

                        AUTHOR:         Jean-Christophe PINASA
```

PROGRAM DOCUMENT

Table of contents

PROGRAM DOCUMENT

```
101
102
103
104
105
106                             1.0 Introduction
107
108                             1.1 Program abstract
109
110
111                             This diagnostic was designed to test the GS03-WD LOGIC MODULE.
112
113                             The program was implemented using the Diagnostic Supervisor.
114
115                             Through dialogue with the operator, it will allow modification of
116                             device parameters, such as :
117
118                                     - UNIBUS address ;
119                                     - vector address ;
120                                     - priority level ;
121                                     - # of lines connected out of the DZ11 into the GS03-WD ;
122                                     - operating mode  (0 ->  harware test ; 1 -> installation
123                             test).
124
125                             WARNING : RUNNING THIS DIAGNOSTIC WILL CAUSE THE GS03  TO  SWITCH
126                                       LINES BETWEEN COMPUTERS.
```

1.2 Hardware description :

2P-M213A-00 is the part number for the GS03-WD logic module.

The GS03-WD option enables an asynchronous serial line mounted in each computer to control a GS03 installation. It is supported on the DZ11 on PDP11's and DMF32 on VAX'es.

1.3 Hardware configuration :

The name of this diagnostic is : CZDZGA0 GS3WD/DZ11 LGC DIAG

The filename is : ZDZGA0.BIN

It will run in stand alone without any operator intervention, in either of the following modes :

- Diagnostic test (mode 0)

This part will check all the GS03-WD hardware and the GS03 functionality.

- Installation test (mode 1)

It will allow by visual inspection to check site installation and system interconnection.

1.3.1 Diagnostic test (MODE 0)

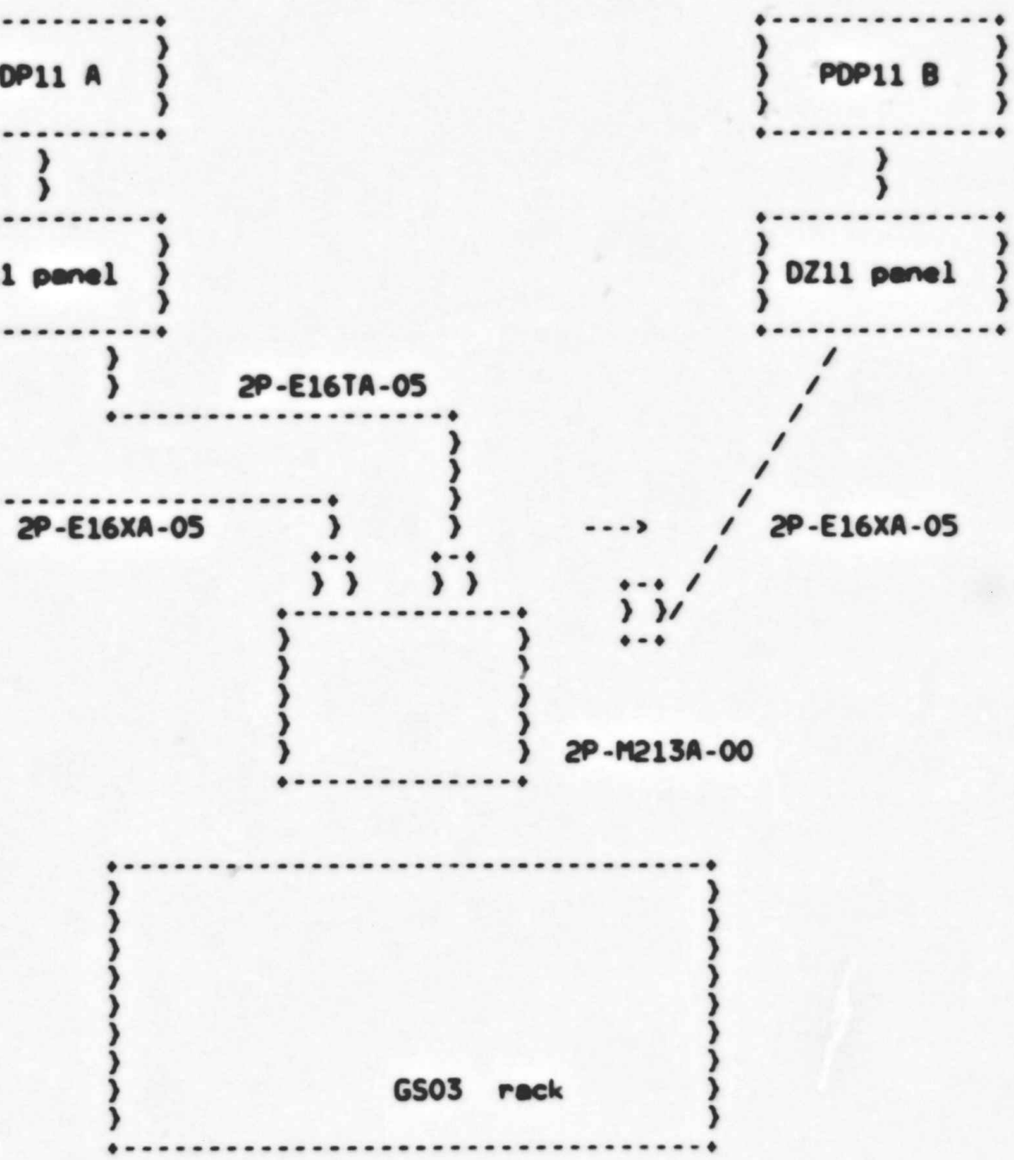
This part of the diagnostic will run on one of the two PDP11's only and test all of the GS03-WD hardware.

Before running this part of the diagnostic, operators will have to remove the 2P-E16XA-05 cable connected to the PDP11 that is not being used. (Disconnect the cable from the 2P-M213A-00 module in the GS03 rack).

A special "Diag test cable" 2P-E16TA-05, will have to be plugged from the 2P-M213A module into the chosen PDP11 DZ11.

PROGRAM DOCUMENT

Example: Diagnostic running on PDP11 A.



PROGRAM DOCUMENT

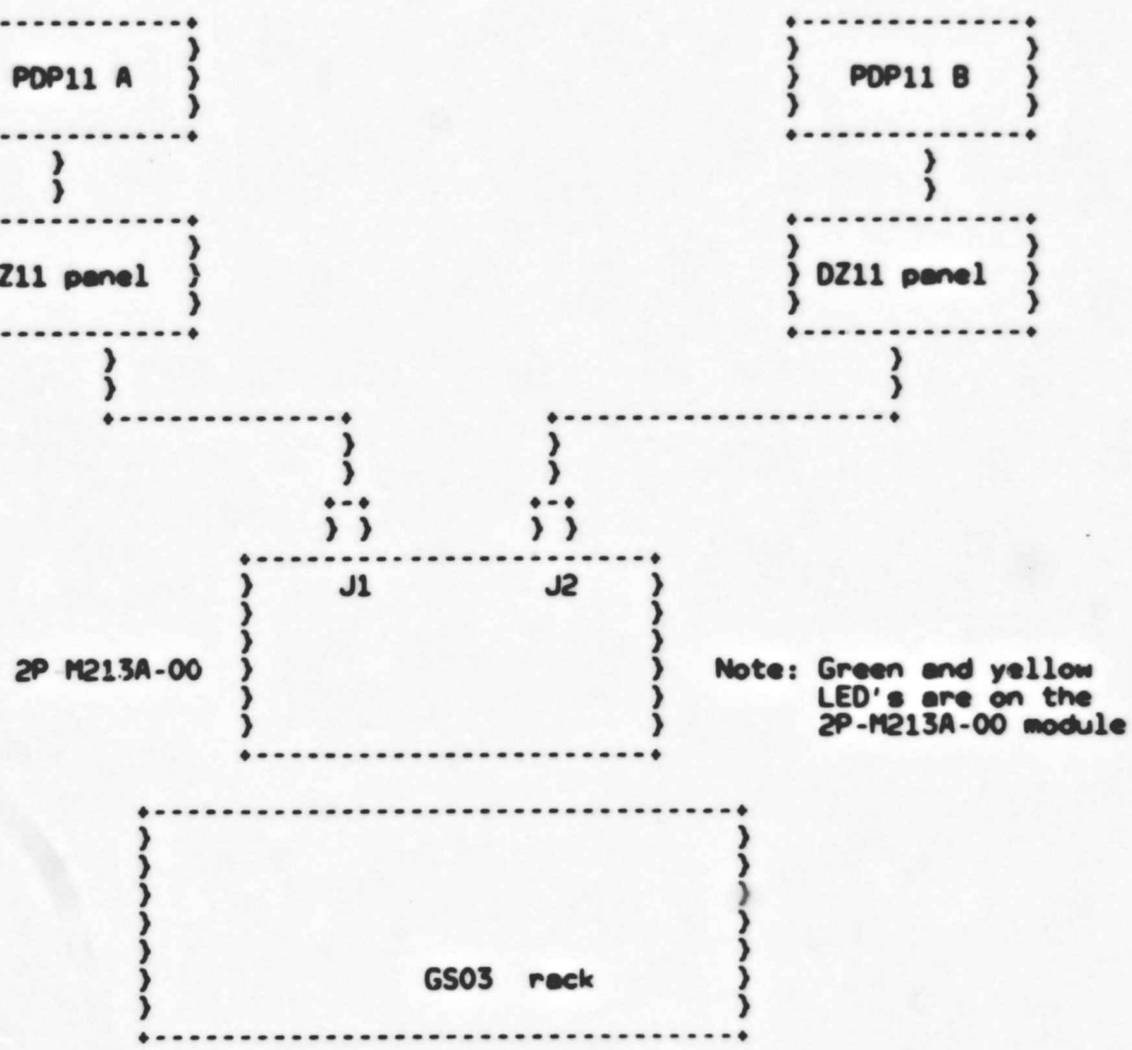
```
227
228
229
230
231                     Before running this part of the diagnostic, the operator will have  to
232                     go through the following checklist :
233
234
235                     -  Disable the highest priority commands by :
236
237                     o Placing the FORCE AB switch to the center position on all racks
238
239                     o Placing all MANUAL switches in the center position on all racks
240
241
242                     -  Remove the "2P-E16XA-05" on the unused PDP11 side.
243
244
245                     -  Connect the "2P-E16TA-05" diag test  cable  from  the  2P-M213A-00
246                     module to the PDP11 DZ11 in use.
247
248
249                     -  Check that the dip switch E18-1 on the 2P-M213A-00 module is off.
250
251
252                     -  Run the ZDZGA0 diagnostic on the chosen system (select mode 0).
253                     See 6.1.3.
254
255
256                     -  When finished, reconfigure the system.
257
```

PROGRAM DOCUMENT

1.3.2 Installation test (MODE 1)

This test will allow to check GS03-WD installation and cable interconnection.

No modification of the installation is required to run this part of the test.



PROGRAM DOCUMENT

The diagnostic can be run on one of the systems or on both of them at the same time.

It will send frames to the corresponding channel of the GS03-WD.

The operator will check test result by watching the LED indicators on the 2P-M213A-00 module.

- Running ZDZGA0 in mode 1 on system A (system connected to 2P-M213A-00 on J1) will make the "green" LED (on 2P-M213A-00) blink.

- Running ZDZGA0 in mode 1 on system B (system connected to 2P-M213A-00 on J2) will make the "yellow" LED (on 2P-M213A-00) blink.

- Running it on both systems will cause both "green" and "yellow" leds to blink.

CAUTION

This test will run continously and will have to be stopped by typing "cntrl C" on the console.

1.4 Diagnostic description :

This diagnostic will first test UNIBUS access to the DZ11 CSR's. It will then check very roughly the transmit and receive functions in maintenance loopback mode.

Depending upon the mode it is run in, the next actions taken by the diagnostic will be :

Mode 0 :

- a first try at receiving echo back from the GS03-WD on either line ;

- a test of correct switching of the GS03-WD back and forth.

Mode 1 :

Activation of the line into the GS03-WD by sending characters over it continuously.

```
368
369
370
371
372
373                         2.0 Hardware requirements
374
375                         The  following  hardware is required to run the static logic
376                         tests on module GS03-WD :
377
378                         Any member of the PDP-11 UNIBUS family (PDP11/24, 34, 44, 70) ;
379                         16k memory ;
380                         console terminal.
381
382                         WARNING :
383                         This  diagnostic  will  not  run on  any  member  of the VAX
384                         family, although a DZ11 may be fitted on a VAX UNIBUS. It is
385                         reminded  that a GS03-WD logic module should be connected to
386                         a VAX through a DMF32.
387
388
389                         3.0 Preliminary program requirements
390
391                         The processor, memory  and  the  DZ11  should  be thoroughly
392                         tested prior to running this diagnostic.
393
394
395                         4.0 General program considerations
396
397
398                         4.1 Diagnostic Supervisor
399
400                         This  program  is  written to run under the PDP11 diagnostic
401                         supervisor.
402                         It requires 16k of memory to run.
403
404
405                         4.2 Execution Time
406
407                         The  total  time  required  to run the GS03-WD static  diag-
408                         nostic ranges from about 2 minutes on the  PDP11/70 to about
409                         4 minutes on the  PDP11/34 per  pass for each unit (with su-
410                         pervisor version c4).
411
412
413                         4.3 XXDP+
414
415                         This  program will be loaded under XXDP+, and may  be run in
416                         dump mode.
417
418
419                         4.4 Memory management
420
421                         Memory management is not enabled by this program.
422
423
424                         5.0 Program load media
```

PROGRAM DOCUMENT

```
425
426                    This program can be loaded from any media supported by
427                    XXDP+. The diagnostic supervisor will be loaded first, fol-
428                    lowed by the diagnostic program.
```

6.0 Operating instructions

6.1 Loading and starting procedures

6.1.1 Loading procedures

When loaded under XXDP+, the diagnostic supervisor will be loaded automatically.

6.1.2 Starting procedures

The program starts at location 200. Use standard DEC procedures to start the program.

6.1.3 Steps for quick and simple execution

The diagnostic can be executed standalone under XXDP+ without reading the remainder of this document, as follows:

a) load and start diagnostic using run command ;
b) receive diagnostic supervisor prompt (DR>) ;
c) enter STA<CR> ;
d) answer hardware questions ;
e) get end of pass messages or error messages ;
f) to end execution, enter control/c.

PROGRAM DOCUMENT

```
                                DIAG.  RUN-TIME SERVICES
                                ZDZGA0-A-0
                                CZDZGA0 GS3WD.DZ11 LGC DIAG
                                UNIT IS GS03WD MODULE
                                RESTART ADDR: 147670
                                DR>START

                                CHANGE HW (L) ? Y

                                # UNITS (D) ? 1

                                UNIT 0
                                CSR (O)  160100 ? 160340    ; The CSR address is 160340 (range =
                                                            ; 160010-163776)
                                VECTOR (O)  300 ? 460       ; Vector address is 460 (range = 300-777)
                                BR (O)  5 ? 6               ; BR interrupt level is 6 (range = 4-7)
                                ACTIVE LINES (O)  3 ? <CR>  ; Defines the line(s) of the DZ11 connected
                                                            ; to the GS03-WD (octal bitmap format :
                                                            ; range = 0-377)
                                                            ; Here (default value) : lines 0 and 1
                                WHICH MODE (O)  0 ? <CR>    ; mode 0 = hardware test
                                                            ;          caution : connect cables as
                                                            ;          described in the diagnostic
                                                            ;          header and in the option desc.
                                                            ; mode 1 = installation test
                                                            ;          with visual inspection of LED's
                                                            ;          caution : in this mode, the
                                                            ;          diagnostic will run continuous-
                                                            ;          ly. To stop it, type "ctrl C".
                                                            ; See header or option description for
                                                            ; more details.
Running on unit  0 in mode 0 : pass-time is 2 minutes on the PDP11/70.
Only tests 1, 2 and 3 are active in this mode.


