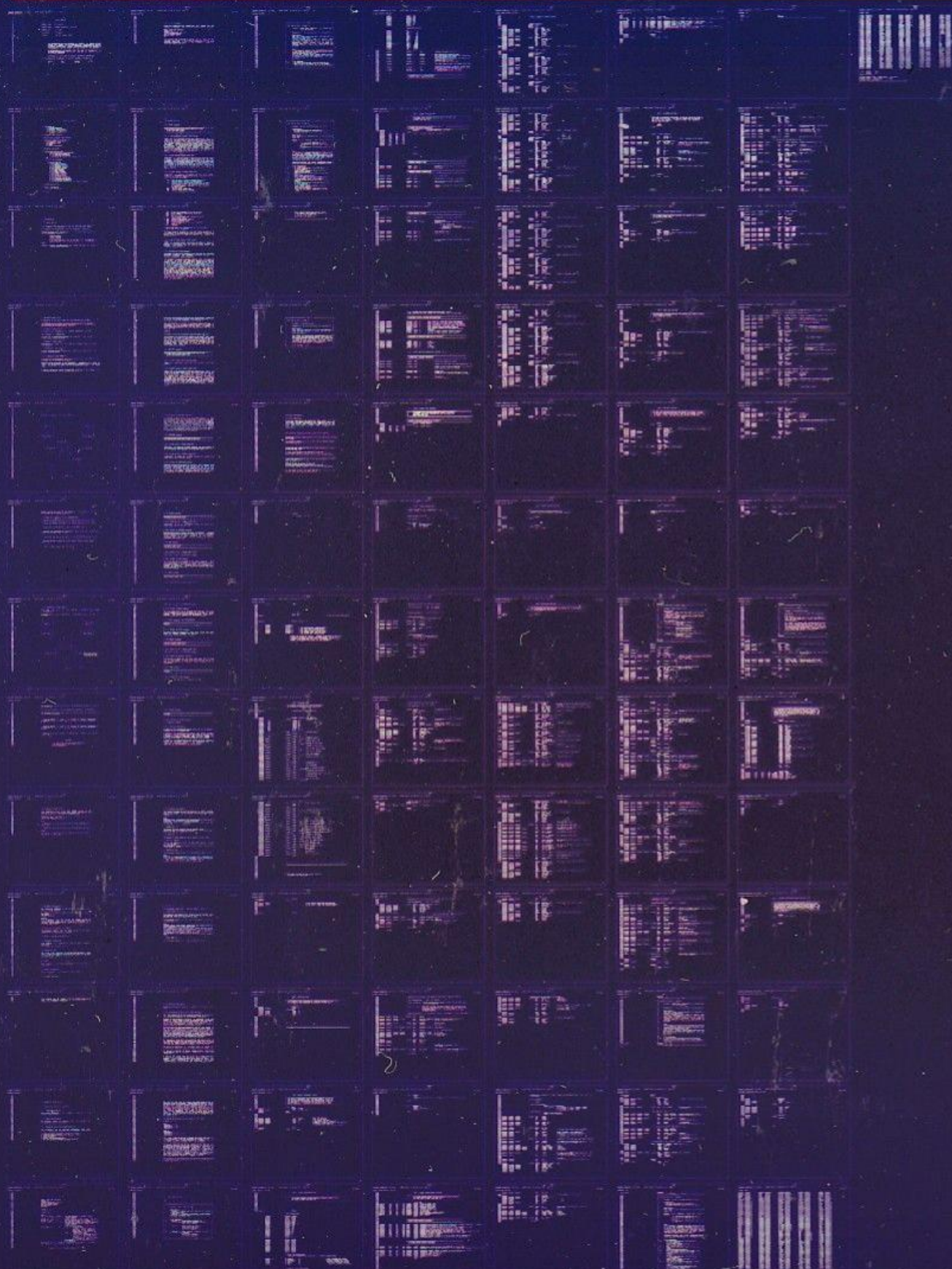


GS03WD

GS3WD/DZ11 LGC DIAG
CZDZGA0

COPYRIGHT (c) 1984
AH-T919A-MC
FICHE 01 OF 01

FEB 1985
digital
Made In USA



CZDZGAO GSSMD/DZ11 LGC DIAG
PROGRAM DOCUMENT

MACRO M1200 03-AUG-84 15:01 PAGE 2

.rem 8

IDENTIFICATION
.....

PRODUCT CODE: AC-T918A-MC
PRODUCT NAME: CZDZGAO GSSMD/DZ11 LGC DIAG
PRODUCT DATE: JULY 1984
MAINTAINER: CSS ANNECY
AUTHOR: Jean-Christophe PINASA

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

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

COPYRIGHT (C) 1979,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

PROGRAM DOCUMENT

Table of contents

43	
44	
45	
46	
47	
48	
49	
50	
51	
52	1.0 Introduction
53	1.1 Program abstract
54	1.2 Hardware description
55	1.3 Hardware configuration
56	1.4 Diagnostic description
57	
58	2.0 Hardware requirements
59	
60	3.0 Preliminary program requirements
61	
62	4.0 General program considerations
63	4.1 Diagnostic supervisor
64	4.2 Execution time
65	4.3 XXDP.
66	4.4 Memory management
67	
68	5.0 Program load media
69	
70	6.0 Operating instructions
71	6.1 Loading and starting procedures
72	6.1.1 Loading procedures
73	6.1.2 Starting procedures
74	6.1.3 Steps for quick and simple execution
75	
76	6.2 Initial dialogue
77	
78	6.3 Program options
79	6.3.1 START command
80	6.3.2 RESTART command
81	6.3.3 CONTINUE command
82	6.3.4 PROCEED command
83	6.3.5 ADD command
84	6.3.6 DROP command
85	6.3.7 PRINT command
86	6.3.8 DISPLAY command
87	6.3.9 FLAGS command
88	6.3.10 ZFLAGS command
89	6.3.11 Control characters
90	6.3.12 Hardware parameters
91	6.3.13 Software parameters
92	6.3.14 Extended discussion of p-table dialogue
93	
94	7.0 Tests descriptions
95	
96	8.0 Error information
97	8.1 Error reporting
98	
99	9.0 History

PROGRAM DOCUMENT

101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

1.0 Introduction

1.1 Program abstract

This diagnostic was designed to test the GS03-MD LOGIC MODULE.