                        Example: Running "CZDZGA0 GS3WD/DZ11 LGC DIAG"
```

PROGRAM DOCUMENT

```
507
508
509
510
511
512                          6.2 Initial dialogue
513
514                          After the program and the  supervisor  are  loaded  and  the
515                          program  is started , the following identification is typed:
516
517                          DIAG.  RUN-TIME SERVICES
518                          ZDZGA0-A-0
519                          CZDZGA0 GS3WD.DZ11 LGC DIAG
520                          UNIT IS GS03WD MODULE
521                          RESTART ADDR: 147670
522                          DR>
523
524                          The operator then proceeds by typing  one  or  more  of  the
525                          commands  described  in  the following section 6.3.(for more
526                          detailed information, refer  to  the  diagnostic  supervisor
527                          functional specification).
```

6.3 Program options

6.3.1 START command

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

6.3.1.1 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.) that specify the tests to be executed. The numbers are seperated by colons. The numbers range from 1 to the largest test number in the diagnostic. They may be specified in any order. 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 6.3.1.5.

6.3.1.2 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) against all units submitted. The default is non-ending execution. In this case exit from the program is accomplished either by typing a control/c or by occurence of an error with the halt on error flag being set. The exit is a return to command mode. See example at end of 6.3.1.5.

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

<FLAG-LIST> is a sequence of elements of the form <FLAG>, <FLAG=1>, or <FLAG=0>, seperated 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

```
586                                     *  PRI   direct all messages to a line printer
587                                        PNT   print number of test being executed
588                                        BOE   bell on error
589                                     ** UAM   run in unattended mode, bypassing manual
590                                              intervention tests
591                                     ** ISR   inhibit statistical reports
592                                        ADR   execute autodrop code
593                                        IDU   inhibit dropping of units by diagnostic
594                                        LOT   loop on test
595                                     ** EVL   evaluate
596
597                                     *  NOT TO BE USED if a line printer is not available.
598                                     ** Of no use in this diagnostic.
599
600                                     The flags named or equated to 1 are set, those equated to  0
601                                     are cleared.  A flag not specified is cleared.  If the flags
602                                     switch is not given all flags are cleared.  See  example  at
603                                     end of 6.3.1.5.
604
605
606                                     6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)
607
608                                     <INCR> is a decimal number indicating how often (in terms of
609                                     passes) it is desired  that  the  end  of  pass  message  be
610                                     printed.   The  default  is  at  the end of every pass.  See
611                                     example at end of 6.3.1.5.
612
613
614                                     6.3.1.5 Effect of start command
615
616                                     The effect of the start command is to initiate the  hardware
617                                     parameter  dialogue,  the  software  parameter dialogue, and
618                                     then the diagnostic tests themselves.
619
620                                     The hardware  parameter dialogue  starts  with  the question
621                                     "# units?"  to  which  the  operator replies with a decimal
622                                     number n from 1 to 16. The term "unit" refers to the  device
623                                     to which this series of diagnostics is dedicated.  Following
624                                     this are the questions whereby the p-tables themselves  will
625                                     be  built.  Each p-table is a core-resident table containing
626                                     all the hardware information for  one  unit.  The  operator
627                                     must  supply  n  (number of units) values for each question.
628                                     He may do this by giving one answer  to  each  question  (in
629                                     which case the series of questions will be posed n times) or
630                                     by giving n values, separated by commas,  to  each  question
631                                     (series  will  be posed once).  Each question is followed by
632                                     the response radix (d for  decimal,  b  for  binary,  o  for
633                                     octal,  l  for  yes/no) in parentheses and the default value
634                                     after the parentheses.
```

Following the hardware questions are the software questions to build the software tables, which define the mode (quick verify etc.) that the diagnostic will execute in.

When the question "# units?" is answered, memory storage is allocated for the p-tables, and if there is not enough to accommodate them the message "TOO MANY UNITS" is issued. In this case the diagnostic must be executed more than once to test all units.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

This command will cause three passes to be made, each pass consisting of tests 1,2,3,4,6,8,9, and 10 executed against all units. 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.

6.3.2 RESTART command

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

6.3.2.1 TESTS, PASS, and FLAGS switches

<TEST-LIST>, <PASS-CNT>, and <FLAG-LIST> are as in the START command.

6.3.2.2 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 dialogue. 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.

```
692
693
694
695
696
697                     6.3.2.3 Effect of RESTART command
698
699                     The RESTART command differs from the START command  in  that
700                     the  p-tables  from  the  previous start command (there must
701                     have been one) are used, instead of new  ones  being  built.
702                     The  units  switch  gives  the ability to select a subset of
703                     these.  The software dialogue may optionally  be  reexecuted
704                     (operator  will  be  asked).  The  command can be used after
705                     command mode has been reentered in any of the  three  normal
706                     ways:   a)  the requested number of passes have been made b)
707                     an error was encountered with the halt on error flag set  c)
708                     a "control/c" was entered by the operator.
709
710
711                     6.3.3 CONTINUE command
712
713                     ************************************************************
714                     CON(TINUE)/PASS:<PASS-CNT>/FLAGS:<FLAG-LIST>
715                     ************************************************************
716
717
718                     6.3.3.1 PASS switch (/PASS:<PASS-CNT>)
719
720                     <PASS-CNT>  is  same as in START command, but the default is
721                     the unsatisfied pass-cnt from the previous START or RESTART.
722                     If none remains, the default is non-ending execution.
723
724
725                     6.3.3.2 FLAG switch (/FLAGS:<FLAG-LIST>)
726
727                     <FLAG-LIST>  is  same  as  in START command, but unspecified
728                     flags retain their current value.
729
730
731                     6.3.3.3 Effect of CONTINUE command
732
733                     CONTINUE must follow a start or restart,  and  command  mode
734                     must  have  been  entered  due  to  a  halt  on  error  or a
735                     control/c. The effect  of  the  command  is  to  go  to  the
736                     beginning  of the test that was being executed when the halt
737                     or control/c took place.  Software dialogue  may  optionally
738                     be reexecuted.  Hardware parameters may not be changed.
```

```
                        6.3.4 PROCEED command

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

                        6.3.4.1 FLAGS switch (/FLAGS:<FLAG-LIST>)

                        <FLAG-LIST> is as in the START command, but unspecified
                        flags retain their current value.


                        6.3.4.2 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.


                        6.3.5 ADD command

                        ;*****************************************************************
                        ADD/UNITS:<UNIT-LIST>
                        ;*****************************************************************


                        6.3.5.1 UNITS switch (/UNITS:<UNIT-LIST>

                        <UNIT-LIST> is as in the RESTART command.


                        6.3.5.2 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.


                        6.3.6 DROP command

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


                        6.3.6.1 UNITS switch (/UNITS:<UNIT-LIST>)
```

797 <UNIT-LIST> is as in the RESTART command.
798
799
800 6.3.6.2 Effect of DROP command
801
802 The units specified will be dropped from testing. The units
803 will be reselected only by the execution of an ADD or START
804 command. The units switch must be entered. This command
805 must be followed by a RESTART or a CONTINUE command.
806
807
808 6.3.7 PRINT command : NOT IMPLEMENTED
809
810 **
811 PRI(NT)
812 **
813
814
815 6.3.7.1 Effect of PRINT command
816
817 The total number of errors for each unit since the last
818 start or restart command are printed. The ier (inhibit
819 statistical reporting) flag is cleared.
820
821
822 6.3.8 DISPLAY command
823
824 **
825 DIS(PLAY)/UNITS:<UNIT-LIST>
826 **
827
828
829 6.3.8.1 UNITS switch (/UNITS:<UNIT-LIST>)
830
831 <UNIT-LIST> is as in the RESTART command.
832
833
834 6.3.8.2 Effect of DISPLAY command
835
836 The hardware p-tables for all units under test are printed
837 out in the format in which they were entered. Any units
838 that were dropped by the operator "drop" command are so
839 designated.
840
841
842 6.3.9 FLAGS command
843
844 **
845 FLA(GS)
846 **
847
848
849 6.3.9.1 Effect of FLAGS command
850
851 The current settings of all flags are printed.

6.3.10 ZFLAGS command

****
ZFL(AGS)
****

6.3.10.1 Effect of ZFLAGS command

All flags are cleared.

6.3.11 Control Characters

A control c (c) entered during the execution of a diagnostic causes a return to command mode.

A control z (z) entered during one of the three operator dialogues -initial dialogue (see 6.2),hardware dialogue (see 6.3.1.5), or software dialogue (see 6.3.1.5)- causes the defaults to be taken for the remainder of that dialogue.

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 telutype output.

6.3.12 Hardware Parameters

The following questions will be asked on a START command. The value located to the left of the question mark is the default value that will be taken on a carriage return response.

Note :
Entering these parameters is a crucial part of running this diagnostic, which should not be overlooked.
The default values, for instance, should not be relied upon too quickly.

1. CHANGE HW (L) ?

The answer to this question has no default value.
Answering "NO" will cause all the default values to be assumed, which may be a cause for errors.

2. # UNITS (D) ?

The answer to this question has no default value either.

3. CSR (O) 160100 ?

This is the address at which the DZ11 CSR register resides on the unibus.
The allowable range is 160010..163776 (octal), and the default value is 160100.

4. VECTOR (O) 300 ?

The allowable range is 300..777, and default value is 300.

Note :
Entering a wrong value here will cause the diagnostic to stop. An "ILL INTER NN4" error message will be printed and a new value will have to be entered into the hardware p-table after issuing the "START" command again.

```
5. BR (0)  5 ?

The allowable range is 4..7 and the default value is 5.

6. ACTIVE LINES (0) 3 ?

This asks for a bitmap of the  line(s) out of the DZ11 into
the GS03-WD. When running in mode 0, two lines will be nee-
ded and when running in mode 1, only one.

The allowable range is 0..377  and  the default  value is 3
(lines 0 and 1).

Note :
The DRS, which  asks these  questions, only checks that the
number specified is in the range 0-377.
The  diagnostic initialization  code  checks that two lines
are specified for mode  0 operation and  1 line  for mode 1
operation. If an incorrect  number of  lines  is specified,
the diagnostic will report this as an error.
Such an error will mean having to issue the "START" command
again.

7. WHICH MODE (0) 0 ?

The allowable range is 0..1 and the default value is 0.
```

6.3.13 Software Parameters

No software parameter question is asked in this static logic test.

6.3.14 Extended Discussion Of P-Table Dialogue

The full capability of the hardware dialogue is revealed by the following discussion of what happens internally.

As soon as the question "# units?" is answered (with the number n, say) space in core is allocated for n p-tables. All of the p-tables are of the same format, and there is a one-to one correspondence between the hardware parameter questions and the slots in the p-table format.

On the first trip thru the questions, all of the slots in all of the p-tables are filled. If the operator types in less than n explicit values in response to a particular question, these values are placed in the p-tables (one value going into the proper slot of each p-table beginning with the first p-table) until the string of values is exhausted. The last value in the string becomes the new default and is used to fill that slot in the remaining p-tables.

On subsequent trips thru the questions, the same process is carried out, except that the earliest p-table not to have received an explicit value in any of its slots now assumes the role that table number one played in the first trip.

The series of questions is reissued until at least one question has received n explicit values from the operator.

In giving a string of values, commas without intervening values may be used to indicate a repetition of the last named value.

A string of values may be given as a range (6-10 for example). If the values represent pure numerical data, this sample range translates to the string 6,7,8,9,10 (an increment of 1). If the values are addresses, the sample range translates to the string 6,8,10 (an increment of 2).

Now let us see how we could use these capabilities to construct a set of p-tables. Assume that we have 16 units, and that there are three hardware parameters for each (three slots in the p-table, three hardware questions in the dialogue). Let the desired value for the first parameter be the number 75 for all 16 tables. Let the desired value for the second parameter be equal to the unit number (0,1,2,....,15) except for unit 12, which should receive the value 11. Let the desired value for the third parameter be the number 76 for the first 7 units and the number 77 for the last 9 units.

The following dialogue would accomplish this goal:

```
# UNITS (D) ? 16

UNIT 1
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 21
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77
```

The first time the series is asked, slot one receives a 75 in all 16 tables. Slot two receives the values 0,1,2,....,6 in tables 0 thru 6 and a constant 6 in tables 7 thru 15. Slot three receives a constant 76 in all 16 tables.

The second time thru the series, tables 16 thru the end are going to be affected (note that this piece of information is printed out for the the operator in the form "unit xx" at the beginning of each series). Question 1 is responded to by a <cr>, so slot one stays at constant 75 in tables 7 thru 15, since no new exlicit values are typed in. Slot two gets the values 7,8,9,10,11 in tables 7 thru 11, and gets a 11 in slot 12, and gets the values 13,14,15 in tables 13 thru 15. Slot three gets the value 77 in tables 7 thru 15.

The dialogue is terminated when the software recognizes that 16 explicit values have been given for at least one question (namely question 2).

PROGRAM DOCUMENT

```
1061
1062
1063
1064                                7.0  Tests Descriptions
1065
1066
1067
1068
1069                        ************************** TEST 1 ****************************
1070                        *                                                           *
1071                        *       Purpose : basic test of DZ11.                       *
1072                        *                                                           *
1073                        *       Description :                                       *
1074                        *       - Subtest 1 : Check that DZ11 CSR can be  written   *
1075                        *       to and read from ;                                  *
1076                        *       - Subtest 2 : Transmit a character in maintenance   *
1077                        *       (internal) loopback mode on the selected line(s)    *
1078                        *       and check for proper echo.                          *
1079                        *                                                           *
1080                        *       Error messages :                                    *
1081                        *  #0,1  - Subtest 1 : "Unsuccessful attempt to             *
1082                        *          write to/read DZ11 CSR at address <address>"     *
1083                        *                      "Check DZ11 address."                *
1084                        *                                                           *
1085                        *  #2    - Subtest 2 : "DZ11 failed to reset."              *
1086                        *                      "Check DZ11 address."                *
1087                        *                      "Run DZ11 diagnostic."               *
1088                        *                                                           *
1089                        *  #3    - Subtest 2 : "DZ11 internal loopback malfunction  *
1090                        *        on line # <line number>"                           *
1091                        *                      "Check DZ11 address."                *
1092                        *                      "Run DZ11 diagnostic."               *
1093                        *                                                           *
1094                        *************************************************************
```

```
1096
1097
1098
1099
1100
1101                    ********************** TEST 2 **********************
1102                    *                                                  *
1103                    * Test active only in mode 0 :                     *
1104                    *                                                  *
1105                    *      Purpose : check that  characters  are echoed back *
1106                    *                from the GS03-WD.                 *
1107                    *                                                  *
1108                    *      Assumption :  the previous test ran successfully. *
1109                    *                                                  *
1110                    *      Description :                               *
1111                    *      The  two  lines  out of the DZ11  are arbitrarily *
1112                    *      named line x and line y.                    *
1113                    *      A first  attempt will be  made  to  receive  echo *
1114                    *      back from  the GS03-WD  on  line  x. If it is not *
1115                    *      successful,  another  attempt  will  be  made  to *
1116                    *      receive  echo  on  line  y. If  this  cannot  be *
1117                    *      achieved  either, a  hard  error  warning  will be *
1118                    *      printed.                                    *
1119                    *                                                  *
1120                    *      Note :                                      *
1121                    *      This diagnostic detects that the GS03-WD switches *
1122                    *      to one line  by receiving  echoed characters back *
1123                    *      from the GS03-WD on that line.              *
1124                    *      This is why,  before  other  tests  check correct *
1125                    *      switching, this test first checks that  echo  can *
1126                    *      be received back from the  GS03-WD, on  at  least *
1127                    *      one line.                                   *
1128                    *                                                  *
1129                    *      Error message :                             *
1130                    * 04   - "No echo received back from the GS03-WD on  *
1131                    *      either line # <line number> or # <line number>"  *
1132                    *        "Check cabling and dip switch E18 (must be OFF)"*
1133                    *                                                  *
1134                    ********************************************************
```

```
1136
1137
1138
1139
1140
1141                    ************************** TEST 3 **************************
1142                    *                                                          *
1143                    * Test active only in mode 0 :                             *
1144                    *                                                          *
1145                    *       Purpose : switch the GS03-MD back and forth.       *
1146                    *                                                          *
1147                    *       Assumptions :                                      *
1148                    *       - all previous tests ran successfully ;            *
1149                    *       - WATCHDOG FUNCTION has priority (cf. note).       *
1150                    *                                                          *
1151                    *       Description :                                      *
1152                    *       This  test is the implementation of the following  *
1153                    *       algorithm :                                        *
1154                    *                                                          *
1155                    *       Repeat  twice,  swapping  lines x and y, the se-   *
1156                    *       quence :                                           *
1157                    *              - Try and switch GS03-MD to line x ;        *
1158                    *              - Try and switch GS03-MD from  line x  to   *
1159                    *              line y ;                                    *
1160                    *              - Try and switch GS03-MD back from line y   *
1161                    *               to line x ;                                *
1162                    *                                                          *
1163                    *       Note :                                             *
1164                    *       This diagnostic assumes that the switches are set  *
1165                    *       to  give  the  WATCHDOG  FUNCTION  priority. This  *
1166                    *       means that the front panel switches should all be  *
1167                    *       in  the  center  position and  the relay  modules   *
1168                    *       should all be configured for  the  same  priority  *
1169                    *       (see Option Description for details).              *
1170                    *                                                          *
1171                    *       IMPROPER SETTINGS  CAN  CAUSE  UNEXPECTED ERRORS,  *
1172                    *       WHICH WILL NOT NECESSARILY  BE DIAGNOSED AS SUCH.  *
1173                    *                                                          *
1174                    *       Error messages :                                   *
1175                    *  05   - "No echo received back from GS03-MD on line      *
1176                    *          # <line number>"                                *
1177                    *         "Check FORCE, MANUAL switches, priority          *
1178                    *          setting and cables".                            *
1179                    *  06   - "Both lines have switch priority over each       *
1180                    *          other."                                         *
1181                    *         "Check GS03 configuration."                      *
1182                    *  07   - "Echo from GS03-MD received on both lines        *
1183                    *          # <line number> and # <line number>."           *
1184                    *  08   - "GS03-MD failed to switch to line # <line        *
1185                    *          number>"                                        *
1186                    *         "No echo received back from GS03-MD on line      *
1187                    *          # <line number>"                                *
1188                    *         "Check FORCE, MANUAL switches, priority          *
1189                    *          setting and cables".                            *
1190                    *  09   - "Echo from the GS03-MD received on wrong         *
1191                    *          line # <line number> (expected : # <line        *
1192                    *          number>)."                                      *
```

PROGRAM DOCUMENT

```
1193                    *       "Echo is still being received on line #         *
1194                    *        <line number> when actually transmitting       *
1195                    *        on line # <line number> only."                 *
1196                    *       "Check GS03 configuration."                     *
1197                    *                                                       *
1198                    *********************************************************
```

PROGRAM DOCUMENT

```
1200
1201
1202
1203
1204
1205                    ********************** TEST 4 **********************
1206                    *                                                  *
1207                    * Test active only in mode 1 :                     *
1208                    *                                                  *
1209                    *       Purpose : installation test.               *
1210                    *                                                  *
1211                    *       Assumption : all previous tests ran successfully. *
1212                    *                                                  *
1213                    *       Description :                              *
1214                    *       This  test activates the line into the GS03-WD in *
1215                    *       order for the  operator to  check  that the LED's *
1216                    *       react correctly :                          *
1217                    *                                                  *
1218                    *       The  GREEN  or YELLOW  LED  corresponding to this *
1219                    *       CPU's line into the GS03-WD should  then turn on. *
1220                    *       The associated RED LED should turn off  after one *
1221                    *       full GS03-WD clock pulse after  this test  begins *
1222                    *       (which means that the RED  clock LED should blink *
1223                    *       twice at the most before this happens).    *
1224                    *                                                  *
1225                    *       Error message : none.                      *
1226                    *                                                  *
1227                    ****************************************************
```

8.0 Error Information

8.1 Error Reporting

Errors are reported by the program as they occur (if not inhibited). The report conforms to the diagnostic supervisor error report format, and consists of a description of the error, the test number, subtest number, pc of the error call, device address, and basic and extended error information.

The following examples provide typical error reports:

```
-----------------------------------------------------------------
ZDZGA0 DVC FTL ERR  00000 ON UNIT 00 TST 001 SUB 001 PC: 010052
BUS TIMEOUT

Unsuccessful attempt to write to DZ11 CSR at address 160100
Check DZ11 address.
-----------------------------------------------------------------

-----------------------------------------------------------------
ZDZGA0 HRD ERR  00005 ON UNIT 00 TST 003 SUB 000 PC: 011046
NO ECHO ON ONE LINE

No echo received back from GS03-WD on line # 0
Check FORCE, MANUAL switches, priority setting and cables.
-----------------------------------------------------------------

-----------------------------------------------------------------
ZDZGA0 HRD ERR  00008 ON UNIT 00 TST 003 SUB 000 PC: 011420
FAIL TO SWITCH TO

GS03-WD failed to switch to line # 2
No echo received back from GS03-WD on line # 2
Check FORCE, MANUAL switches, priority setting and cables.
-----------------------------------------------------------------
```

For all other errors, the report may be more extensive and require additional data to be reported.

PROGRAM DOCUMENT

```
1277
1278
1279
1280
1281
1282                                    9.0     History
1283
1284                                        -   1rst release : JULY 84
1285
1286                                    0
1287
1288
1289
```

```
PROGRAM DOCUMENT

                002000                          .=2000
                                                .MCALL  SVC
                002000                          SVC                             ; INITIALIZE SUPERVISOR MACROS
                002000                          BGNMOD  ZDZGA0
                000000                  $LSTIN= 0
                000000                  $LSTTAG= 0
                000000                  SVCINS= 0       ; LIST INSTRUCTIONS, SHIFTED RIGHT
                000000                  SVCTST= 0       ; LIST TEST TAGS, SHIFTED RIGHT
                000000                  SVCSUB= 0       ; LIST SUBTEST TAGS, SHIFTED RIGHT
                000000                  SVCGBL= 0       ; LIST GLOBAL TAGS, SHIFTED RIGHT
                000000                  SVCTAG= 0       ; 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.
```

PROGRAM HEADER

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

                                POINTER BGNAU, BGNDU, BGNSETUP




                                HEADER  ZDZGAO, A, 0, 270., 0
                              L$NAME::                  ;DIAGNOSTIC NAME
                                .ASCII /Z/
                                .ASCII /D/
                                .ASCII /Z/
                                .ASCII /G/
                                .ASCII /A/
                                .ASCII /0/
                                .BYTE   0
                                .BYTE   0
                              L$REV::                   ;REVISION LEVEL
                                .ASCII  /A/
                              L$DEPO::                  ;0
                                .ASCII  /0/
                              L$UNIT::                  ;NUMBER OF UNITS
                                .WORD   T$PTHV
                              L$TIML::                  ;LONGEST TEST TIME
                                .WORD   270.
                              L$HPCP::                  ;POINTER TO H.W. QUES.
                                .WORD   L$HARD
                              L$SPCP::                  ;POINTER TO S.W. QUES.
                                .WORD   0
                              L$HPTP::                  ;PTR. TO DEF. H.W. PTABLE
                                .WORD   L$HW
                              L$SPTP::                  ;PTR. TO S.W. PTABLE
                                .WORD   0
                              L$LADP::                  ;DIAG. END ADDRESS
                                .WORD   L$LAST
                              L$STA::                   ;RESERVED FOR APT STATS
                                .WORD   0
                              L$CO::
                                .WORD   0
                              L$DTYP::                  ;DIAGNOSTIC TYPE
                                .WORD   0
                              L$APT::                   ;APT EXPANSION
                                .WORD   0
                              L$DTP::                   ;PTR. TO DISPATCH TABLE
                                .WORD   L$DISPATCH
                              L$PRIO::                  ;DIAGNOSTIC RUN PRIORITY
                                .WORD   0
                              L$ENVI::                  ;FLAGS DESCRIBE HOW IT WAS SETUP
                                .WORD   0
                              L$EXP1::                  ;EXPANSION WORD
                                .WORD   0
                              L$MREV::                  ;SVC REV AND EDIT #
                                .BYTE   C$REVISION
```

PROGRAM HEADER

```
002051    003                       .BYTE   C$EDIT
002052                        L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                      .WORD   0
002054  000000                      .WORD   0
002056                        L$SPC::
002056  000000                      .WORD   0
002060                        L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  002324                      .WORD   L$DVTYP
002062                        L$REPP::                ;PTR. TO REPORT CODE
002062  000000                      .WORD   0
002064                        L$EXP4::
002064  000000                      .WORD   0
002066                        L$EXP5::
002066  000000                      .WORD   0
002070                        L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  007730                      .WORD   L$AU
002072                        L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  007650                      .WORD   L$DU
002074                        L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                      .WORD   0
002076                        L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  002156                      .WORD   L$DESC
002100                        L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                      EMT     E$LOAD
002102                        L$ETP::                 ;POINTER TO ERRTBL
002102  000000                      .WORD   0
002104                        L$ICP::                 ;PTR. TO INIT CODE
002104  005774                      .WORD   L$INIT
002106                        L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  007636                      .WORD   L$CLEAN
002110                        L$ACP::                 ;PTR. TO AUTO CODE
002110  007544                      .WORD   L$AUTO
002112                        L$PRT::                 ;PTR. TO PROTECT TABLE
002112  002122                      .WORD   L$PROT
002114                        L$TEST::                ;TEST NUMBER
002114  000000                      .WORD   0
002116                        L$DLY::                 ;DELAY COUNT
002116  000000                      .WORD   0
002120                        L$HIME::                ;PTR. TO HIGH MEM
002120  000000                      .WORD   0
1359
1360
1371
1372
1373
1374
1375    ;##############################################################################
1376
1377
1378
1379
1380
1381    ;++
1382    ; THIS TABLE IS USED BY THE RUNTIME SERVICES
1383    ; TO PROTECT THE LOAD MEDIA.
1384    ;--
1385
```

PROGRAM HEADER

```
1386 002122                                  BGNPROT
     002122                            L$PROT::
1387
1388 002122  000000                          0              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
1389 002124  177777                          -1             ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
1390 002126  177777                          -1             ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
1391
1392
1406
1407
1408 002130                                  ENDPROT
1409
```

L3

DISPATCH TABLE

```
                                .SBTTL DISPATCH TABLE

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

002130                          DISPATCH 4
002130  000004                  .WORD   4
002132                  L$DISPATCH::
002132  010006                  .WORD   T1
002134  010616                  .WORD   T2
002136  010750                  .WORD   T3
002140  011504                  .WORD   T4

                        ;==============================================================================
```

DEFAULT HARDWARE P-TABLE

```
                                        .SBTTL DEFAULT HARDWARE P-TABLE

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

                                .enabl  AMA
002142                                  BGNHW   DFPTBL
002142  000005                          .WORD   L10001-L$HW/2
002144                          L$HW::
002144                          DFPTBL::

002144  160100                          .word   160100          ; DZ11 CSR address
002146  000300                          .word   300             ; DZ11 vector address
002150  000005                          .word   5               ; interrupt priority level (5)
002152  000003                          .word   3               ; bitmap of lines out of DZ11 into GS03-WD
002154  000000                          .word   0               ; diagnostic test mode (0)
                                                                ; or installation test mode (1) selector

002156                                  ENDHW
002156                          L10001:
```

GLOBAL EQUATES SECTION

```
                                .SBTTL  GLOBAL EQUATES SECTION


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


002156                          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
                        ;
```

GLOBAL EQUATES SECTION

```
                000340          PRI07== 340
                000300          PRI06== 300
                000240          PRI05== 240
                000200          PRI04== 200
                000140          PRI03== 140
                000100          PRI02== 100
                000040          PRI01== 40
                000000          PRI00== 0
                                ;
                                ;OPERATOR FLAG BITS
                                ;
                000004          EVL==       4
                000010          LOT==      10
                000020          ADR==      20
                000040          IDU==      40
                000100          ISR==     100
                000200          UAM==     200
                000400          BOE==     400
                001000          PNT==    1000
                002000          PRI==    2000
                004000          IXE==    4000
                010000          IBE==   10000
                020000          IER==   20000
                040000          LOE==   40000
                100000          MOE==  100000
1515
1516            000340          MAXPRI  ==      340             ; Highest processor priority : 7
1517
1518            000020          CSRCLR  ==      000020          ; DZ11 CSR device clear bit set
1519            040040          MSETIE  ==      040040          ; DZ11 CSR Master Scan Enable and Transmitter
1520                                                            ; Interrupt Enable bits set
1521            000050          MSEMAI  ==      000050          ; DZ11 CSR Master Scan Enable and MAIntenance
1522                                                            ; loopback mode bits set
1523            010000          RCVRON  ==      010000          ; DZ11 LPR Receiver On bit set
1524
1525            174000          RBUFCTL ==      174000          ; DZ11 RBUF mask to get line number after BIC
1526
1527            000100          DLAYarg ==      100             ; argument providing a rough 0.1 second delay
1528                                                            ; when used with the DLAY macro on the 11/70
1529            177754          DLAY2s  ==      -20.            ; 2 seconds delay to wait for echo
1530            177622          DLAY11s ==      -110.           ; 11 seconds delay to wait for echo
1531
1532
1533                            ;;*************************************************************************
1534                            ;* PROGRAM EVENT FLAG DEFINITIONS
1535                            ;;*************************************************************************
1536
1537
1538
```

GLOBAL DATA SECTION

```
1541                                    .SBTTL  GLOBAL DATA SECTION
1542
1543                                ;////////////////////////////////////////////////////////////////////////////////
1544                                ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1545                                ;/      IN MORE THAN ONE TEST.
1546                                ;////////////////////////////////////////////////////////////////////////////////
1547
1553
1554
1555                                ;;******************************************************************************
1556                                ;* STORAGE FOR DEVICE REGISTERS
1557                                ;;******************************************************************************
1558 002156                             DESCRIPT        <CZDZGA0 GS3WD.DZ11 LGC DIAG>
       002156                       L$DESC::
       002156  103     132     104         .ASCIZ  /CZDZGA0 GS3WD.DZ11 LGC DIAG/
       002161  132     107     101
       002164  060     040     107
       002167  123     063     127
       002172  104     056     104
       002175  132     061     061
       002200  040     114     107
       002203  103     040     104
       002206  111     101     107
       002211  000
                                        .EVEN
1559
1560
1561
1574
1575                                ;       ERRTBL
1576
1577
1578
1579
1580                                ;******************************************************************************
1581                                ;* PROGRAM CONTROL FLAGS
1582                                ;******************************************************************************
1583
1584 002212  000000                  FTIME: .word   0                ; boolean to record first initialization
1585 002214  000000                  TMODE: .word   0                ; Test mode
1586
1587                                ;******************************************************************************
1588                                ;* PROGRAM CONTROL PARAMETERS
1589                                ;******************************************************************************
1590
1591 002216  000000                    UUT: .word   0                ; Unit under test
1592
1593 002220  000000                 SWPRTY: .word   0                ; Switch priority (line x, y or none)
1594
1595 002222  000005                 MAXERR: .word   5                ; max error count before dropping unit
1596 002224  000000                 ERRCNT: .word   0                ; error count
1597
1598 002226  000000                 SAVE4:  .word   0                ; temporary storage for timeout trap
1599 002230  000000                 SAVE6:  .word   0                ; vector
1600
```

GLOBAL DATA SECTION

```
                                ;*****************************************************************************
                                ;* MISCELLANEOUS STORAGE
                                ;*****************************************************************************

002232  000000                  TXPSW:  .word   0               ; transmitter interrupt vector PSW

002234  000000                   DZPTY: .word   0               ; DZ11 priority
002236  011070                   TLPR0: .word   011070          ; predefined parameter description for lines into the
                                                                ; GS03-WD :
                                                                ;       - 8 bit characters
                                                                ;       - 1 start bit, 2 stop bits
                                                                ;       - 110 bauds or 1 character every 100 ms
                                                                ;       - receiver on

002240  000000                   TLPRx: .word   0               ; test LPR for line x
002242  000000                   TLPRy: .word   0               ; test LPR for line y

002244  000000                  DLAYC1: .word   0
002246  000000                  DLAYC2: .word   0

002250  000000                    ECHO: .WORD   0               ; to store RBUF contents echoed back through
                                                                ; the DZ11
```

GLOBAL DATA SECTION

```
1628
1629                                    ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1630                                    ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1631
1632                                    ;*************************************************************************
1633                                    ;* POINTERS TO DZ11 VECTORS AND REGISTERS
1634                                    ;*************************************************************************
1635
1636 002252  000000                     DZrVCCa:.word   0       ; DZ11 receiver interrupt vector PC address
1637 002254  000000                     DZrVCSa:.word   0       ; DZ11 receiver interrupt vector PSW address
1638 002256  000000                     DZtVCCa:.word   0       ; DZ11 transmitter interrupt vector PC address
1639 002260  000000                     DZtVCSa:.word   0       ; DZ11 transmitter interrupt vector PSW address
1640 002262  000000                     DZCSRa: .word   0       ; DZ11 control status register address
1641 002264                             DZRBUFa:                ; DZ11 receive buffer/line parameter register
1642 002264  000000                     DZLPRa: .word   0       ; address
1643 002266  000000                     DZTCRa: .word   0       ; pointer to DZ11 transmit control register
1644 002270  000000                     DZTDRa: .word   0       ; pointer to DZ11 transmit data register
1645
1646                                    ;*************************************************************************
1647                                    ;* POINTERS TO GS03-WD LINES OUT OF THE DZ11
1648                                    ;*************************************************************************
1649
1650 002272  000000                      LNNBR: .word   0       ; Line
1651 002274  000000                     LNNBRx: .word   0       ; numbers
1652 002276  000000                     LNNBRy: .word   0       ; (0..7)
1653 002300  000000                      LNMAP: .word   0       ; Line
1654 002302  000000                     LNMAPx: .word   0       ; bitmaps
1655 002304  000000                     LNMAPy: .word   0       ; (0..377)
1656
1657
1658                                    ;;*****  STACK USED FOR SUBROUTINE LINKAGE  *****
1659
1660
1661                                    ;*************************************************************************
1662                                    ;* SUBROUTINE LINKAGE PARAMETERS
1663                                    ;*************************************************************************
1664
1665 002306  000000                      sbAOK: .WORD   0               ; Subroutine execution report
1666
1667 002310  000000                     LNTSTD: .WORD   0               ; Number of line echo is to be expected on
1668
1669 002312  000000                     OLDLNNB:.WORD   0               ; Number of the line the GS03-WD is switched to
1670                                                                    ; when calling subroutine sbSW31
1671
1672 002314  000000                     NEWLNMP:.WORD   0               ; Bitmap and
1673 002316  000000                     NEWLNNB:.WORD   0               ; Number of the line the GS03-WD is to be
1674                                                                    ; switched to if subroutine sbSW31 succeeds
1675
1676 002320  000000                     ADDR:   .word   0               ; Parameters
1677 002322  000000                     UNIT:   .word   0               ; for error reports
1678
1679
```

GLOBAL TEXT SECTION

```
                                        .SBTTL  GLOBAL TEXT SECTION

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

                                ;;**************************************************************************
                                ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;;**************************************************************************
                                        DEVTYP  <GS03WD MODULE>
002324                          L$DVTYP::
002324                                  .ASCIZ  /GS03WD MODULE/
002324  107     123     060
002327  063     127     104
002332  040     115     117
002335  104     125     114
002340  105     000

                                        .EVEN
```

```
1708                                    .SBTTL  GLOBAL SUBROUTINES
1709
1710                                ;-----------------------------------------------------------------------
1711                                ; MACRO'S NEEDED TO CALL SUBROUTINES
1712                                ;-----------------------------------------------------------------------
1713
1714
1715                                ; macro to wait a few ms
1716
1717                                ; Call sequence :       DLAY    D               0 < D < 177777
1718
1719
1720                                .macro  DLAY    D
1721                                        MOV     #176630, DLAYC1
1722                                        MOV     #D, DLAYC2
1723                                        JSR     PC, $DWTG2
1724                                .endm
1725
1726
```