The program was implemented using the Diagnostic Supervisor.

Through dialogue with the operator, it will allow modification of device parameters, such as :

- UNIBUS address ;
- vector address ;
- priority level ;
- # of lines connected out of the DZ11 into the GS03-MD ;
- operating mode (0 -> hardware test ; 1 -> installation

test).

WARNING : RUNNING THIS DIAGNOSTIC WILL CAUSE THE GS03 TO SWITCH
LINES BETWEEN COMPUTERS.

PROGRAM DOCUMENT

128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

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 : CZDZGAO GS3WD/DZ11 LGC DIAG

The filename is : ZDZGAO.BIN

It will run in stand alone without any operator inter ention, 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.

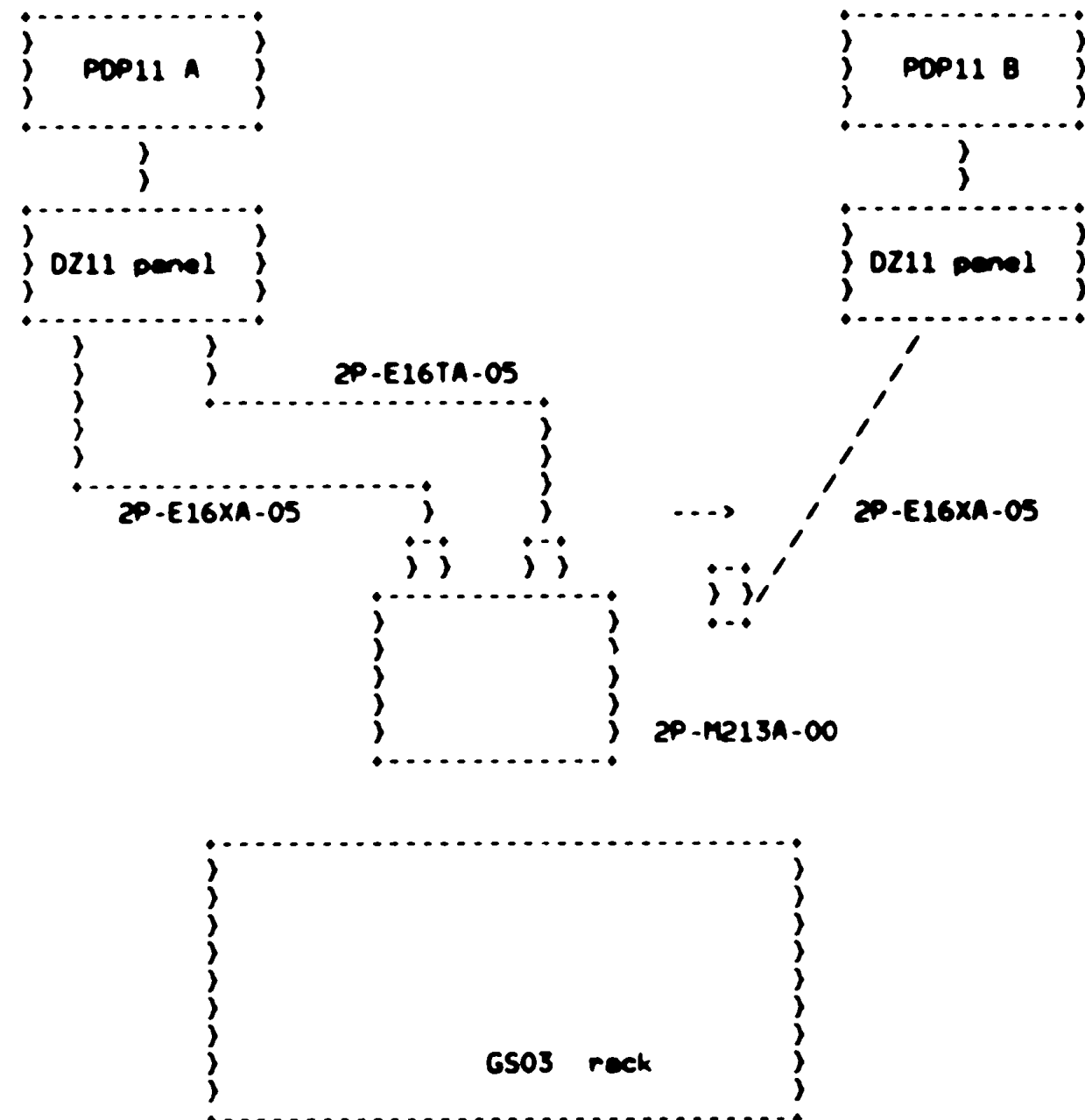
Before running this part of the diagnostic, operators will have to remove the 2P-E16XA-05 cable connected to the PDP11 that is not being used. (Disconnect the cable from the 2P-M213A-00 module in the GS03 rack).

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

174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225

Example: Diagnostic running on PDP11 A.



PROGRAM DOCUMENT

227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257

Before running this part of the diagnostic, the operator will have to go through the following checklist :

- Disable the highest priority commands by :
 - o Placing the FORCE AB switch to the center position on all racks
 - o Placing all MANUAL switches in the center position on all racks
- Remove the "2P-E16XA-05" on the unused PDP11 side.
- Connect the "2P-E16TA-05" diag test cable from the 2P-M213A-00 module to the PDP11 DZ11 in use.
- Check that the dip switch E18-1 on the 2P-M213A-00 module is off.
- Run the ZDZGA0 diagnostic on the chosen system (select mode 0). See 6.1.3.
- When finished, reconfigure the system.