GLOBAL SUBROUTINES

```
1729                                    ; Subroutine to wait for event or timeout
1730
1731                                    ; Calling sequences :   JSR   PC, sbWTG1
1732                                    ;                       JSR   PC, sbWTG2
1733
1734
1735                                    ; Inputs parameters :   DLAYC1, DLAYC2
1736
1737
1738                                    ; subroutine sbWTG1 :
1739                                    ;       for I := DLAYC1 to 0 do begin end
1740
1741 002342  013746  002244            sbWTG1: MOV     DLAYC1, - (SP)
1742
1743 002346  005237  002244            loopG1: INC     DLAYC1
1744 002352  001375                            BNE     loopG1
1745
1746 002354  012637  002244                    MOV     (SP) +, DLAYC1
1747 002360  000207                            RTS     PC
1748                                    ; end sbWTG1
1749
1750
1751                                    ; subroutine sbWTG2 :
1752                                    ;       for DLAYC2 := DLAYC2 downto 0
1753                                    ;       do begin for J := DLAYC1 to 0
1754                                    ;                  do
1755                                    ;                ;
1756                                    ;
1757
1758 002362  004737  002342            sbWTG2: JSR     PC, sbWTG1
1759
1760 002366                                    BREAK
     002366  104422                            TRAP    C$BRK
1761
1762 002370  005337  002246                    DEC     DLAYC2
1763 002374  001372                            BNE     sbWTG2
1764
1765 002376  000207                            RTS     PC
1766                                    ; end sbWTG2
1767
1768
1769
```

GLOBAL SUBROUTINES

```
1772                                    ; Routine to drop unit after 5 errors
1773                                    ;
1774                                    ; Call sequence : JSR   PC, CHKMAX
1775                                    ;
1776                                    ;
1777 002400                             CHKMAX: INLOOP                  ; If looping on error
     002400  104420                             TRAP    C$INLP
1778 002402                                     BCOMPLETE       1$      ; then exit
     002402  103432                             BCS     1$
1779                                    ;
1780 002404                                     RFLAGS  R0              ;
     002404  104421                             TRAP    C$RFLA
1781 002406  032700  000040                     BIT     #IDU, R0        ; If dropping of units is inhibitted
1782 002412  001026                             BNE     1$              ; then exit
1783                                    ;
1784 002414  005237  002224                     INC     ERRCNT          ; Update error count
1785 002420  023737  002224  002222             CMP     ERRCNT, MAXERR  ; If there are'nt too many errors
1786 002426  003420                             BLE     1$              ; then exit
1787                                    ;
1788 002430                                     PRINTF  #TMNYERS, MAXERR, UUT
     002430  013746  002216                     MOV     UUT,-(SP)
     002434  013746  002222                     MOV     MAXERR,-(SP)
     002440  012746  002472                     MOV     #TMNYERS,-(SP)
     002444  012746  000003                     MOV     #3,-(SP)
     002450  010600                             MOV     SP,R0
     002452  104417                             TRAP    C$PNTF
     002454  062706  000010                     ADD     #10,SP
1789                                                                    ; else print 'Maximum error count
1790                                                                    ; of <maxerr> exceeded for unit <UUT>'
1791 002460                                     DODU    UUT             ; and drop unit
     002460  013700  002216                     MOV     UUT,R0
     002464  104451                             TRAP    C$DODU
1792                                    ;
1793 002466                                     DOCLN                   ; Abort subpass
     002466  104444                             TRAP    C$DCLN
1794                                    ;
1795 002470  000207                     1$:     RTS     PC
1796                                    ;
1797                                    ;
1798                                    ;
1799                                    ;
1800                                            .nlist  BEX
1801 002472     045     116     045    TMNYERS:.ASCIZ  /%N%AMaximum error count of %D3%A exceeded for unit %D2/
1802                                            .list   BEX
1803                                            .EVEN
1804                                    ;
1805                                    ;
```

GLOBAL SUBROUTINES

```
1808                                    ; service routine to transmit in interrupt mode :
1809 002562                             $GNSRV  svTXG1
     002562                             svTXG1::
1810 002562  112777  000101  177500              MOVB    #'A, @DZTDR    ;
1811
1812 002570                             ENDSRV
     002570                             L10002:
     002570  000002                             RTI
1813
```

GLOBAL SUBROUTINES

```
1816                                    ; subroutine to initialize DZ11 for interrupt mode transmission
1817                                    ;
1818                                    ; Calling sequence :    JSR   PC, sbIDG1
1819                                    ;
1820 002572                            sbIDG1: SETVEC  DZTVCCa, #svTXG1, TXPSW
     002572  013746  002232                     MOV     TXPSW,-(SP)
     002576  012746  002562                     MOV     #svTXG1,-(SP)
     002602  013746  002256                     MOV     DZTVCCa,-(SP)
     002606  012746  000003                     MOV     #3,-(SP)
     002612  104437                            TRAP    C$SVEC
     002614  062706  000010                     ADD     #10,SP
                                                                    ;Set up transmitter interrupt vector
1821
1822
1823 002620  012777  000020  177434            MOV     #CSRCLR, @DZCSRa; Set CLR bit of DZ11 CSR
1824
1825 002626  032777  000020  177426  nCLDG1:   BIT     #CSRCLR, @DZCSRa; Test CLR
1826 002634                                    BREAK                   ; Authorize "control-C" abort
     002634  104422                            TRAP    C$BRK
1827 002636  001373                            BNE     nCLDG1          ; Wait until CSRCLR = 0
1828
1829 002640  000207                            RTS     PC
1830                                    ; end sbIDG1
1831
```

GLOBAL SUBROUTINES

```
1834                                    ; subroutine to transmit and check for echo back from the GS03-WD
1835                                    ;
1836                                    ; Calling sequence :    JSR   PC, sbTEG1
1837                                    ;
1838                                    ; Input parameter : LNTSTD contains the number of the line on which echo
1839                                    ;                    is to be tested for
1840                                    ;
1841                                    ; Implicit input : DZ11 LPR register has been loaded with corresponding
1842                                    ;                  parameters and DZ11 TCR with the bitmap of the line(s)
1843                                    ;                  to be activated (i. e. : including that which was
1844                                    ;                  "already" active)
1845                                    ;
1846                                    ; Output parameters : if successful, return with sbAOK = 1 else with sbAOK = 0
1847                                    ;
1848 002642  005037  002306             sbTEG1: CLR     sbAOK           ;
1849
1850 002646  012777  040040  177406             MOV     #MSETIE, @DZCSRa; Enable interrupt mode transmission
1851 002654  012701  177622                     MOV     #delay11s, R1   ; Set up 11 seconds delay
1852
1853 002660  105777  177376             nRDNG1: TSTB    @DZCSRa         ; If silo empty,
1854 002664  100014                             BPL     WAITG1          ; then wait
1855
1856 002666  017737  177372  002250     nETYG1: MOV     @DZRBUFa, ECHO  ; else empty it
1857
1858 002674  100010                             BPL     WAITG1          ; until it becomes empty
1859
1860 002676  042737  174000  002250             BIC     #RBUFCTL, ECHO  ; or an echo
1861 002704  123737  002251  002310             CMPB    ECHO + 1, LNTSTD; on line "LNTSTD"
1862 002712  001414                             BEQ     succG1          ; is detected
1863 002714  000764                             BR      nETYG1
1864
1865 002716                             WAITG1: DLAY    DLAYarg         ;
1866 002736  005201                             INC     R1              ;
1867 002740  001347                             BNE     nRDNG1
1868
1869 002742  000207                             RTS     PC              ; When delay is elapsed, return
1870                                                                     ; with sbAOK = 0
1871
1872 002744  012737  000001  002306     succG1: MOV     #1, sbAOK       ; echo on line i means GS03-WD is switched
1873 002752  000207                             RTS     PC              ; to line i : return with sbAOK = 1
1874
1875                                    ; end sbTEG1
1876
1877
```

GLOBAL SUBROUTINES

```
1880                            .macro  ED$CALL XY
1881                                    .LIST
1882                                    ;****************************** TEST'XY' ***************************
1883                                    .NLIST
1884                            .endm
1885
1886
1887
1888                            .macro  BADHEAD
1889                                    .RADIX  10
1890                                    ED$CALL \T$TESTNUM+1
1891                                    .RADIX  8
1892                            .endm
1893
1894
1895
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                        ;////////////////////////////////////////////////////////////////
                        ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
                        ;/      THAT ARE USED IN MORE THAN ONE TEST.
                        ;////////////////////////////////////////////////////////////////

                                .nlist  BEX
002754  105     103     110  FAISWF: .ASCIZ  /ECHO ON WRONG LINE/
002777  106     101     111  FAISWT: .ASCIZ  /FAIL TO SWITCH TO/
003021  105     103     110  WD2ECH: .ASCIZ  /ECHO ON BOTH LINES/
003044  120     122     111  PTYCFL: .ASCIZ  /PRIORITY CONFLICT/
003066  116     117     040  NO1LEC: .ASCIZ  /NO ECHO ON ONE LINE/
003112  116     117     040  NOWDEC: .ASCIZ  /NO WD ECHO/
003125  104     132     061  DZLBER: .ASCIZ  /DZ11 INTERNAL LOOPBACK ERROR/
003162  104     132     061  DZINER: .ASCIZ  /DZ11 INITIALIZATION FAULT/
003214  102     125     123  BUSTIM: .ASCIZ  /BUS TIMEOUT/
                                .EVEN

                        ;*****************************************************************
                        ;* BASIC ERROR REPORTS MESSAGES :
                        ;*****************************************************************

003230  045     116     045  FSWF:   .ASCIZ  /%N%AEcho from GS03-WD received on wrong line # %D1%A (expected : # %D1%A)./
003343  045     116     045  FSWT:   .ASCIZ  /%N%AGS03-WD failed to switch to line # %D1/
003416  045     116     045  WD2E:   .ASCIZ  /%N%AEcho from GS03-WD received on both lines # %D1%A and # %D1/
003515  045     116     045  PYCF:   .ASCIZ  /%N%ABoth lines have switch priority over each other./
003602  045     116     045  N1LE:   .ASCIZ  /%N%ANo echo received back from GS03-WD on line # %D1/
003667  045     116     045  NWDE:   .ASCIZ  /%N%ANo echo received back from GS03-WD on either line # %D1%A or # %D1/
003776  045     116     045  DZLB:   .ASCIZ  /%N%ADZ11 internal loop back malfunction on line # %D1/
004064  045     116     045  DZIN:   .ASCIZ  /%N%ADZ11 failed to reset./
004116  045     116     045  CSRw:   .ASCIZ  /%N%AUnsuccessful attempt to write to DZ11 CSR at address %O6/
004213  045     116     045  CSRr:   .ASCIZ  /%N%AUnsuccessful attempt to read DZ11 CSR at address %O6/
                                .EVEN

                        ;*****************************************************************
                        ;* EXTENDED ERROR REPORTS MESSAGES :
                        ;*****************************************************************

004304  045     116     045  NOEC1L: .ASCIZ  /%N%ANo echo received back from GS03-WD on line # %D1/
004371  045     116     045  STEC1L: .ASCIZ  /%N%AEcho is still being received on line # %D1%A when actually transmitting
o
004523  045     116     045  CKFMSW: .ASCIZ  /%N%ACheck FORCE, MANUAL switches, priority setting and cables./
004622  045     116     045  CKDPSW: .ASCIZ  /%N%ACheck cabling and dip switch E18 (must be OFF)./
004706  045     116     045  CKGSCF: .ASCIZ  /%N%ACheck GS03 configuration./
004744  045     116     045  DZDIAG: .ASCIZ  /%N%ARun DZ11 diagnostic./
004775  045     116     045  CKDZAD: .ASCIZ  /%N%ACheck DZ11 address./
                                .list   BEX
                                .EVEN
```

GLOBAL ERROR REPORT SECTION

```
                                ;-------------------------------------------------------------------
                                ; MACRO'S NEEDED TO REPORT ERRORS
                                ;-------------------------------------------------------------------



                                ; Error # 0 report
                                BGNMSG  pCSRr                   ; CSR read error
                                pCSRr::
                                PRINTB  #CSRr, ADDR
005026  013746  002320                  MOV     ADDR,-(SP)
005032  012746  004213                  MOV     #CSRr,-(SP)
005036  012746  000002                  MOV     #2,-(SP)
005042  010600                          MOV     SP,R0
005044  104414                          TRAP    C$PNTB
005046  062706  000006                  ADD     #6,SP
                                PRINTX  #CKDZAD
005052  012746  004775                  MOV     #CKDZAD,-(SP)
005056  012746  000001                  MOV     #1,-(SP)
005062  010600                          MOV     SP,R0
005064  104415                          TRAP    C$PNTX
005066  062706  000004                  ADD     #4,SP
005072  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
                                ENDMSG
                                L10003:
005076  104423                          TRAP    C$MSG

                                ; Error # 1 report
                                BGNMSG  pCSRw                   ; CSR write error
                                pCSRw::
                                PRINTB  #CSRw, ADDR
005100  013746  002320                  MOV     ADDR,-(SP)
005104  012746  004116                  MOV     #CSRw,-(SP)
005110  012746  000002                  MOV     #2,-(SP)
005114  010600                          MOV     SP,R0
005116  104414                          TRAP    C$PNTB
005120  062706  000006                  ADD     #6,SP
                                PRINTX  #CKDZAD
005124  012746  004775                  MOV     #CKDZAD,-(SP)
005130  012746  000001                  MOV     #1,-(SP)
005134  010600                          MOV     SP,R0
005136  104415                          TRAP    C$PNTX
005140  062706  000004                  ADD     #4,SP
005144  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
                                ENDMSG
                                L10004:
005150  104423                          TRAP    C$MSG

                                ; Error # 2 report
                                BGNMSG  pDZIN                   ; DZ11 initialization error
                                pDZIN::
                                PRINTB  #DZIN
005152  012746  004064                  MOV     #DZIN,-(SP)
005156  012746  000001                  MOV     #1,-(SP)
005162  010600                          MOV     SP,R0
005164  104414                          TRAP    C$PNTB
005166  062706  000004                  ADD     #4,SP
```

GLOBAL ERROR REPORT SECTION

```
1979 005172
     005172  012746  004775          PRINTX  #CKDZAD
     005172  012746  004775                  MOV     #CKDZAD,-(SP)
     005176  012746  000001                  MOV     #1,-(SP)
     005202  010600                          MOV     SP,R0
     005204  104415                          TRAP    C$PNTX
     005206  062706  000004                  ADD     #4,SP
1980 005212                          PRINTX  #DZDIAG
     005212  012746  004744                  MOV     #DZDIAG,-(SP)
     005216  012746  000001                  MOV     #1,-(SP)
     005222  010600                          MOV     SP,R0
     005224  104415                          TRAP    C$PNTX
     005226  062706  000004                  ADD     #4,SP
1981 005232  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
1982 005236                          ENDMSG
     005236                          L10005:
     005236  104423                          TRAP    C$MSG
1983
1984
1985 005240                          ; Error # 3 report
     005240                          BGNMSG  pDZLB                   ; DZ11 loopback error
                                     pDZLB::
1986 005240                          PRINTB  #DZLB, LNNBR
     005240  013746  002272                  MOV     LNNBR,-(SP)
     005244  012746  003776                  MOV     #DZLB,-(SP)
     005250  012746  000002                  MOV     #2,-(SP)
     005254  010600                          MOV     SP,R0
     005256  104414                          TRAP    C$PNTB
     005260  062706  000006                  ADD     #6,SP
1987 005264                          PRINTX  #CKDZAD
     005264  012746  004775                  MOV     #CKDZAD,-(SP)
     005270  012746  000001                  MOV     #1,-(SP)
     005274  010600                          MOV     SP,R0
     005276  104415                          TRAP    C$PNTX
     005300  062706  000004                  ADD     #4,SP
1988 005304                          PRINTX  #DZDIAG
     005304  012746  004744                  MOV     #DZDIAG,-(SP)
     005310  012746  000001                  MOV     #1,-(SP)
     005314  010600                          MOV     SP,R0
     005316  104415                          TRAP    C$PNTX
     005320  062706  000004                  ADD     #4,SP
1989 005324  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
1990 005330                          ENDMSG
     005330                          L10006:
     005330  104423                          TRAP    C$MSG
1991
1992
1993 005332                          ; Error # 4 report
     005332                          BGNMSG  pNXDE                   ; WD fail to echo error
                                     pNXDE::
1994 005332                          PRINTB  #NXDE, LNNBRx, LNNBRy
     005332  013746  002276                  MOV     LNNBRy,-(SP)
     005336  013746  002274                  MOV     LNNBRx,-(SP)
     005342  012746  003667                  MOV     #NXDE,-(SP)
     005346  012746  000003                  MOV     #3,-(SP)
     005352  010600                          MOV     SP,R0
     005354  104414                          TRAP    C$PNTB
     005356  062706  000010                  ADD     #10,SP
1995 005362                          PRINTX  #CKDPSW
     005362  012746  004622                  MOV     #CKDPSW,-(SP)
```

GLOBAL ERROR REPORT SECTION

```
                005366  012746  000001                  MOV     #1,-(SP)
                005372  010600                          MOV     SP,R0
                005374  104415                          TRAP    C$PNTX
                005376  062706  000004                  ADD     #4,SP
1996            005402  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
1997            005406                          ENDMSG
                005406                          L10007:
                005406  104423                          TRAP    C$MSG
1998
1999                                            ; Error # 5 report
2000            005410                          BGNMSG  pN1LE                   ; No echo received on line x error
                005410                          pN1LE::
2001            005410                          PRINTB  #N1LE, LN#BRx
                005410  013746  002274                  MOV     LN#BRx,-(SP)
                005414  012746  003602                  MOV     #N1LE,-(SP)
                005420  012746  000002                  MOV     #2,-(SP)
                005424  010600                          MOV     SP,R0
                005426  104414                          TRAP    C$PNTB
                005430  062706  000006                  ADD     #6,SP
2002            005434                          PRINTX  #CKFMSW
                005434  012746  004523                  MOV     #CKFMSW,-(SP)
                005440  012746  000001                  MOV     #1,-(SP)
                005444  010600                          MOV     SP,R0
                005446  104415                          TRAP    C$PNTX
                005450  062706  000004                  ADD     #4,SP
2003            005454  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
2004            005460                          ENDMSG
                005460                          L10010:
                005460  104423                          TRAP    C$MSG
2005
2006                                            ; Error # 6 report
2007            005462                          BGNMSG  pPYCF                   ; Both lines have priority error
                005462                          pPYCF::
2008            005462                          PRINTB  #PYCF
                005462  012746  003515                  MOV     #PYCF,-(SP)
                005466  012746  000001                  MOV     #1,-(SP)
                005472  010600                          MOV     SP,R0
                005474  104414                          TRAP    C$PNTB
                005476  062706  000004                  ADD     #4,SP
2009            005502                          PRINTX  #CKGSCF
                005502  012746  004706                  MOV     #CKGSCF,-(SP)
                005506  012746  000001                  MOV     #1,-(SP)
                005512  010600                          MOV     SP,R0
                005514  104415                          TRAP    C$PNTX
                005516  062706  000004                  ADD     #4,SP
2010            005522  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
2011            005526                          ENDMSG
                005526                          L10011:
                005526  104423                          TRAP    C$MSG
2012
2013                                            ; Error # 7 report
2014            005530                          BGNMSG  pWD2E                   ; Echo on both lines error
                005530                          pWD2E::
2015            005530                          PRINTB  #WD2E, LN#BRx, LN#BRy
                005530  013746  002276                  MOV     LN#BRy,-(SP)
                005534  013746  002274                  MOV     LN#BRx,-(SP)
                005540  012746  003416                  MOV     #WD2E,-(SP)
```

GLOBAL ERROR REPORT SECTION

```
                005544  012746  000003                  MOV     #3,-(SP)
                005550  010600                          MOV     SP,R0
                005552  104414                          TRAP    C$PNTB
                005554  062706  000010                  ADD     #10,SP
     2016       005560  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
     2017       005564                          ENDMSG
                005564                          L10012:
                005564  104423                          TRAP    C$MSG
     2018
     2019                                       ; Error # 8 report
     2020       005566                          BGNMSG  pFSWT                   ; WD fail to switch to line i error
                005566                          pFSWT::
     2021       005566                          PRINTB  #FSWT, NEWLNNB
                005566  013746  002316                  MOV     NEWLNNB,-(SP)
                005572  012746  003343                  MOV     #FSWT,-(SP)
                005576  012746  000002                  MOV     #2,-(SP)
                005602  010600                          MOV     SP,R0
                005604  104414                          TRAP    C$PNTB
                005606  062706  000006                  ADD     #6,SP
     2022       005612                          PRINTX  #NDEC1L, NEWLNNB
                005612  013746  002316                  MOV     NEWLNNB,-(SP)
                005616  012746  004304                  MOV     #NDEC1L,-(SP)
                005622  012746  000002                  MOV     #2,-(SP)
                005626  010600                          MOV     SP,R0
                005630  104415                          TRAP    C$PNTX
                005632  062706  000006                  ADD     #6,SP
     2023       005636                          PRINTX  #CKFMSW
                005636  012746  004523                  MOV     #CKFMSW,-(SP)
                005642  012746  000001                  MOV     #1,-(SP)
                005646  010600                          MOV     SP,R0
                005650  104415                          TRAP    C$PNTX
                005652  062706  000004                  ADD     #4,SP
     2024       005656  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
     2025       005662                          ENDMSG
                005662                          L10013:
                005662  104423                          TRAP    C$MSG
     2026
     2027                                       ; Error # 9 report
     2028       005664                          BGNMSG  pFSWF                   ; WD echo on wrong line error
                005664                          pFSWF::
     2029       005664                          PRINTB  #FSWF, OLDLNNB, NEWLNNB
                005664  013746  002316                  MOV     NEWLNNB,-(SP)
                005670  013746  002312                  MOV     OLDLNNB,-(SP)
                005674  012746  003230                  MOV     #FSWF,-(SP)
                005700  012746  000003                  MOV     #3,-(SP)
                005704  010600                          MOV     SP,R0
                005706  104414                          TRAP    C$PNTB
                005710  062706  000010                  ADD     #10,SP
     2030       005714                          PRINTX  #STEC1L, OLDLNNB, NEWLNNB
                005714  013746  002316                  MOV     NEWLNNB,-(SP)
                005720  013746  002312                  MOV     OLDLNNB,-(SP)
                005724  012746  004371                  MOV     #STEC1L,-(SP)
                005730  012746  000003                  MOV     #3,-(SP)
                005734  010600                          MOV     SP,R0
                005736  104415                          TRAP    C$PNTX
                005740  062706  000010                  ADD     #10,SP
     2031       005744                          PRINTX  #CKGSCF
```

F5

GLOBAL ERROR REPORT SECTION

```
        005744  012746  004706                  MOV     @CKGSCF,-(SP)
        005750  012746  000001                  MOV     #1,-(SP)
        005754  010600                           MOV     SP,R0
        005756  104415                           TRAP    C$PNTX
        005760  062706  000004                  ADD     #4,SP
  2032  005764  004737  002400                  JSR     PC, CHKMAX      ; check if too many errors
  2033  005770                        ENDMSG
        005770                        L10014:
        005770  104423                           TRAP    C$MSG
  2034
  2035
  2036
  2037
  2038
```

F5

REPORT CODING SECTION

```
                                .SBTTL  REPORT CODING SECTION


                               ;**
                               ; THE REPORT CODING SECTION CONTAINS THE
                               ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                               ;--

005772                                 BGNRPT
005772                         L$RPT::


005772                                 ENDRPT
005772                         L10015:
005772  104425                         TRAP    C$RPT
```

INITIALIZE SECTION

```
                                .SBTTL INITIALIZE SECTION

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

                                BGNINIT
                        L$INIT::
```

INITIALIZE SECTION

```
2105                                    ; Context initialization
2106 005774  005737  002212                    TST     FTIME           ; If this is the first pass through this
2107 006000  001011                            BNE     nFTMI1          ; routine,
2108 006002  013737  000004  002226            MOV     @04, SAVE4      ; then the "trap through 4" vector is saved
2109 006010  013737  000006  002230            MOV     @06, SAVE6      ;
2110 006016  012737  000001  002212            MOV     #1, FTIME       ;
2111 006024  013737  002226  000004  nFTMI1:   MOV     SAVE4, @04      ; else it is restored
2112 006032  013737  002230  000006            MOV     SAVE6, @06      ;
2113
2114 006040                                    READEF  #EF.START       ; "START",
     006040  012700  000040                    MOV     #EF.START,R0
     006044  104447                            TRAP    C$REFG
2115 006046                                    BCOMPLETE       ftUNI1  ;
     006046  103422                            BCS     ftUNI1
2116
2117 006050                                    READEF  #EF.RESTART     ; "RESTART" commands,
     006050  012700  000037                    MOV     #EF.RESTART,R0
     006054  104447                            TRAP    C$REFG
2118 006056                                    BCOMPLETE       ftUNI1  ;
     006056  103416                            BCS     ftUNI1
2119
2120 006060                                    READEF  #EF.PWR         ;  or POWER UP :
     006060  012700  000034                    MOV     #EF.PWR,R0
     006064  104447                            TRAP    C$REFG
2121 006066                                    BCOMPLETE       ftUNI1  ;  start with first unit (# 0)
     006066  103412                            BCS     ftUNI1
2122
2123 006070                                    READEF  #EF.CONTINUE    ; If this a "continue" command,
     006070  012700  000036                    MOV     #EF.CONTINUE,R0
     006074  104447                            TRAP    C$REFG
2124 006076                                    BNCOMPLETE      contI1  ; then exit
     006076  103002                            BCC     contI1
2125 006100                                    EXIT    INIT            ; (no re-initialization)
     006100  104432                            TRAP    C$EXIT
     006102  001440                            .WORD   L10016-.
2126
2127 006104                            contI1: READEF  #EF.NEW         ; If this is not a new pass,
     006104  012700  000035                    MOV     #EF.NEW,R0
     006110  104447                            TRAP    C$REFG
2128 006112                                    BNCOMPLETE      nxUNI1  ; then get next unit
     006112  103003                            BCC     nxUNI1
2129
2130 006114  012737  177777  002216    ftUNI1: MOV     #-1, UUT        ;
2131
2132 006122  005237  002216            nxUNI1: INC     UUT             ; Point to next unit
2133 006126  023737  002216  002012            CMP     UUT, L$UNIT     ; If there is'nt any,
2134 006134  002161                            BGE     aborI1          ; then end-of-pass
2135
```

INITIALIZE SECTION

```
2138                                    ; Load hardware parameters for unit under test :
2139 006136                                     GPHARD  UUT, R1         ; Call to DRS to put p-table address in R1
     006136  013700  002216                     MOV     UUT,R0
     006142  104442                             TRAP    C$GPHRD
     006144  010001                             MOV     R0,R1
2140 006146                                     BCOMPLETE gtPMI1        ;
     006146  103413                             BCS     gtPMI1
2141
2142 006150                                     PRINTF  #NOTAV, UUT     ; If not available,
     006150  013746  002216                     MOV     UUT,-(SP)
     006154  012746  007316                     MOV     #NOTAV,-(SP)
     006160  012746  000002                     MOV     #2,-(SP)
     006164  010600                             MOV     SP,R0
     006166  104417                             TRAP    C$PNTF
     006170  062706  000006                     ADD     #6,SP
2143 006174  000752                             BR      nxUNI1          ; then get next unit
2144
2145 006176  011137  002262            gtPMI1: MOV     (R1), DZCSRa    ; Get address of DZ11 CSR
2146
2147 006202  011137  002264                     MOV     (R1), DZRBUFa   ; Get address of DZ11 RBUF/LPR
2148 006206  062737  000002  002264             ADD     #2, DZRBUFa     ; (DZRBUFa = DZLPRa)
2149
2150 006214  011137  002266                     MOV     (R1), DZTCRa    ; Get address of DZ11 TCR
2151 006220  062737  000004  002266             ADD     #4, DZTCRa
2152
2153 006226  012137  002270                     MOV     (R1)+, DZTDRa   ; Get address of DZ11 TDR
2154 006232  062737  000006  002270             ADD     #6, DZTDRa
2155
2156 006240  011137  002252                     MOV     (R1), DZrVCCa   ; Get address of DZ11 receiver interrupt
2157 006244  011137  002254                     MOV     (R1), DZrVCSa   ; vector
2158 006250  062737  000002  002254             ADD     #2, DZrVCSa
2159
2160 006256  011137  002256                     MOV     (R1), DZtVCCa   ; Get address of DZ11 transmitter interrupt
2161 006262  062737  000004  002256             ADD     #4, DZtVCCa     ; vector
2162 006270  012137  002260                     MOV     (R1)+, DZtVCSa
2163 006274  062737  000006  002260             ADD     #6, DZtVCSa
2164
2165 006302  012137  002234                     MOV     (R1)+, DZPTY    ; Get pointer to tx priority level
2166
2167 006306  012137  002300                     MOV     (R1)+, LNMAP    ; Get bitmap of active lines
2168
2169 006312  011137  002214                     MOV     (R1), TMODE     ; Get test mode
2170 006316  001025                             BNE     MOD1I1
2171
2172 006320                                     PRINTF  #RUNG0a, UUT    ; 'Running on unit <UUT> in mode 0...'
     006320  013746  002216                     MOV     UUT,-(SP)
     006324  012746  006740                     MOV     #RUNG0a,-(SP)
     006330  012746  000002                     MOV     #2,-(SP)
     006334  010600                             MOV     SP,R0
     006336  104417                             TRAP    C$PNTF
     006340  062706  000006                     ADD     #6,SP
2173 006344                                     PRINTF  #RUNG0b, UUT    ; 'Only tests 1, 2 and 3 are active...'
     006344  013746  002216                     MOV     UUT,-(SP)
     006350  012746  007056                     MOV     #RUNG0b,-(SP)
     006354  012746  000002                     MOV     #2,-(SP)
     006360  010600                             MOV     SP,R0
     006362  104417                             TRAP    C$PNTF
```

INITIALIZE SECTION

```
                006364  062706  000006                  ADD     #6,SP
          2174  006370  000424                          BR      contI2
          2175
          2176  006372                          MOD1I1: PRINTF  #RUNG1a, UUT    ; 'Running on unit <UUT> in mode 1...'
                006372  013746  002216                  MOV     UUT,-(SP)
                006376  012746  007141                  MOV     #RUNG1a,-(SP)
                006402  012746  000002                  MOV     #2,-(SP)
                006406  010600                          MOV     SP,R0
                006410  104417                          TRAP    C$PNTF
                006412  062706  000006                  ADD     #6,SP
          2177  006416                                  PRINTF  #RUNG1b, UUT    ; 'Only tests 1 and 4 are active...'
                006416  013746  002216                  MOV     UUT,-(SP)
                006422  012746  007236                  MOV     #RUNG1b,-(SP)
                006426  012746  000002                  MOV     #2,-(SP)
                006432  010600                          MOV     SP,R0
                006434  104417                          TRAP    C$PNTF
                006436  062706  000006                  ADD     #6,SP
          2178
```

INITIALIZE SECTION

```
2181
2182 006442  013701  002234          ; Compute program variables accordingly :
2183 006446  072127  000005  contI2: MOV     DZPTY, R1          ; Load DZ11
2184 006452  010137  002232          ASH     #5, R1             ; bus priority
2185                                 MOV     R1, TXPSW          ; into TXPSW
2186 006456  005037  002224
2187 006462  004737  006506          CLR     ERRCNT             ;
2188                                 JSR     PC, sbLNI1         ; Compute "LNNBRi" and "LNMAPi" from "LNMAP"
2189 006466  005737  002306
2190 006472  001402                  TST     sbAOK              ; If wrong "LNMAP" format,
2191                                 BEQ     sborI1             ; then abort pass
2192 006474
     006474  104432                  EXIT    INIT
     006476  001044                  TRAP    C$EXIT
                                     .WORD   L10016-.
2193
2194 006500                  sborI1: DOCLN                      ;CLEAN UP AND ABORT PASS
     006500  104444                  TRAP    C$DCLN
2195 006502                          EXIT    INIT
     006502  104432                  TRAP    C$EXIT
     006504  001036                  .WORD   L10016-.
2196
2197
```

INITIALIZE SECTION

```
                                        ; subroutine to compute line map, number and parameters for lines x and y out
                                        ; of the DZ11
                                        ;
                                        ; Input parameter :
                                        ;
                                        ;       LNMAP
                                        ;
                                        ; Output parameters :
                                        ;               - cbAOK = 1 <=> successful ;
                                        ;               - if successful, line numbers (0..7) in LNNBRx, LNNBRy
                                        ;                                 line bitmaps in LNMAP, LNMAPx, LNMAPy
                                        ;                                 line parameters TLPRx, TLPRy
                                        ; Side effects :
                                        ;               - LNMAP is not modified ;
                                        ;               - LNNBR is left undefined.
                                        ;
006506  005037  002306          cbLNI1: CLR     cbAOK           ;
006512  005037  002302                  CLR     LNMAPx
006516  005037  002304                  CLR     LNMAPy

006522  113702  002300                  MOVB    LNMAP, R2
006526  112701  000001                  MOVB    #001, R1
006532  005037  002272                  CLR     LNNBR

006536  130102                  nxBII1: BITB    R1, R2
006540  001005                          BNE     RLMPI1          ; If found, then store value in LNMAPx-y

006542  005237  002272                  INC     LNNBR           ; else increment line number and
006546  106301                          ASLB    R1              ; shift set bit in R1 left one position
006550  103372                          BCC     nxBII1          ; as long as no overflow occurs
006552  000424                          BR      erLMI1

006554  040102                  RLMPI1: BIC     R1, R2          ; Clear bit in R2 that has just been found set

006556  105737  002302                  TSTB    LNMAPx          ; If LNMAPx has already been assigned a value,
006562  001045                          BNE     RLMPI2          ; then assign one to LNMAPy now

006564  110137  002302                  MOVB    R1, LNMAPx      ; Store
006570  013737  002272  002274          MOV     LNNBR, LNNBRx   ; results
006576  013737  002236  002240          MOV     TLPR0, TLPRx    ; into line x
006604  063737  002274  002240          ADD     LNNBRx, TLPRx   ; parameters

006612  005737  002214                  TST     TMODE           ; If mode 0 and LNMAP format was given right,
                                                                ; then now R2 = LNMAPy.
006616  001747                          BEQ     nxBII1          ; <- This is just to check for right format.

                                                                ; If mode 1,
006620  005702                          TST     R2              ; then only one line should be specified
006622  001442                          BEQ     succI1

006624                          erLMI1: PRINTF  #MGLMP1, LNMAP, TMODE
006624  013746  002214                  MOV     TMODE,-(SP)
006630  013746  002300                  MOV     LNMAP,-(SP)
006634  012746  007357                  MOV     #MGLMP1,-(SP)
006640  012746  000003                  MOV     #3,-(SP)
006644  010600                          MOV     SP,R0
006646  104417                          TRAP    C$PNTF
```

INITIALIZE SECTION

```
        006650  062706  000010                  ADD     #10.SP
                                                                    ; 'Wrong number of DZ11 lines...'
2251
2252    006654                                  PRINTF  #WGLMP2
        006654  012746  007467                  MOV     #WGLMP2,-(SP)
        006660  012746  000001                  MOV     #1,-(SP)
        006664  010600                          MOV     SP,R0
        006666  104417                          TRAP    C$PNTF
        006670  062706  000004                  ADD     #4,SP
2253    006674  000207                          RTS     PC
2254
2255    006676  105702                  RLMPI2: TSTB    R2              ; Check that no more than 2 lines were
2256    006700  001351                          BNE     erLMI1          ; specified
2257
2258    006702  110137  002304                  MOVB    R1, LNMAPy      ; Store
2259    006706  013737  002272  002276          MOV     LNNBR, LNNBRy   ; results
2260    006714  013737  002236  002242          MOV     TLPR0, TLPRy    ; into line y
2261    006722  063737  002276  002242          ADD     LNNBRy, TLPRy   ; parameters
2262
2263    006730  012737  000001  002306  succI1: MOV     #1, sbAOK       ;
2264    006736  000207                          RTS     PC
2265                                    ; end sbLNI1
2266
```