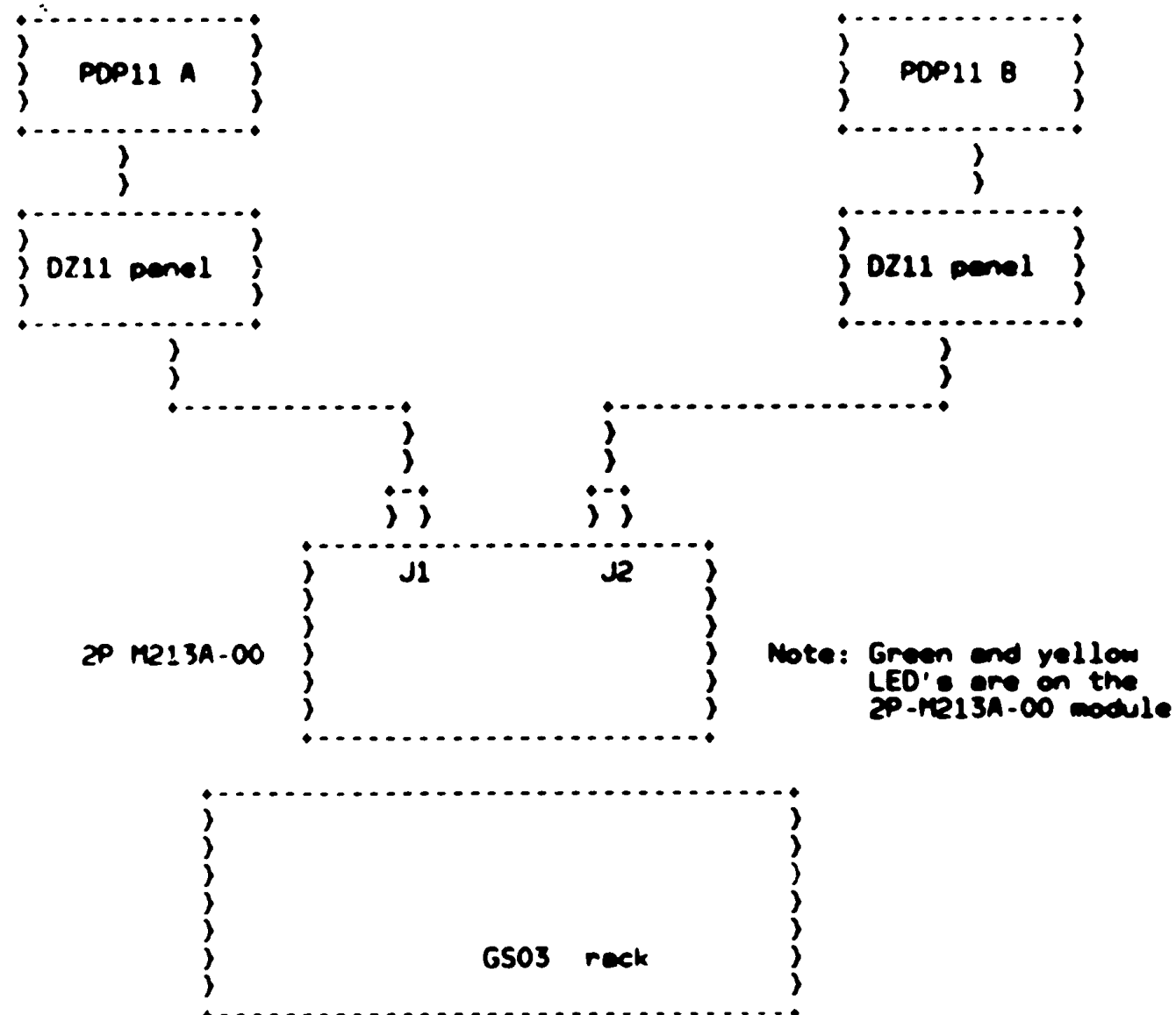
PROGRAM DOCUMENT

259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307

1.3.2 Installation test (MODE 1)

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

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



PROGRAM DOCUMENT

309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340

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-MD.

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 continuously and will have to be stopped by typing "cntrl C" on the console.

PROGRAM DOCUMENT

342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366

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.

PROGRAM DOCUMENT

368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424

2.0 Hardware requirements

The following hardware is required to run the static logic tests on module GS03-WD :

Any member of the PDP-11 UNIBUS family (PDP11/24, 34, 44, 70) ;
16k memory ;
console terminal.

WARNING :

This diagnostic will not run on any member of the VAX family, although a DZ11 may be fitted on a VAX UNIBUS. It is reminded that a GS03-WD logic module should be connected to a VAX through a DMF32.

3.0 Preliminary program requirements

The processor, memory and the DZ11 should be thoroughly tested prior to running this diagnostic.

4.0 General program considerations

4.1 Diagnostic Supervisor

This program is written to run under the PDP11 diagnostic supervisor.
It requires 16k of memory to run.

4.2 Execution Time

The total time required to run the GS03-WD static diagnostic ranges from about 2 minutes on the PDP11/70 to about 4 minutes on the PDP11/34 per pass for each unit (with supervisor version c4).

4.3 XXDP+

This program will be loaded under XXDP+, and may be run in dump mode.

4.4 Memory management

Memory management is not enabled by this program.

5.0 Program load media

PROGRAM DOCUMENT

425
426
427
428

This program can be loaded from any media supported by
XXDP+. The diagnostic supervisor will be loaded first, fol
lowed by the diagnostic program.