INITIALIZE SECTION

```
                                                .nlist  BEX
006740      045     116     045  RUNG0a: .ASCIZ  /%%AARunning on unit %D2%A in mode 0 : pass-time is 2 minutes on the PDP11-7
0.
007056      045     116     045  RUNG0b: .ASCIZ  /%%AAOnly tests 1, 2 and 3 are active in this mode./
007141      045     116     045  RUNG1a: .ASCIZ  /%%AARunning on unit %D2%A in mode 1 : type "ctrl C" to stop./
007236      045     116     045  RUNG1b: .ASCIZ  /%%AAOnly tests 1 and 4 are active in this mode./
007316      045     116     045  NOTAV:  .ASCIZ  /%%AAUnit %D2%A is not available./
007357      045     116     045  WGLMP1: .ASCIZ  /%%%%AAWrong number of DZ11 lines in bitmap "%O3%A" for mode %D1%A test./
007467      045     116     045  WGLMP2: .ASCIZ  /%%AAChange Hardware P-table to correct.%N/
                                    .list   BEX

                                    .EVEN

007542                          ENDINIT
007542                          L10016:
007542  104411                          TRAP    C$INIT
```

AUTODROP SECTION

```
                                .SBTTL  AUTODROP SECTION

                        ;++
                        ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
                        ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
                        ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
                        ; DROPPED FROM TESTING.
                        ;--
                                .EVEN
007544                          BGNAUTO
007544                  L$AUTO::

                        ; Check if DZ11 responds

007544  013701  002262          MOV     DZCSRa, R1      ;
007550  012705  000004          MOV     #4, R5          ; 4 DZ11  registers to be tested

                        ; Set up timeout trap :
007554  012737  007606  000004  MOV     #2$, @#4        ; address for timeout error trap handler
007562  012737  000340  000006  MOV     #MAXPRI, @#6    ; priority level 7 in trap PSW to lock out
                                                        ; other interrupts

007570  005711          1$:     TST     (R1)            ;
007572  000240                  NOP
007574  062701  000002          ADD     #2, R1          ; next register
007600  005305                  DEC     R5              ; Decrement register count
007602  001372                  BNE     1$              ; and branch back if not last register
007604  000405                  BR      3$

                        ; time out error trap handler :
007606  062706  000004  2$:     ADD     #4,SP           ; Pop old PC, PSW
007612                          DODU    UUT             ; Drop unit under test
007612  013700  002216          MOV     UUT,R0
007616  104451                  TRAP    C$DODU

007620  013737  002226  000004  3$:     MOV     SAVE4, @#4      ; Restore original timeout vector
007626  013737  002230  000006          MOV     SAVE6, @#6      ;

007634                          ENDAUTO
007634                  L10017:
007634  104461                  TRAP    C$AUTO
```

CLEANUP CODING SECTION

```
                                .SBTTL  CLEANUP CODING SECTION

                        ;//////////////////////////////////////////////////////////////////////
                        ;// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                        ;// AT THE END OF EACH PASS.
                        ;//////////////////////////////////////////////////////////////////////

                                BGNCLN
                        L$CLEAN::

                                BRESET          ; bus reset
                                TRAP    C$RESET
                                CLRVEC  DZTVCCa ; Clear transmit interrupt vector
                                MOV     DZTVCCa,R0
                                TRAP    C$CVEC

                                ENDCLN
                        L10020:
                                TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

007650                          BGNDU
007650                  L$DU::
                                .EVEN
007650                          PRINTF  #DROPD, R0      ; DRS has put # of unit to be dropped in R0
007650  010046                  MOV     R0,-(SP)
007652  012746  007676          MOV     #DROPD,-(SP)
007656  012746  000002          MOV     #2,-(SP)
007662  010600                  MOV     SP,R0
007664  104417                  TRAP    C$PNTF
007666  062706  000006          ADD     #6,SP
007672                          EXIT    DU
007672  000167                  .WORD   J$JMP
007674  000030                  .WORD   L10021-2-.
                                .nlist  BEX
007676      045     116     045 DROPD:  .ASCIZ  /%N%AUnit %D2%A dropped./
                                .list   BEX
                                .EVEN
007726                          ENDDU
007726                  L10021:
007726  104453                  TRAP    C$DU
```

ADD UNIT SECTION

```
                                .SBTTL  ADD UNIT SECTION

                        ;////////////////////////////////////////////////////////////////////
                        ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                        ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
                        ;/  "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
                        ;////////////////////////////////////////////////////////////////////


                                BGNAU
                        L$AU::
                                .EVEN

                                PRINTF  @ADDED, R0      ; DRS has put # of unit to be added in R0
007730  010046                  MOV     R0,-(SP)
007732  012746  007756          MOV     @ADDED,-(SP)
007736  012746  000002          MOV     #2,-(SP)
007742  010600                  MOV     SP,R0
007744  104417                  TRAP    C$PNTF
007746  062706  000006          ADD     #6,SP

007752                          EXIT    DU
007752  000167                  .WORD   J$JMP
007754  177750                  .WORD   L10021-2-.

                                .nlist  BEX
007756      045     116     045 ADDED:  .ASCIZ  /%N%AUnit %D2%A added./
                                .list   BEX
                                .EVEN

                                ENDAU
010004
                        L10022:
010004  104452                 TRAP    C$AU
```

HARDWARE TESTS

```
                                .SBTTL  HARDWARE TESTS


                                ;START OF CODE BLOCK WHICH IS USED AS DATA
010006                          ROMMAP::++
                                ; TEST TO ...
                                ;--


                                ;       BGNTST


                                ;       EXIT    TST


                                        .EVEN
                                ;       ENDTST
```

HARDWARE TESTS

```
2539 010006                                     BADHEAD
                                                ;*************************** TEST1 ***************************
2540                                            ;*
2541                                            ;*      Purpose : basic test of DZ11.
2542                                            ;*
2543                                            ;*      Description :
2544                                            ;*      - Subtest 1 : Check that DZ11 CSR can be  written
2545                                            ;*      to and read from ;
2546                                            ;*      - Subtest 2 : Transmit a character in maintenance
2547                                            ;*      (internal) loopback  mode on the selected line(s)
2548                                            ;*      and check for proper echo.
2549                                            ;*
2550                                            ;*      Error messages :
2551                                            ;* #0,1 - Subtest 1 : "Unsuccessful attempt to
2552                                            ;*       write to/read DZ11 CSR"
2553                                            ;*                          "Check DZ11 address."
2554                                            ;*
2555                                            ;* #2   - Subtest 2 : "DZ11 failed to reset."
2556                                            ;*                          "Check DZ11 address."
2557                                            ;*                          "Run DZ11 diagnostic."
2558                                            ;*
2559                                            ;* #3   - Subtest 2 : "DZ11 internal loopback malfunction
2560                                            ;*      on line # <line number>"
2561                                            ;*                          "Check DZ11 address."
2562                                            ;*                          "Run DZ11 diagnostic."
2563                                            ;*
2564 010006                                     BADHEAD
                                                ;*************************** TEST1 ***************************
2565
2566 010006                             DGNTST
        010006                          T1::
2567 010006                             BGNSUB  ; Start of subtest 1
        010006                          T1.1:
        010006  104402                          TRAP    C$BSUB
2568 010010                             BGNSEG
        010010  104404                          TRAP    C$BSEG
2569                                            ; Set up timeout trap :
2570 010012  012737  010040  000004             MOV     #1$, @#4        ; address for CSR write error trap handler
2571 010020  012737  000340  000006             MOV     #340, @#6       ; priority level 7 in trap PSW to lock out
2572                                                                     ; other interrupts
2573
2574 010026  012777  000020  172226             MOV     #CSRCLR, @DZCSRa; Set CLR bit of DZ11 CSR
2575 010034  000240                             NOP
2576 010036  000422                             BR      contD1          ;
2577
2578                                            ; CSR write error trap handler : DEVICE FATAL ERROR
2579 010040  062706  000004                 1$: ADD     #4, SP          ; Pop old PC, PSW
2580 010044  013737  002262  002320             MOV     DZCSRa, ADDR    ; Report address location
2581 010052                                     ERRDF   0, BUSTIM, pCSRw;
        010052  104455                          TRAP    C$ERDF
        010054  000000                          .WORD   0
        010056  003214                          .WORD   BUSTIM
        010060  005100                          .WORD   pCSRw
2582 010062  013737  002226  000004             MOV     SAVE4, @#4
2583 010070  013737  002230  000006             MOV     SAVE6, @#6
2584 010076                                     DOCLN                   ; Abort pass
        010076  104444                          TRAP    C$DCLN
```

```
HARDWARE TESTS

 2585 010100                                ENDSEG
      010100                                10000$:
      010100  104405                              TRAP    C$ESEG
 2586
 2587 010102                                BGNSEG
      010102  104404                              TRAP    C$BSEG
 2588 010104  012737  010124  000004       contD1: MOV     #1$, @04        ; address for CSR read error trap handler
 2589
 2590 010112  032777  000020  172142               BIT     #CSRCLR, @DZCSRa; Test CLR
 2591 010120  000240                               NOP
 2592 010122  000423                               BR      contD2
 2593
 2594                                       ; CSR read error handler : DEVICE FATAL ERROR
 2595 010124  062706  000004                 1$:   ADD     #4, SP          ; Pop old PC, PSW
 2596 010130  013737  002262  002320               MOV     DZCSRa, ADDR    ; Report address location
 2597 010136  013737  002216  002322               MOV     UUT, UNIT       ; Report unit number
 2598 010144                                       ERRDF   1, BUSTIM, pCSRr;
      010144  104455                               TRAP    C$ERDF
      010146  000001                               .WORD   1
      010150  003214                               .WORD   BUSTIM
      010152  005026                               .WORD   pCSRr
 2599 010154  013737  002226  000004               MOV     SAVE4, @04
 2600 010162  013737  002230  000006               MOV     SAVE6, @06
 2601 010170                                       DOCLN                   ; Abort pass
      010170  104444                               TRAP    C$DCLN
 2602
 2603 010172  013737  002226  000004       contD2: MOV     SAVE4, @04
 2604 010200  013737  002230  000006               MOV     SAVE6, @06
 2605 010206                                ENDSEG
      010206                                10001$:
      010206  104405                              TRAP    C$ESEG
 2606 010210                                ENDSUB
      010210                                L10024:
      010210  104403                              TRAP    C$ESUB
 2607
 2608 010212                                BGNSUB  ; Start of subtest 2
      010212                                T1.2:
      010212  104402                              TRAP    C$BSUB
 2609                                      ; Initlialize DZ11 :
 2610 010214  012777  000020  172040               MOV     #CSRCLR, @DZCSRa; Set CLR bit of DZ11 CSR
 2611 010222  012701  177754                       MOV     #DLAY2s, R1     ; Set up 2 seconds delay
 2612
 2613 010226  032777  000020  172026       nCLDD1: BIT     #CSRCLR, @DZCSRa; Wait
 2614 010234  001417                               BEQ     contD3          ; for
 2615
 2616 010236                                       DLAY    DLAYarg         ; CSR CLear bit
 2617 010256  005201                               INC     R1              ; to clear (reset complete)
 2618 010260  001362                               BNE     nCLDD1          ; If time-out
 2619 010262                                       ERRDF   2, DZINER, pDZIN; then there's a problem
      010262  104455                               TRAP    C$ERDF
      010264  000002                               .WORD   2
      010266  003162                               .WORD   DZINER
      010270  005152                               .WORD   pDZIN
 2620 010272                                       DOCLN                   ; Abort pass
      010272  104444                               TRAP    C$DCLN
 2621
 2622                                      ; test transmission on line x :
```

J6

HARDWARE TESTS

```
2623
2624 010274  013777  002240  171762  contD3: MOV     TLPRx, @DZLPRa  ; Load parameters for line x
2625 010302  113777  002302  171756          MOVB    LNMAPx, @DZTCRa ; Enable transmission on line x
2626
2627 010310  004737  010426                  JSR     PC, @bTED1      ; Transmit and test echo on line x
2628
2629 010314  005737  002306                  TST     @bAOK           ; If normal,
2630 010320  001010                          BNE     contD4          ; then go on testing line y
2631
2632 010322  013737  002274  002272          MOV     LNNBRx, LNNBR   ; else report DEVICE FATAL error
2633 010330                                  ERRDF   3, DZLBER, pDZLB;
     010330  104455                          TRAP    C$ERDF
     010332  000003                          .WORD   3
     010334  003125                          .WORD   DZLBER
     010336  005240                          .WORD   pDZLB
2634 010340                                  DOCLN                   ; Abort pass
     010340  104444                          TRAP    C$DCLN
2635
2636                                 ; If mode 0, then test transmission on line y, too :
2637
2638 010342  005737  002214          contD4: TST     TMODE
2639 010346  001402                          BEQ     contD5
2640 010350                                  EXIT    TST
     010350  104432                          TRAP    C$EXIT
     010352  000242                          .WORD   L10023-.
2641
2642 010354  013777  002242  171702  contD5: MOV     TLPRy, @DZLPRa
2643 010362  113777  002304  171676          MOVB    LNMAPy, @DZTCRa
2644
2645 010370  004737  010426                  JSR     PC, @bTED1      ; Transmit and test echo on line y
2646
2647 010374  005737  002306                  TST     @bAOK           ; If normal,
2648 010400  001402                          BEQ     contD6          ; then
2649 010402                                  EXIT    TST             ; exit test
     010402  104432                          TRAP    C$EXIT
     010404  000210                          .WORD   L10023-.
2650
2651 010406  013737  002276  002272  contD6: MOV     LNNBRy, LNNBR   ; else report DEVICE FATAL error
2652 010414                                  ERRDF   3, DZLBER, pDZLB;
     010414  104455                          TRAP    C$ERDF
     010416  000003                          .WORD   3
     010420  003125                          .WORD   DZLBER
     010422  005240                          .WORD   pDZLB
2653 010424                                  DOCLN                   ; Abort pass
     010424  104444                          TRAP    C$DCLN
2654
```

K6

```
                                        ; subroutine to transmit one character in maintenance loopback mode
                                        ; and check for echo
                                        ;
                                        ; Output parameter : sbAOK = 1 <=> success
                                        ;
010426  005037  002306          sbTED1: CLR     sbAOK           ;
010432  012777  000050  171622          MOV     #MSEMAI, @DZCSRa; Enable maintenance loopback mode transmission
010440  012701  177754                  MOV     #DLAY2s, R1     ; Set up 2 seconds delay
010444  005777  171612          nTRYD1: TST     @DZCSRa         ; Wait
010450  100414                          BMI     contD7          ; for
                                        DLAY    DLAYarg         ; CSR Transmit ReaDY bit
010472  005201                          INC     R1              ; to set
010474  001363                          BNE     nTRYD1          ; If time-out
010476  000137  010610                  JMP     PBLMD1          ; then there's a problem
010502  112777  000101  171560  contD7: MOVB    #'A, @DZTDRa    ; Load character into Transmit Data Register
010510  012701  177754                  MOV     #DLAY2s, R1     ; Set up 2 seconds delay
010514  105777  171542          nRDND1: TSTB    @DZCSRa         ; REPEAT Wait
010520  100021                          BPL     contD8          ; UNTIL echo received
010522  017737  171536  002250          MOV     @DZRBUFa, ECHO  ; Read received data
010530  122737  000101  002250          CMPB    #'A, ECHO       ; If data received differs from data sent,
010536  001024                          BNE     PBLMD1          ; then there is a problem
010540  000240                          NOP
010542  000240                          NOP
010544  017737  171514  002250          MOV     @DZRBUFa, ECHO  ; Try and read more data
010552  100416                          BMI     PBLMD1          ; If silo is not empty, there is a problem
010554  012737  000001  002306          MOV     #1, sbAOK       ; else All is OK
010562  000207                          RTS     PC              ;
010564                          contD8: DLAY    DLAYarg         ; Wait
010604  005201                          INC     R1              ; routine
010606  001342                          BNE     nRDND1
010610  000207                  PBLMD1: RTS     PC              ; When delay is elapsed or a problem arises,
                                                                ; return with sbAOK = 0
                                ; end sbTED1

                                ENDSUB
010612                          L10025:
010612  104403                          TRAP    C$ESUB

                                ENDTST
010614                          L10023:
010614  104401                          TRAP    C$ETST
                                        .EVEN
```

HARDWARE TESTS

```
                                BADHEAD
                ;****************************** TEST2 ****************************
                ;*
                ;* Test active only in mode 0 :
                ;*
                ;*      Purpose : check that  characters  are echoed back
                ;*                from the GS03-WD.
                ;*
                ;*      Assumption :  the previous test ran successfully.
                ;*
                ;*      Description :
                ;*      The  two  lines  out of the DZ11  are arbitrarily
                ;*      named line x and line y.
                ;*      A first  attempt will be  made  to  receive  echo
                ;*      back from  the GS03-WD  on  line  x. If it is not
                ;*      successful, another  attempt  will  be  made  to
                ;*      receive  echo  on  line  y. If  this  cannot  be
                ;*      achieved  either, a  hard  error warning  will be
                ;*      printed.
                ;*
                ;*      Note :
                ;*      This diagnostic detects that the GS03-WD switches
                ;*      to one line  by receiving  echoed characters back
                ;*      from the GS03-WD on that line.
                ;*      This is why,  before  other  tests  check correct
                ;*      switching, this test first checks that  echo  can
                ;*      be received back from the  GS03-WD, on  at  least
                ;*      one line.
                ;*
                ;*      Error message :
                ;* @4   - "No echo received back from the GS03-WD on
                ;*      either line # <line number> or # <line number>"
                ;*        "Check cabling and dip switch E18 (must be OFF)."
                ;*
                BADHEAD
                ;****************************** TEST2 ****************************
```

M6

HARDWARE TESTS

```
2745 010616                            BGNTST
     010616                            T2::
2746                                   ; Initialization :
2747 010616  005737  002214                    TST     TMODE           ; If mode 1, then skip this test
2748 010622  001402                            BEQ     contL1
2749 010624                                    EXIT    TST
     010624  104432                            TRAP    C$EXIT
     010626  000120                            .WORD   L10026-.
2750
2751 010630  004737  002572           contL1: JSR     PC, @bIDG1      ; Initialize DZ11 for interrupt mode
2752                                           ;                       ; transmission
2753
2754                                   ; Test echo on line x :
2755
2756 010634  013777  002240  171422            MOV     TLPRx, @DZLPRa  ; Load parameters for line x
2757 010642  113777  002302  171416            MOVB    LNMAPx, @DZTCRa ; Enable transmission on line x
2758
2759 010650  013737  002274  002310            MOV     LNNBRx, LNTSTD  ; Transmit
2760 010656  004737  002642                    JSR     PC, @bTEG1      ; and test echo on line x
2761
2762 010662  005737  002306                    TST     @bAOK           ; if successful
2763 010666  001022                            BNE     endL1           ; then shut off DZ11 and exit test
2764
2765                                   ; no receive on line x : test echo on line y :
2766
2767 010670  013777  002242  171366            MOV     TLPRy, @DZLPRa
2768 010676  113777  002304  171362            MOVB    LNMAPy, @DZTCRa
2769
2770 010704  013737  002276  002310            MOV     LNNBRy, LNTSTD  ; Transmit
2771 010712  004737  002642                    JSR     PC, @bTEG1      ; and test echo on line y
2772
2773 010716  005737  002306                    TST     @bAOK           ; if successful
2774 010722  001004                            BNE     endL1           ; then shut off DZ11 and exit test
2775
2776                                   ; no receive on line y either : there is a problem
2777
2778 010724                             ERRR11: ERRHRD  4, NOWDEC, pNWDE; Report error
     010724  104456                            TRAP    C$ERHRD
     010726  000004                            .WORD   4
     010730  003112                            .WORD   NOWDEC
     010732  005332                            .WORD   pNWDE
2779
2780 010734  012777  000020  171320    endL1: MOV     @CSRCLR, @DZCSRa; Shut off DZ11
2781 010742                                    ESCAPE  TST
     010742  104410                            TRAP    C$ESCAPE
     010744  000002                            .WORD   L10026-.
2782
2783 010746                            ENDTST
     010746                            L10026:
     010746  104401                            TRAP    C$ETST
2784                                           .EVEN
```

M6

HARDWARE TESTS

```
 2787 010750                    BADHEAD
                                ;*************************** TEST3 ***************************
 2788                           ;*
 2789                           ;* Test active only in mode 0 :
 2790                           ;*
 2791                           ;*      Purpose : switch the GS03-WD back and forth.
 2792                           ;*
 2793                           ;*      Assumptions :
 2794                           ;*      - all previous tests ran successfully ;
 2795                           ;*      - WATCHDOG FUNCTION has priority (cf. note).
 2796                           ;*
 2797                           ;*      Description :
 2798                           ;*      This  test is the implementation of the following
 2799                           ;*      algorithm :
 2800                           ;*
 2801                           ;*      Repeat  twice,  swapping  lines x and y, the se-
 2802                           ;*      quence :
 2803                           ;*              - Try and switch GS03-WD to line x ;
 2804                           ;*              - Try and switch GS03-WD from  line x  to
 2805                           ;*              line y ;
 2806                           ;*              - Try and switch GS03-WD back from line y
 2807                           ;*               to line x ;
 2808                           ;*
 2809                           ;*      Note :
 2810                           ;*      This diagnostic assumes that the switches are set
 2811                           ;*      to  give  the  WATCHDOG  FUNCTION  priority. This
 2812                           ;*      means that the front panel switches should all be
 2813                           ;*      in  the  center  position and  the relay  modules
 2814                           ;*      should all be configured for  the  same  priority
 2815                           ;*      (see Option Description for details).
 2816                           ;*
 2817                           ;*      IMPROPER SETTINGS  CAN  CAUSE  UNEXPECTED ERRORS,
 2818                           ;*      WHICH WILL NOT NECESSARILY  BE DIAGNOSED AS SUCH.
 2819                           ;*
 2820                           ;*      Error messages :
 2821                           ;* #5   - "No echo received back from GS03-WD on line
 2822                           ;*         # <line number>"
 2823                           ;*           "Check FORCE, MANUAL switches, priority
 2824                           ;*            setting and cables".
 2825                           ;* #6   - "Both lines have switch priority over each
 2826                           ;*         other."
 2827                           ;*           "Check GS03 configuration."
 2828                           ;* #7   - "Echo from GS03-WD received on both lines
 2829                           ;*         # <line number> and # <line number>."
 2830                           ;* #8   - "GS03-WD failed to switch to line # <line
 2831                           ;*         number>"
 2832                           ;*           "No echo received back from GS03-WD on line
 2833                           ;*            # <line number>"
 2834                           ;*           "Check FORCE, MANUAL switches, priority
 2835                           ;*            setting and cables".
 2836                           ;* #9   - "Echo from the GS03-WD received on wrong
 2837                           ;*         line # <line number> (expected : # <line
 2838                           ;*         number>)."
 2839                           ;*           "Echo is still being received on line
 2840                           ;*            # <line number> when actually transmitting
 2841                           ;*            on line # <line number> only."
 2842                           ;*           "Check GS03 configuration."
```

```
HARDWARE TESTS

 2843                                   ;*
 2844 010750                            BADHEAD
                                        ;********************************* TEST3 **************************

 2845
```

```
HARDWARE TESTS

 2848 010750                            BGNTST
      010750                            T3::
 2849                                   ; Initialization :
 2850 010750  005737  002214                    TST     TMODE           ; If mode 1, then skip this test
 2851 010754  001402                            BEQ     contG1
 2852 010756                                    EXIT    TST
      010756  104432                            TRAP    C$EXIT
      010760  000522                            .WORD   L10027-.
 2853
 2854 010762  004737  002572            contG1: JSR     PC, @bIDG1      ; Initialize DZ11 for interrupt mode
 2855                                           ;                       ; transmission
 2856
 2857 010766  013777  002240  171270            MOV     TLPRx, @DZLPR   ; Load parameters for line x
 2858 010774  013777  002242  171262            MOV     TLPRy, @DZLPR   ; Load parameters for line y
 2859
 2860 011002  005037  011500                    CLR     FTIMG1
 2861
 2862 011006  005037  002220            reptG1: CLR     SWPRTY          ; Set switch priority to none.
 2863
 2864                                   ; Try and switch the GS03-MD to line x
 2865
 2866 011012  113777  002302  171246            MOVB    LNMAPx, @DZTCR  ; Enable transmission on line x
 2867
 2868 011020  013737  002274  002310            MOV     LNNBRx, LNTSTD  ; Transmit
 2869 011026  004737  002642                    JSR     PC, @bTEG1      ; and test for echo on line x
 2870
 2871 011032  005737  002306                    TST     @bAOK           ; If successful,
 2872 011036  001012                            BNE     contG2          ; then go on testing
 2873
 2874 011040  013737  002274  002272            MOV     LNNBRx, LNNBR   ; else report error
 2875 011046                                    ERRMRD  5, NO1LEC, pN1LE; and
      011046  104456                            TRAP    C$ERMRD
      011050  000005                            .WORD   5
      011052  003066                            .WORD   NO1LEC
      011054  005410                            .WORD   pN1LE
 2876 011056  012777  000020  171176            MOV     @CSRCLR, @DZCSR ; shut off DZ11
 2877
 2878 011064                            contG2: ESCAPE TST
      011064  104410                            TRAP    C$ESCAPE
      011066  000414                            .WORD   L10027-.
 2879
 2880                                   ; Try and switch the GS03-MD from line x to line y
 2881
 2882 011070  013737  002274  002312            MOV     LNNBRx, OLDLNNB ; Load
 2883 011076  013737  002276  002316            MOV     LNNBRy, NEWLNNB ; parameters
 2884 011104  013737  002304  002314            MOV     LNMAPy, NEWLNMP ;
 2885
 2886 011112  004737  011252                    JSR     PC, @bSWG1      ; and test
 2887
 2888 011116                                    ESCAPE  TST
      011116  104410                            TRAP    C$ESCAPE
      011120  000362                            .WORD   L10027-.
 2889
 2890                                   ; Try and switch the GS03-MD from line y to line x
 2891
 2892 011122  013737  002276  002312            MOV     LNNBRy, OLDLNNB ; Load
 2893 011130  013737  002274  002316            MOV     LNNBRx, NEWLNNB ; parameters
```

```
HARDWARE TESTS

 2894 011136  013737  002302  002314          MOV     LNMAPx, NEWLNMP ;
 2895
 2896 011144  004737  011252                  JSR     PC, @bSWG1      ; end test
 2897
 2898 011150                                  ESCAPE  TST
      011150  104410                          TRAP    C$ESCAPE
      011152  000330                          .WORD   L10027-.
 2899
 2900 011154  005737  011500                  TST     FTIMG1
 2901 011160  001027                          BNE     endG1
 2902
 2903                                 ; Swap lines x and y and repeat this test :
 2904 011162  013737  002274  002272          MOV     LNNBRx, LNNBR   ;
 2905 011170  013737  002276  002274          MOV     LNNBRy, LNNBRx  ;
 2906 011176  013737  002272  002276          MOV     LNNBR, LNNBRy   ;
 2907
 2908 011204  013737  002302  002272          MOV     LNMAPx, LNNBR   ; LNNBR is used as a temporary here
 2909 011212  013737  002304  002302          MOV     LNMAPy, LNMAPx  ;
 2910 011220  013737  002272  002304          MOV     LNNBR, LNMAPy   ;
 2911
 2912 011226  012737  000001  011500          MOV     #1, FTIMG1
 2913 011234  000137  011006                  JMP     reptG1
 2914
 2915                                 ; End of test :
 2916 011240  012777  000020  171014  endG1: MOV     #CSRCLR, @DZCSR ; Shut off DZ11
 2917 011246                                  EXIT    TST
      011246  104432                          TRAP    C$EXIT
      011250  000232                          .WORD   L10027-.
```

HARDWARE TESTS

```
                                        ; subroutine to try and switch the GS03-WD from line OLDLNNB to line NEWLNNB
                                        ;
                                        ; Assumption : line OLDLNNB is already alive.
                                        ;
2924 011252  013777  002300  171006  sbSWG1: MOV     LNMAP, @DZTCRa  ; Enable transmission on both lines
2926 011260  013737  002316  002310          MOV     NEWLNNB, LNTSTD ; Start transmitting
2927 011266  004737  002642                  JSR     PC, sbTEG1      ; and test for echo on line NEWLNNB
2929 011272  005737  002306                  TST     sbAOK           ; If echo on line NEWLNNB,
2930 011276  001432                          BEQ     contG4          ;
                                                                     ; then
2932 011300  005737  002220                  TST     SWPRTY          ; begin line NEWLNNB has switch priority
2933 011304  001406                          BEQ     contG3          ;       so, if OLDLNNB already had it :
2935 011306                                  ERRHRD  6, PTYCFL, pPYCF;       'Both lines have switch priority...'
     011306  104456                          TRAP    C$ERHRD
     011310  000006                          .WORD   6
     011312  003044                          .WORD   PTYCFL
     011314  005462                          .WORD   pPYCF
                                                                     ;       'Check FORCE, MANUAL...'
2937 011316                                  ESCAPE  TST
     011316  104410                          TRAP    C$ESCAPE
     011320  000162                          .WORD   L10027-.
2939 011322  012737  000001  002220  contG3: MOV     #1, SWPRTY      ;       else record that NEWLNNB has priority
2941 011330  013737  002312  002310          MOV     OLDLNNB, LNTSTD ;       If there is still echo
2942 011336  004737  002642                  JSR     PC, sbTEG1      ;       on OLDLNNB,
2943 011342  005737  002306                  TST     sbAOK           ;       that means there's echo on both lines :
2944 011346  001406                          BEQ     contG4          ;
2946 011350                                  ERRHRD  7, WD2ECH, pWD2E;       'Echo from GS03-WD received on both
     011350  104456                          TRAP    C$ERHRD
     011352  000007                          .WORD   7
     011354  003021                          .WORD   WD2ECH
     011356  005530                          .WORD   pWD2E
                                                                     ;       lines.'
2948 011360                                  ESCAPE  TST
     011360  104410                          TRAP    C$ESCAPE
     011362  000120                          .WORD   L10027-.
                                                                     ; end
2951 011364  013777  002314  170674  contG4: MOV     NEWLNMP, @DZTCRa; Stop transmitting on line OLDLNNB
2953 011372  013737  002316  002310          MOV     NEWLNNB, LNTSTD ; If no echo
2954 011400  004737  002642                  JSR     PC, sbTEG1      ; on line
2955 011404  005737  002306                  TST     sbAOK           ; NEWLNNB,
2956 011410  001011                          BNE     contG5          ; then :
2958 011412  013737  002316  002272          MOV     NEWLNNB, LNNBR  ; 'GS03-WD fail to switch to line <NEWLNNB>'
2959 011420                                  ERRHRD  8, FAISWT, pFSWT; 'No echo received on line <NEWLNNB>'
     011420  104456                          TRAP    C$ERHRD
     011422  000010                          .WORD   8
     011424  002777                          .WORD   FAISWT
     011426  005566                          .WORD   pFSWT
                                                                     ; 'Check FORCE, MANUAL...'
```

F7


```
HARDWARE TESTS

 2961 011430                                        ESCAPE  TST
      011430  104410                                TRAP    C$ESCAPE
      011432  000050                                .WORD   L10027-.
 2962
 2963 011434  013737  002312  002310  contG5: MOV     OLDLNNB, LNTSTD ; If there is
 2964 011442  004737  002642                  JSR     PC, sbTEG1      ; still echo
 2965 011446  005737  002306                  TST     sbAOK           ; on line
 2966 011452  001411                          BEQ     esbG1           ; OLDLNNB :
 2967
 2968 011454  013737  002312  002272          MOV     OLDLNNB, LNNBR  ; 'GS03-WD fail to switch from line <OLDLNNB>'
 2969 011462                                   ERRHRD  9, FAISWF, pFSWF; 'Echo is still being received
      011462  104456                          TRAP    C$ERHRD
      011464  000011                          .WORD   9
      011466  002754                          .WORD   FAISWF
      011470  005664                          .WORD   pFSWF
 2970                                                                  ; on line <OLDLNNB>'
 2971                                                                  ; 'Check FORCE, MANUAL...'
 2972 011472                                        ESCAPE  TST
      011472  104410                                TRAP    C$ESCAPE
      011474  000006                                .WORD   L10027-.
 2973
 2974 011476  000207                  esbG1:  RTS     PC
 2975                                 ; end sbSWG1
```

G7

HARDWARE TESTS

```
 2984
 2985 011500  000000                FTIMG1: .WORD   0               ; Boolean value to flag first run through test
 2986
 2993
 2994 011502                         ENDTST
      011502                         L10027:
      011502  104401                         TRAP    C$ETST
 2995                                        .EVEN
 2996
```

HARDWARE TESTS

```
 2999 011504                                    BADHEAD
                                                ;****************************** TEST4 ****************************
 3000                                           ;*
 3001                                           ;* Test active only in mode 1 :
 3002                                           ;*
 3003                                           ;*      Purpose : installation test.
 3004                                           ;*
 3005                                           ;*      Assumption : all previous tests ran successfully.
 3006                                           ;*
 3007                                           ;*      Description :
 3008                                           ;*      This  test activates the line into the GS03-WD in
 3009                                           ;*      order for the  operator to  check  that the LED's
 3010                                           ;*      react correctly :
 3011                                           ;*
 3012                                           ;*      The  GREEN  or YELLOW  LED  corresponding to this
 3013                                           ;*      CPU's line into the GS03-WD should  then turn on.
 3014                                           ;*      The associated RED LED should turn off  after one
 3015                                           ;*      full GS03-WD clock pulse after  this test  begins
 3016                                           ;*      (which means that the RED  clock LED should blink
 3017                                           ;*      twice at the most before this happens).
 3018                                           ;*
 3019                                           ;*      Error message : none.
 3020                                           ;*
 3021 011504                                    BADHEAD
                                                ;****************************** TEST4 ****************************
 3022
 3023 011504                            BGNTST
      011504                            T4::
 3024                                   ; Initialization :
 3025 011504  005737  002214                    TST     TMODE           ; If mode 0, then skip this test
 3026 011510  001002                            BNE     contA1
 3027 011512                                    EXIT    TST
      011512  104432                            TRAP    C$EXIT
      011514  000042                            .WORD   L10030-.
 3028
 3029 011516  004737  002572            contA1: JSR     PC, @bIDG1      ; Initialize DZ11 for interrupt mode
 3030                                           ;                       ; transmission
 3031
 3032                                   ; Transmit on line x :
 3033
 3034 011522  042737  010000  002240            BIC     #RCVRON, TLPRx  ; Receiver will not be used
 3035 011530  013777  002240  170526            MOV     TLPRx, @DZLPR   ; Load parameters for line x
 3036 011536  113777  002302  170522            MOVB    LNMAPx, @DZTCR  ; Enable transmission on line x
 3037
 3038 011544  012777  040040  170510            MOV     #MSETIE, @DZCSR ; Enable interrupt mode transmission
 3039
 3040 011552                            loopA1: BREAK
      011552  104422                            TRAP    C$BRK
 3041 011554  000776                            BR      loopA1
 3042
 3043 011556                            ENDTST
      011556                            L10030:
      011556  104401                            TRAP    C$ETST
 3044                                           .EVEN
 3045
```