PROGRAM DOCUMENT

430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463

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

465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505

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 (0) 160100 ? 160340

VECTOR (0) 300 ? 460

BR (0) 5 ? 6

ACTIVE LINES (0) 3 ? <CR>

WHICH MODE (0) 0 ? <CR>

; The CSR address is 160340 (range =
; 160010-163776)

; Vector address is 460 (range = 300-777)

; BR interrupt level is 6 (range = 4-7)

; 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

; 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
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527

6.2 Initial dialogue

After the program and the supervisor are loaded and the program is started, the following identification is typed:

DIAG. RUN-TIME SERVICES
ZDZGAO-A-0
CZDZGAO GS3MD.DZ11 LGC DIAG
UNIT IS GS03MD MODULE
RESTART ADDR: 147670
DR>

The operator then proceeds by typing one or more of the commands described in the following section 6.3. (for more detailed information, refer to the diagnostic supervisor functional specification).

PROGRAM DOCUMENT

529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585

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 separated 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 occurrence 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>, separated by colons, where <FLAG> has one of the following values:

MOE	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

PROGRAM DOCUMENT

586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634

- * PRI direct all messages to a line printer
- PNT print number of test being executed
- BOE bell on error
- ** UAM run in unattended mode, bypassing manual intervention tests
- ** ISR inhibit statistical reports
- ADR execute autodrop code
- IDU inhibit dropping of units by diagnostic
- LOT loop on test
- ** EVL evaluate

- * NOT TO BE USED if a line printer is not available.
- ** Of no use in this diagnostic.

The flags named or equated to 1 are set, those equated to 0 are cleared. A flag not specified is cleared. If the flags switch is not given all flags are cleared. See example at end of 6.3.1.5.

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

<INCR> is a decimal number indicating how often (in terms of passes) it is desired that the end of pass message be printed. The default is at the end of every pass. See example at end of 6.3.1.5.

6.3.1.5 Effect of a start command

The effect of the start command is to initiate the hardware parameter dialogue, the software parameter dialogue, and then the diagnostic tests themselves.

The hardware parameter dialogue starts with the question "0 units?" to which the operator replies with a decimal number n from 1 to 16. The term "unit" refers to the device to which this series of diagnostics is dedicated. Following this are the questions whereby the p-tables themselves will be built. Each p-table is a core-resident table containing all the hardware information for one unit. The operator must supply n (number of units) values for each question. He may do this by giving one answer to each question (in which case the series of questions will be posed n times) or by giving n values, separated by commas, to each question (series will be posed once). Each question is followed by the response radix (d for decimal, b for binary, o for octal, l for yes/no) in parentheses and the default value after the parentheses.

PROGRAM DOCUMENT

636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690

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.

PROGRAM DOCUMENT

692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738

6.3.2.3 Effect of RESTART command

The RESTART command differs from the START command in that the p-tables from the previous start command (there must have been one) are used, instead of new ones being built. The units switch gives the ability to select a subset of these. The software dialogue may optionally be reexecuted (operator will be asked). The command can be used after command mode has been reentered in any of the three normal ways: a) the requested number of passes have been made b) an error was encountered with the halt on error flag set c) a "control/c" was entered by the operator.

6.3.3 CONTINUE command

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

6.3.3.1 PASS switch (/PASS:<PASS-CNT>)

<PASS-CNT> is same as in START command, but the default is the unsatisfied pass-cnt from the previous START or RESTART. If none remains, the default is non-ending execution.

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

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

6.3.3.3 Effect of CONTINUE command

CONTINUE must follow a start or restart, and command mode must have been entered due to a halt on error or a control/c. The effect of the command is to go to the beginning of the test that was being executed when the halt or control/c took place. Software dialogue may optionally be reexecuted. Hardware parameters may not be changed.

PROGRAM DOCUMENT

740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796

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>)

PROGRAM DOCUMENT

797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851

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

6.3.6.2 Effect of DROP command

The units specified will be dropped from testing. The units will be reselected only by the execution of an ADD or START command. The units switch must be entered. This command must be followed by a RESTART or a CONTINUE command.

6.3.7 PRINT command : NOT IMPLEMENTED

PRI(NT)

6.3.7.1 Effect of PRINT command

The total number of errors for each unit since the last start or restart command are printed. The ier (inhibit statistical reporting) flag is cleared.

6.3.8 DISPLAY command

DIS(PLAY)/UNITS;<UNIT-LIST>

6.3.8.1 UNITS switch (/UNITS:<UNIT-LIST>)

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

6.3.8.2 Effect of DISPLAY command

The hardware p-tables for all units under test are printed out in the format in which they were entered. Any units that were dropped by the operator "drop" command are so designated.

6.3.9 FLAGS command

FLA(GS)

6.3.9.1 Effect of FLAGS command

The current settings of all flags are printed.

PROGRAM DOCUMENT

853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883

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 suppressed for the remainder of the diagnostic or until another control o is typed, which restores normal teletype output.

PROGRAM DOCUMENT

885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928

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 NNA" 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.

PROGRAM DOCUMENT

930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960

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 needed 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.

PROGRAM DOCUMENT

962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009

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).

PROGRAM DOCUMENT

1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059

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 (0) ? 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 explicit 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
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094

7.0 Tests Descriptions

```
***** TEST 1 *****
*
*   Purpose : basic test of DZ11.
*
*   Description :
*   - Subtest 1 : Check that DZ11 CSR can be written
*   to and read from ;
*   - Subtest 2 : Transmit a character in maintenance
*   (internal) loopback mode on the selected line(s)
*   and check for proper echo.
*
*   Error messages :
* #0.1 - Subtest 1 : "Unsuccessful attempt to
*   write to/read DZ11 CSR at address <address>"
*   "Check DZ11 address."
*
* #2   - Subtest 2 : "DZ11 failed to reset."
*   "Check DZ11 address."
*   "Run DZ11 diagnostic."
*
* #3   - Subtest 2 : "DZ11 internal loopback malfunction
*   on line # <line number>"
*   "Check DZ11 address."
*   "Run DZ11 diagnostic."
*
*****
```

PROGRAM DOCUMENT

1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134

```

***** TEST 2 *****
*
* Test active only in mode 0 :
*
*   Purpose : check that characters are echoed back
*             from the GS03-MD.
*
*   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-MD 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-MD switches
*   to one line by receiving echoed characters back
*   from the GS03-MD 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-MD, on at least
*   one line.
*
*   Error message :
*   - "No echo received back from the GS03-MD on
*     either line # <line number> or # <line number>"
*     "Check cabling and dip switch E18 (must be OFF)"
*
*****

```

PROGRAM DOCUMENT

1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192

```
***** TEST 3 *****  
*  
* Test active only in mode 0 :  
*  
*   Purpose : switch the GS03-MD back and forth.  
*  
*   Assumptions :  
*   - all previous tests ran successfully ;  
*   - WATCHDOG FUNCTION has priority (cf. note).  
*  
*   Description :  
*   This test is the implementation of the following  
*   algorithm :  
*  
*   Repeat twice, swapping lines x and y, the se-  
*   quence :  
*       - Try and switch GS03-MD to line x ;  
*       - Try and switch GS03-MD from line x to  
*         line y ;  
*       - Try and switch GS03-MD back from line y  
*         to line x ;  
*  
*   Note :  
*   This diagnostic assumes that the switches are set  
*   to give the WATCHDOG FUNCTION priority. This  
*   means that the front panel switches should all be  
*   in the center position and the relay modules  
*   should all be configured for the same priority  
*   (see Option Description for details).  
*  
*   IMPROPER SETTINGS CAN CAUSE UNEXPECTED ERRORS,  
*   WHICH WILL NOT NECESSARILY BE DIAGNOSED AS SUCH.  
*  
*   Error messages :  
*  
* #5 - "No echo received back from GS03-MD on line  
*       # <line number>"  
*       "Check FORCE, MANUAL switches, priority  
*       setting and cables".  
* #6 - "Both lines have switch priority over each  
*       other."  
*       "Check GS03 configuration."  
* #7 - "Echo from GS03-MD received on both lines  
*       # <line number> and # <line number>."  
* #8 - "GS03-MD failed to switch to line # <line  
*       number>"  
*       "No echo received back from GS03-MD on line  
*       # <line number>"  
*       "Check FORCE, MANUAL switches, priority  
*       setting and cables".  
* #9 - "Echo from the GS03-MD received on wrong  
*       line # <line number> (expected : # <line  
*       number>)."
```

PROGRAM DOCUMENT

1193
1194
1195
1196
1197
1198

```

•      "Echo is still being received on line #      •
•      <line number> when actually transmitting      •
•      on line # <line number> only."                •
•      "Check GS03 configuration."                    •
•      .....

```

PROGRAM DOCUMENT

1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227

```
***** TEST 4 *****  
*  
* Test active only in mode 1 :  
*  
*   Purpose : installation test.  
*  
*   Assumption : all previous tests ran successfully.  
*  
*   Description :  
*   This test activates the line into the GS03-MD in  
*   order for the operator to check that the LED's  
*   react correctly :  
*  
*   The GREEN or YELLOW LED corresponding to this  
*   CPU's line into the GS03-MD should then turn on.  
*   The associated RED LED should turn off after one  
*   full GS03-MD clock pulse after this test begins  
*   (which means that the RED clock LED should blink  
*   twice at the most before this happens).  
*  
*   Error message : none.  
*  
*****
```

PROGRAM DOCUMENT

1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275

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:

ZDZGAO 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.

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

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

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

GS03-MD failed to switch to line # 2
No echo received back from GS03-MD 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
1283
1284
1285
1286
1287
1288
1289

9.0 History
- first release : JULY 84
8

PROGRAM DOCUMENT

```
1299          002000          .-2000
1300
1301
1302
1303
1304
1305
1306          .MCALL  SVC
1307 002000          SVC          ; INITIALIZE SUPERVISOR MACROS
1308
1309
1310
1311 002000          BGNMOD  ZDZGAO
1312
1313
1314          000000          #LSTIN= 0
1315          000000          #LSTTAG= 0
1316          000000          SVCINS= 0          ; LIST INSTRUCTIONS, SHIFTED RIGHT
1317          000000          SVCTST= 0         ; LIST TEST TAGS, SHIFTED RIGHT
1318          000000          SVCSUB= 0         ; LIST SUBTEST TAGS, SHIFTED RIGHT
1319          000000          SVCGBL= 0         ; LIST GLOBAL TAGS, SHIFTED RIGHT
1320          000000          SVCTAG= 0         ; LIST OTHER TAGS, SHIFTED RIGHT
1321
1322          ;      CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1323          ;      TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1324          ;      SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1325          ;      CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1326
1327
```