HARDWARE PARAMETER CODING SECTION

```
                                        .SBTTL HARDWARE PARAMETER CODING SECTION
                                ;////////////////////////////////////////////////////////////////////////
                                ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ;/ WITH THE OPERATOR.
                                ;////////////////////////////////////////////////////////////////////////
                                        BGN$MD
011560  000053                          .WORD L10031-L$HARD/2
011562                          L$HARD::
                                        GPRMA   CSR, 0, 0, 160010, 163776, YES
011562  000031                          .WORD   T$CODE
011564  011642                          .WORD   CSR
011566  160010                          .WORD   T$LOLIM
011570  163776                          .WORD   T$HILIM
                                        GPRMA   VECTOR, 2, 0, 300, 777, YES
011572  001031                          .WORD   T$CODE
011574  011646                          .WORD   VECTOR
011576  000300                          .WORD   T$LOLIM
011600  000777                          .WORD   T$HILIM
                                        GPRMD   PRIORTY, 4, 0, 000007, 4, 7, YES
011602  002032                          .WORD   T$CODE
011604  011655                          .WORD   PRIORTY
011606  000007                          .WORD   000007
011610  000004                          .WORD   T$LOLIM
011612  000007                          .WORD   T$HILIM
                                        GPRMD   ACLINES, 6, 0, 000377, 1, 377, YES
011614  003032                          .WORD   T$CODE
011616  011660                          .WORD   ACLINES
011620  000377                          .WORD   000377
011622  000001                          .WORD   T$LOLIM
011624  000377                          .WORD   T$HILIM
                                        GPRMD   WCHMODE, 10, 0, 177777, 0, 1, YES
011626  004032                          .WORD   T$CODE
011630  011675                          .WORD   WCHMODE
011632  177777                          .WORD   177777
011634  000000                          .WORD   T$LOLIM
011636  000001                          .WORD   T$HILIM

011640                                  EXIT    MRD
011640  024004                          .WORD   T$CODE

                                        .nlist  BEX
011642  103 123 122     CSR:    .ASCIZ  /CSR/
011646  126 105 103  VECTOR:    .ASCIZ  /VECTOR/
011655  102 122 000  PRIORTY:   .ASCIZ  /BR/
011660  101 103 124  ACLINES:   .ASCIZ  /ACTIVE LINES/
011675  127 110 111  WCHMODE:   .ASCIZ  /WHICH MODE/
                                        .list   BEX
```

J7

HARDWARE PARAMETER CODING SECTION

```
3085                                    .EVEN
3086
3087 011710                             ENDMRD
                                        .EVEN
     011710                        L10031:
3088
3089
3090
3091
3092
3093
```

K7

SOFTWARE PARAMETER CODING SECTION

```
3095                                    .sbttl  SOFTWARE PARAMETER CODING SECTION
3096
3097
3098                                    ;////////////////////////////////////////////////////////////////////////
3099                                    ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
3100                                    ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
3101                                    ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
3102                                    ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
3103                                    ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
3104                                    ;/ WITH THE OPERATOR.
3105                                    ;////////////////////////////////////////////////////////////////////////
3106
3107 011710                                     BGNSFT
      011710  000000                           .WORD L10032-L$SOFT/2
      011712                            L$SOFT::
3108
3109
3118
3119 011712                                     ENDSFT
                                                .EVEN
      011712                            L10032:
3120
3121
3128
3129
```

SOFTWARE PARAMETER CODING SECTION

```
3131
3132 011712                             $PATCH::
3133 011712                                     .BLKW   70
3140
3141 012072                                     LASTAD
                                                .EVEN
     012072  012114                             .WORD T$FREE
     012074  000007                             .WORD T$SIZE
     012076                             L$LAST::
3142 012076                                     ENDMOD
3143
3144
```

SOFTWARE PARAMETER CODING SECTION

```
3146
3147
3160
3161 012076                                  BGNSETUP        1
3162 012076                                  BGNPTAB
     012076  000000                          .WORD   0
     012100  000005                          .WORD   L10035-./2-1
     012102                          L10033:
3163
3164 012102  160100                          .word   160100
3165 012104  000300                          .word   300
3166 012106  000005                          .word   5
3167 012110  000003                          .word   3
3168 012112  000000                          .word   0
3169
3170 012114                                  ENDPTAB
     012114                          L10035:
3171 012114                                  ENDSETUP
3172
3173
3174
3175
3176
3177          000001                 .END
```

SYMBOL TABLE

```
ABORI1  006500          CSRCLR= 000020 G        DIAGMC= 000000          F$PWR  = 000017          LNNBR   002272
ACLINE  011660          CSRR    004213          DLAYAR= 000100 G        F$RPT  = 000012          LNNBRX  002274
ADDED   007756          CSRW    004116          DLAYC1  002244          F$SEG  = 000003          LNNBRY  002276
ADDR    002320          C$AU   = 000052          DLAYC2  002246          F$SOFT= 000005          LNTSTD  002310
ADR    = 000020 G       C$AUTO= 000061          DLAY11= 177622 G        F$SRV  = 000010          LOE    = 040000 G
ASSEMB= 000010          C$BRK  = 000022          DLAY2S= 177754 G        F$SUB  = 000002          LOOPA1  011552
BIT0   = 000001 G       C$BSEG= 000004          DROPD   007676          F$SW   = 000014          LOOPG1  002346
BIT00  = 000001 G       C$BSUB= 000002          DZCSRA  002262          F$TEST= 000001          LOT    = 000010 G
BIT01  = 000002 G       C$CEFG= 000045          DZDIAG  004744          GTPMI1  006176          L$ACP   002110 G
BIT02  = 000004 G       C$CLCK= 000062          DZIN    004064          G$CNTO= 000200          L$APT   002036 G
BIT03  = 000010 G       C$CLEA= 000012          DZINER  003162          G$DELM= 000372          L$AU    007730 G
BIT04  = 000020 G       C$CLOS= 000035          DZLB    003776          G$DISP= 000003          L$AUT   002070 G
BIT05  = 000040 G       C$CLP1= 000006          DZLBER  003125          G$EXCP= 000400          L$AUTO  007544 G
BIT06  = 000100 G       C$CVEC= 000036          DZLPRA  002264          G$HILI= 000002          L$CCP   002106 G
BIT07  = 000200 G       C$DCLN= 000044          DZPTY   002234          G$LOLI= 000001          L$CLEA  007636 G
BIT08  = 000400 G       C$DODU= 000051          DZRBUF  002264          G$NO   = 000000          L$CO    002032 G
BIT09  = 001000 G       C$DRPT= 000024          DZRVCC  002252          G$OFFS= 000400          L$DEPO  002011 G
BIT1   = 000002 G       C$DU   = 000053         DZRVCS  002254          G$OFSI= 000376          L$DESC  002156 G
BIT10  = 002000 G       C$EDIT= 000003          DZTCRA  002266          G$PRMA= 000001          L$DESP  002076 G
BIT11  = 004000 G       C$ERDF= 000055          DZTDRA  002270          G$PRMD= 000002          L$DEVP  002060 G
BIT12  = 010000 G       C$ERMR= 000056          DZTVCC  002256          G$PRML= 000000          L$DISP  002132 G
BIT13  = 020000 G       C$ERRO= 000060          DZTVCS  002260          G$RADA= 000140          L$DLY   002116 G
BIT14  = 040000 G       C$ERSF= 000054          ECHO    002250          G$RADB= 000000          L$DTP   002040 G
BIT15  = 100000 G       C$ERSO= 000057          EF.CON= 000036 G        G$RADD= 000040          L$DTYP  002034 G
BIT2   = 000004 G       C$ESCA= 000010          EF.NEW= 000035 G        G$RADL= 000120          L$DU    007650 G
BIT3   = 000010 G       C$ESEG= 000005          EF.PWR= 000034 G        G$RADO= 000020          L$DUT   002072 G
BIT4   = 000020 G       C$ESUB= 000003          EF.RES= 000037 G        G$XFER= 000004          L$DVTY  002324 G
BIT5   = 000040 G       C$ETST= 000001          EF.STA= 000040 G        G$YES  = 000010          L$EF    002052 G
BIT6   = 000100 G       C$EXIT= 000032          ENDG1   011240          HELP   = 000000          L$ENVI  002044 G
BIT7   = 000200 G       C$GETB= 000026          ENDL1   010734          HOE    = 100000 G       L$ETP   002102 G
BIT8   = 000400 G       C$GETW= 000027          ERLMI1  006624          IBE    = 010000 G       L$EXP1  002046 G
BIT9   = 001000 G       C$GMAN= 000043          ERRCNT  002224          IDU    = 000040 G       L$EXP4  002064 G
BOE    = 000400 G       C$GPHR= 000042          ERRR11  010724          IER    = 020000 G       L$EXP5  002066 G
BUSTIM  003214          C$GPLO= 000030          ESBG1   011476          ISR    = 000100 G       L$HARD  011562 G
CHKMAX  002400          C$GPRI= 000040          EVL    = 000004 G       IXE    = 004000 G       L$HIME  002120 G
CKDPSW  004622          C$INIT= 000011          E$END  = 002100          I$AU   = 000041          L$HPCP  002016 G
CKDZAD  004775          C$INLP= 000020          E$LOAD= 000035          I$AUTO= 000041          L$HPTP  002022 G
CKFMSW  004523          C$MANI= 000050          FAISWF  002754          I$CLN  = 000041          L$HW    002144 G
CKGSCF  004706          C$MEM  = 000031         FAISWT  002777          I$DU   = 000041          L$ICP   002104 G
CONTA1  011516          C$MSG  = 000023         FSWF    003230          I$HRD  = 000041          L$INIT  005774 G
CONTD1  010104          C$OPEN= 000034          FSWT    003343          I$INIT= 000041          L$LADP  002026 G
CONTD2  010172          C$PNTB= 000014          FTIME   002212          I$MOD  = 000041          L$LAST  012076 G
CONTD3  010274          C$PNTF= 000017          FTIMG1  011500          I$MSG  = 000041          L$LOAD  002100 G
CONTD4  010342          C$PNTS= 000016          FTUNI1  006114          I$PROT= 000040          L$LUN   002074 G
CONTD5  010354          C$PNTX= 000015          F$AU   = 000015         I$PTAB= 000041          L$MREV  002050 G
CONTD6  010406          C$QIO  = 000377         F$AUTO= 000020          I$PWR  = 000041          L$NAME  002000 G
CONTD7  010502          C$RDBU= 000007          F$BGN  = 000040         I$RPT  = 000041          L$PRIO  002042 G
CONTD8  010564          C$REFG= 000047          F$CLEA= 000007          I$SEG  = 000041          L$PROT  002122 G
CONTG1  010762          C$RESE= 000033          F$DU   = 000016         I$SETU= 000041          L$PRT   002112 G
CONTG2  011064          C$REVI= 000003          F$END  = 000041         I$SFT  = 000041          L$REPP  002062 G
CONTG3  011322          C$RFLA= 000021          F$HARD= 000004          I$SRV  = 000041          L$REV   002010 G
CONTG4  011364          C$RPT  = 000025         F$HW   = 000013         I$SUB  = 000041          L$RPT   005772 G
CONTG5  011434          C$SEFG= 000046          F$INIT= 000006          I$TST  = 000041          L$SOFT  011712 G
CONTI1  006104          C$SPRI= 000041          F$JMP  = 000050         J$JMP  = 000167          L$SPC   002056 G
CONTI2  006442          C$SVEC= 000037          F$MOD  = 000000         LNMAP   002300          L$SPCP  002020 G
CONTL1  010630          C$TPRI= 000013          F$MSG  = 000011         LNMAPX  002302          L$SPTP  002024 G
CSR     011642          DFPTBL  002144 G        F$PROT= 000021          LNMAPY  002304          L$STA   002030 G
```

SYMBOL TABLE

```
L$TEST  002114 G    NEWLNN  002316      PRIO4 = 000200 G    TLPRO   002236      T$$DAT= 010035
L$TIML  002014 G    NFTMI1  006024      PRIO5 = 000240 G    TMAYER  002472      T$$DU = 010021
L$UNIT  002012 G    NOEC1L  004304      PRIO6 = 000300 G    TMODE   002214      T$$HAR= 010031
L10001  002156      NOTAV   007316      PRIO7 = 000340 G    TXPSW   002232      T$$HW = 010001
L10002  002570      NOWDEC  003112      PTYCFL  003044      T$ARGC= 000002      T$$INI= 010016
L10003  005076      NO1LEC  003066      PWD2E   005530 G    T$CODE= 024004      T$$MSG= 010014
L10004  005150      NRDWD1  010514      PYCF    003515      T$ERRN= 000011      T$$PC = 000001
L10005  005236      NRDWG1  002660      RBUFCT= 174000 G    T$EXCP= 000000      T$$PRO= 010000
L10006  005330      NTRYD1  010444      RCVRON= 010000 G    T$FLAG= 000041      T$$PTA= 010034
L10007  005406      NWDE    003667      REPTG1  011006      T$FREE= 012114      T$$RPT= 010015
L10010  005460      NXBII1  006536      RLMPI1  006554      T$GMAN= 000000      T$$SEG= 010001
L10011  005526      NXUNI1  006122      RLMPI2  006676      T$MILI= 000001      T$$SOF= 010032
L10012  005564      N1LE    003602      ROPMAP  010006      T$LAST= 000001      T$$SRV= 010002
L10013  005662      OLDLNN  002312      RUNGOA  006740      T$LOLI= 000000      T$$SUB= 010025
L10014  005770      O$APTS= 000000      RUNGOB  007056      T$LSYM= 010000      T$$TES= 010030
L10015  005772      O$AU  = 000001      RUNG1A  007141      T$LTNO= 000004      T1      010006 G
L10016  007542      O$BGNR= 000000      RUNG1B  007236      T$NEST= 177777      T1.1    010006
L10017  007634      O$BGNS= 000000      SAVE4   002226      T$NS0 = 000000      T1.2    010212
L10020  007646      O$DU  = 000001      SAVE6   002230      T$NS1 = 000005      T2      010616 G
L10021  007726      O$ERRT= 000000      SBACK   002306      T$NS2 = 000002      T3      010750 G
L10022  010004      O$GNSW= 000000      SBIDG1  002572      T$NS3 = 000003      T4      011504 G
L10023  010614      O$POIN= 000001      SBLNI1  006506      T$PCNT= 000000      UAM   = 000200 G
L10024  010210      O$SETU= 000001      SBSWG1  011252      T$PTAB= 010034      UNIT    002322
L10025  010612      PBLMD1  010610      SBTED1  010426      T$PTMV= 000001      UUT     002216
L10026  010746      PCSAR   005026 G    SBTEG1  002642      T$PTNU= 000001      VECTOR  011646
L10027  011502      PCSAW   005100 G    SBWTG1  002342      T$SAVL= 177777      WAITG1  002716
L10030  011556      PDZIN   005152 G    SBWTG2  002362      T$SEGL= 177777      WCMMOD  011675
L10031  011710      PDZLB   005240 G    STEC1L  004371      T$SEXO= 010001      WD2E    003416
L10032  011712      PFSWF   005664 G    SUCCG1  002744      T$SIZE= 000007      WD2ECM  003021
L10033  012102      PFSWT   005666 G    SUCCI1  006730      T$SUBN= 000000      WGLMP1  007357
L10035  012114      PNT   = 001000 G    SVCGBL= 000000      T$TAGL= 177777      WGLMP2  007467
MAXERR  002222      PNADE   005332 G    SVCINS= 000000      T$TAGN= 010036      X$ALWA= 000000
MAXPRI= 000340 G    PNILE   005410 G    SVCSUB= 000000      T$TEMP= 000000      X$FALS= 000040
MOD1I1  006372      PPYCF   005462 G    SVCTAG= 000000      T$TEST= 000004      X$OFFS= 000400
MSEMAI= 000050 G    PRI   = 002000 G    SVCTST= 000000      T$TSTM= 177777      X$TRUE= 000020
MSETIE= 040040 G    PRIORT  011655      SVTXG1  002562 G    T$TSTS= 000001      ZDZGA0  002000 G
NCLDD1  010226      PRIO0 = 000000 G    SWPRTY  002220      T$$AU = 010022      $LSTIN= 000000
NCLDG1  002626      PRIO1 = 000040 G    S$LSYM= 010000      T$$AUT= 010017      $LSTTA= 000000
NETYG1  002666      PRIO2 = 000100 G    TLPRX   002240      T$$CLE= 010020      $PATCH  011712 G
NEWLNM  002314      PRIO3 = 000140 G    TLPRY   002242

. ABS.  012114      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28900 WORDS  ( 113 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:02:28
ZDZGA0.BIN,ZDZGA0-[PINASA.D.GS.PDP.SSDC]LIBA.MLB/ML,ZDZGA0
.
```