PROGRAM HEADER

1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1357
1358

002000

002000

002000

002000 132

002001 104

002002 132

002003 107

002004 101

002005 060

002006 000

002007 000

002010

002010 101

002011

002011 060

002012

002012 000001

002014

002014 000416

002016

002016 011562

002020

002020 000000

002022

002022 002144

002024

002024 000000

002026

002026 012076

002030

002030 000000

002032

002032 000000

002034

002034 000000

002036

002036 000000

002040

002040 002132

002042

002042 000000

002044

002044 000000

002046

002046 000000

002050

002050 003

.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 /O/
               .BYTE  0
               .BYTE  0
L#REV::        ;REVISION LEVEL
               .ASCII /A/
L#DEPO::       ;0
               .ASCII /O/
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#HREV::       ;SVC REV AND EDIT #
               .BYTE   C#REVISION
```

PROGRAM HEADER

002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 002324
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 007730
002072
002072 007650
002074
002074 000000
002076
002076 002156
002100
002100 104035
002102
002102 000000
002104
002104 005774
002106
002106 007636
002110
002110 007544
002112
002112 002122
002114
002114 000000
002116
002116 000000
002120
002120 000000

.BYTE CREDIT
L\$EF:: .WORD 0 ;DIAG. EVENT FLAGS
 .WORD 0
L\$SPC:: .WORD 0
L\$DEVP:: .WORD L\$DVTYP ; POINTER TO DEVICE TYPE LIST
L\$REPP:: .WORD 0 ;PTR. TO REPORT CODE
L\$EXP4:: .WORD 0
L\$EXP5:: .WORD 0
L\$AUT:: .WORD L\$AU ;PTR. TO ADD UNIT CODE
L\$DUT:: .WORD L\$DU ;PTR. TO DROP UNIT CODE
L\$LUN:: .WORD 0 ;LUN FOR EXERCISERS TO FILL
L\$DESP:: .WORD L\$DESC ;POINTER TO DIAG. DESCRIPTION
L\$LOAD:: .WORD L\$DESC ;GENERATE SPECIAL AUTOLOAD EMT
 EMT E\$LOAD
L\$ETP:: .WORD 0 ;POINTER TO ERRTAB
L\$ICP:: .WORD L\$INIT ;PTR. TO INIT CODE
L\$CCP:: .WORD L\$CLEAN ;PTR. TO CLEAN-UP CODE
L\$ACP:: .WORD L\$AUTO ;PTR. TO AUTO CODE
L\$PRT:: .WORD L\$PROT ;PTR. TO PROTECT TABLE
L\$TEST:: .WORD 0 ;TEST NUMBER
L\$DLY:: .WORD 0 ;DELAY COUNT
L\$HIME:: .WORD 0 ;PTR. TO HIGH MEM

1359
1360
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385

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

PROGRAM HEADER

1386 002122
002122BGNPROT
L1PROT::

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

DISPATCH TABLE

1412
1413
1414
1415
1416
1417
1418
1419 002130
002130 000004
002132
002132 010006
002134 010616
002136 010750
002140 011504

1420
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441

.SBTTL DISPATCH TABLE

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

DISPATCH 4

.WORD 4

L#DISPATCH::

.WORD T1

.WORD T2

.WORD T3

.WORD T4

;*****

DEFAULT HARDWARE P-TABLE

.SBTTL DEFAULT HARDWARE P-TABLE

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454 002142

002142 000005

002144

002144

1455

1465

1466

1467 002144 160100

1468 002146 000300

1469 002150 000005

1470 002152 000003

1471 002154 000000

1472

1473

1474 002156

002156

1475

```

////////////////////////////////////
; 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
      BGNHW DFPTBL
      .WORD L10001-L$HW/2
L$HW::
DFPTBL::

```

```

.word 160100 ; DZ11 CSR address
.word 300 ; DZ11 vector address
.word 5 ; interrupt priority level (5)
.word 3 ; bitmap of lines out of DZ11 into GS03-WD
.word 0 ; diagnostic test mode (0)
; or installation test mode (1) selector

```

```

      ENDHW
L10001:

```

GLOBAL EQUATES SECTION

1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1497
1498
1513
1514 002156

.SBTTL GLOBAL EQUATES SECTION

```

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

```

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      HDE== 100000

```

```

1515      000340      MAXPRI ==      340      ; Highest processor priority : 7
1516
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 ; STORAGE FOR DEVICE REGISTERS
1555 ;*****
1556 DESCRIPT <CZDZGAO GSSMD.DZ11 LGC DIAG>
1557 LIDESC::
1558 002156 .ASCIZ /CZDZGAO GSSMD.DZ11 LGC DIAG/
1559 002156 103 132 104
1560 002161 132 107 101
1561 002164 060 040 107
1562 002167 123 063 127
1563 002172 104 056 104
1564 002175 132 061 061
1565 002200 040 114 107
1566 002203 103 040 104
1567 002206 111 101 107
1568 002211 000
1569
1570 .EVEN
1571
1572 ; ERRTAB
1573
1574 ;*****
1575 ; PROGRAM CONTROL FLAGS
1576 ;*****
1577 FTIME: .word 0 ; boolean to record first initialization
1578 THODE: .word 0 ; Test mode
1579
1580 ;*****
1581 ; PROGRAM CONTROL PARAMETERS
1582 ;*****
1583 UUT: .word 0 ; Unit under test
1584 SWPTY: .word 0 ; Switch priority (line x, y or none)
1585 MAXERR: .word 5 ; max error count before dropping unit
1586 ERRCNT: .word 0 ; error count
1587
1588 SAVE4: .word 0 ; temporary storage for timeout trap
1589 SAVE6: .word 0 ; vector
1590
1591 002216 000000
1592 002220 000000
1593 002222 000005
1594 002224 000000
1595 002226 000000
1596 002230 000000
1597
1600
```

GLOBAL DATA SECTION

```

1602 ; .....
1603 ; * MISCELLANEOUS STORAGE
1604 ; .....
1605
1606 002232 000000 TXPSW: .word 0 ; transmitter interrupt vector PSW
1607
1608 002234 000000 DZPTY: .word 0 ; DZ11 priority
1609 002236 011070 TLPRO: .word 011070 ; predefined parameter description for lines into the
1610 ; GS03-MD :
1611 ; - 8 bit characters
1612 ; - 1 start bit, 2 stop bits
1613 ; - 110 bauds or 1 character every 100 ms
1614 ; - receiver on
1615
1616 002240 000000 TLPRx: .word 0 ; test LPR for line x
1617 002242 000000 TLPRy: .word 0 ; test LPR for line y
1618
1619 002244 000000 DLAYC1: .word 0
1620 002246 000000 DLAYC2: .word 0
1621
1622 002250 000000 ECHO: .WORD 0 ; to store RBUF contents echoed back through
1623 ; the DZ11
1624
1625
1626

```

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 PSH address
1638 002256 000000 DZtVCCa: .word 0      ; DZ11 transmitter interrupt vector PC address
1639 002260 000000 DZtVCSa: .word 0      ; DZ11 transmitter interrupt vector PSH address
1640 002262 000000 DZCSRa: .word 0      ; DZ11 control status register address
1641 002264 000000 DZRBUFFa: .word 0      ; 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 LNNAP: .word 0      ; Line
1654 002302 000000 LNNAPx: .word 0      ; bitmaps
1655 002304 000000 LNNAPy: .word 0      ; (0..377)
1656
1657
1658      ;***** STACK USED FOR SUBROUTINE LINKAGE *****
1659
1660
1661      ;*****
1662      ;* SUBROUTINE LINKAGE PARAMETERS
1663      ;*****
1664
1665 002306 000000 sbACK: .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 NEULNMP: .WORD 0      ; Bitmap and
1673 002316 000000 NEULNNB: .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

1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693

1694
1695
1696
1703
1704
1705

.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  
; *****  
DEVTP <GS03MD MODULE>  
L#DVTP::  
    .ASCIZ /GS03MD MODULE/  
  
    .EVEN
```

002324			
002324	107	123	060
002327	063	127	104
002332	040	115	117
002335	104	125	114
002340	105	000	

GLOBAL SUBROUTINES

1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726

.SBTTL GLOBAL SUBROUTINES

; MACRO'S NEEDED TO CALL SUBROUTINES

; macro to wait a few ms

; Call sequence : DLAY D 0 < D < 177777

.macro DLAY D
 MOV @176630, DLAYC1
 MOV @D, DLAYC2
 JSR PC, @bWTG2
.endm

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
1760      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      CHKMAX: INLOOP      ; If looping on error
1778      002400 104420      TRAP   C#INLP
1779      002402 103432      BCOMPLETE 1#      ; then exit
1780      002404      BCS   1#
1781      002404 104421      RFLAGS RO
1782      002406 032700 000040      TRAP   C#RFLA
1783      002412 001026      BIT    #IDU, RO      ; If dropping of units is inhibited
1784      002414 005237 002224      BNE    1#      ; then exit
1785      002420 023737 002224 002222      INC    ERRCNT      ; Update error count
1786      002426 003420      CMP    ERRCNT, MAXERR      ; If there aren't too many errors
1787      002430      BLE    1#      ; then exit
1788      002430      PRINTF  #TMNYSR, MAXERR, UUT
1789      002430 013746 002216      MOV    UUT, -(SP)
1790      002434 013746 002222      MOV    MAXERR, -(SP)
1791      002440 012746 002472      MOV    #TMNYSR, -(SP)
1792      002444 012746 000003      MOV    #3, -(SP)
1793      002450 010600      MOV    SP, RO
1794      002452 104417      TRAP   C#PNTF
1795      002454 062706 000010      ADD    #10, SP
1796
1797      ; else print 'Maximum error count
1798      ; of <maxerr> exceeded for unit <UUT>'
1799      ; and drop unit
1800
1801      DODU    UUT
1802      MOV    UUT, RO
1803      TRAP   C#DODU
1804
1805      DOCLN
1806      TRAP   C#DCLN      ; Abort subpass
1807
1808      1#:     RTS    PC
1809
1810
1811      .nlist
1812      002472 045 116 045 TMNYSR: .ASCII
1813      .list
1814      .EVEN
1815      BEX
1816      /#D3#A Maximum error count of #D3#A exceeded for unit #D2/
1817      BEX

```

GLOBAL SUBROUTINES

```
1808      ; service routine to transmit in interrupt mode :
1809 002562      BGNSRV  svTXG1
          002562      svTXG1::
1810 002562 112777 000101 177500      MOVB  @'A, BDZTDR@  ;
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      sbIDG1: SETVEC  DZTVCCa, @vTXG1, TXPSW
1821      MOV          TXPSW, -(SP)
1822      MOV          @vTXG1, -(SP)
1823      MOV          DZTVCCa, -(SP)
1824      MOV          @3, -(SP)
1825      TRAP         C$SVEC
1826      ADD          @10, SP
1827      ; Set up transmitter interrupt vector
1828      MOV          @CSRCLR, @DZCSRa; Set CLR bit of DZ11 CSR
1829      nCLDG1: BIT    @CSRCLR, @DZCSRa; Test CLR
1830      BREAK        ; Authorize "control-C" abort
1831      TRAP         C$BRK
1832      BNE          nCLDG1      ; Wait until CSRCLR = 0
1833      RTS          PC
1834      ; end sbIDG1
```

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

GLOBAL ERROR REPORT SECTION

.SBTTL GLOBAL ERROR REPORT SECTION

1898
1899
1900
1901
1902
1903
1904
1905
1906 002754
1907 002777
1908 003021
1909 003044
1910 003066
1911 003112
1912 003125
1913 003162
1914 003214
1915
1916
1917
1918
1919
1920
1921 003230
1922 003343
1923 003416
1924 003515
1925 003602
1926 003667
1927 003776
1928 004064
1929 004116
1930 004213
1931
1932
1933
1934
1935
1936
1937 004304
1938 004371
1939 004523
1940 004622
1941 004706
1942 004744
1943 004775
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953

```

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

```

```

.nlist BEX
110 FAISWF: .ASCIZ /ECHO ON WRONG LINE/
111 FAISWT: .ASCIZ /FAIL TO SWITCH TO/
110 WDZECH: .ASCIZ /ECHO ON BOTH LINES/
111 PTYCFL: .ASCIZ /PRIORITY CONFLICT/
040 NO1LEC: .ASCIZ /NO ECHO ON ONE LINE/
040 NOMDEC: .ASCIZ /NO WD ECHO/
061 DZLBER: .ASCIZ /DZ11 INTERNAL LOOPBACK ERROR/
061 DZINER: .ASCIZ /DZ11 INITIALIZATION FAULT/
123 BUSTIM: .ASCIZ /BUS TIMEOUT/
.EVEN

```

```

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

```

```

045 FSWF: .ASCIZ /%A%Echo from GS03-WD received on wrong line # %D1%A (expected : # %D1%A)./
045 FSWT: .ASCIZ /%A%GS03-WD failed to switch to line # %D1/
045 WDZE: .ASCIZ /%A%Echo from GS03-WD received on both lines # %D1%A and # %D1/
045 PYCF: .ASCIZ /%A%Both lines have switch priority over each other./
045 N1LE: .ASCIZ /%A%No echo received back from GS03-WD on line # %D1/
045 NMDE: .ASCIZ /%A%No echo received back from GS03-WD on either line # %D1%A or # %D1/
045 DZLB: .ASCIZ /%A%DZ11 internal loop back malfunction on line # %D1/
045 DZIN: .ASCIZ /%A%DZ11 failed to reset./
045 CSRw: .ASCIZ /%A%Unsuccessful attempt to write to DZ11 CSR at address %06/
045 CSRr: .ASCIZ /%A%Unsuccessful attempt to read DZ11 CSR at address %06/
.EVEN

```

```

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

```

```

045 NOEC1L: .ASCIZ /%A%No echo received back from GS03-WD on line # %D1/
045 STEC1L: .ASCIZ /%A%Echo is still being received on line # %D1%A when actually transmitting
045 CKFMSW: .ASCIZ /%A%Check FORCE, MANUAL switches, priority setting and cables./
045 CKDPSW: .ASCIZ /%A%Check cabling and dip switch E18 (must be OFF)./
045 CKGSCF: .ASCIZ /%A%Check GS03 configuration./
045 DZDIAG: .ASCIZ /%A%Run DZ11 diagnostic./
045 CKDZAD: .ASCIZ /%A%Check DZ11 address./
.nlist BEX
.EVEN

```

GLOBAL ERROR REPORT SECTION

```

1956
1957
1958
1959
1960
1961
1962
1963 005026
      005026
1964 005026      013746      002320
      005032      012746      004213
      005036      012746      000002
      005042      010600
      005044      104414
      005046      062706      000006
1965 005052
      005052      012746      004775
      005056      012746      000001
      005062      010600
      005064      104415
      005066      062706      000004
1966 005072      004737      002400
1967 005076
      005076
      005076      104423
1968
1969
1970 005100
      005100
1971 005100
      005100      013746      002320
      005104      012746      004116
      005110      012746      000002
      005114      010600
      005116      104414
      005120      062706      000006
1972 005124
      005124      012746      004775
      005130      012746      000001
      005134      010600
      005136      104415
      005140      062706      000004
1973 005144      004737      002400
1974 005150
      005150
      005150      104423
1975
1976
1977 005152
      005152
1978 005152
      005152      012746      004064
      005156      012746      000001
      005162      010600
      005164      104414
      005166      062706      000004

```

```

.....
; MACRO'S NEEDED TO REPORT ERRORS
.....

; Error # 0 report
BGNMSG  @CSRr
pCSRr::
PRINTB  @CSRr, ADDR
        MOV     ADDR, -(SP)
        MOV     @CSRr, -(SP)
        MOV     @2, -(SP)
        MOV     SP, R0
        TRAP    C@PNTB
        ADD     @6, SP
PRINTX  @CKDZAD
        MOV     @CKDZAD, -(SP)
        MOV     @1, -(SP)
        MOV     SP, R0
        TRAP    C@PNTX
        ADD     @4, SP
        JSR     PC, CHKMAX      ; check if too many errors
ENDMSG
L10003:
        TRAP    C@MSG

; Error # 1 report
BGNMSG  @CSRw
pCSRw::
PRINTB  @CSRw, ADDR
        MOV     ADDR, -(SP)
        MOV     @CSRw, -(SP)
        MOV     @2, -(SP)
        MOV     SP, R0
        TRAP    C@PNTB
        ADD     @6, SP
PRINTX  @CKDZAD
        MOV     @CKDZAD, -(SP)
        MOV     @1, -(SP)
        MOV     SP, R0
        TRAP    C@PNTX
        ADD     @4, SP
        JSR     PC, CHKMAX      ; check if too many errors
ENDMSG
L10004:
        TRAP    C@MSG

; Error # 2 report
BGNMSG  @DZIN
pDZIN::
PRINTB  @DZIN
        MOV     @DZIN, -(SP)
        MOV     @1, -(SP)
        MOV     SP, R0
        TRAP    C@PNTB
        ADD     @4, SP

```

GLOBAL ERROR REPORT SECTION

```

1979 005172
      005172 012746 004775
      005176 012746 000001
      005202 010600
      005204 104415
      005206 062706 000004
1980 005212
      005212 012746 004744
      005216 012746 000001
      005222 010600
      005224 104415
      005226 062706 000004
1981 005232 004737 002400
1982 005236
      005236
      005236 104423
1983
1984
1985 005240
      005240
1986 005240
      005240 013746 002272
      005244 012746 003776
      005250 012746 000002
      005254 010600
      005256 104414
      005260 062706 000006
1987 005264
      005264 012746 004775
      005270 012746 000001
      005274 010600
      005276 104415
      005300 062706 000004
1988 005304
      005304 012746 004744
      005310 012746 000001
      005314 010600
      005316 104415
      005320 062706 000004
1989 005324 004737 002400
1990 005330
      005330
      005330 104423
1991
1992
1993 005332
      005332
1994 005332
      005332 013746 002276
      005336 013746 002274
      005342 012746 003667
      005346 012746 000003
      005352 010600
      005354 104414
      005356 062706 000010
1995 005362
      005362 012746 004622

```

```

PRINTX @CKDZAD
      MOV @CKDZAD, -(SP)
      MOV @1, -(SP)
      MOV SP, R0
      TRAP C@PNTX
      ADD @4, SP
PRINTX @DZDIAG
      MOV @DZDIAG, -(SP)
      MOV @1, -(SP)
      MOV SP, R0
      TRAP C@PNTX
      ADD @4, SP
      JSR PC, CHKMAX ; check if too many errors
ENDMSG
L10005:
      TRAP C@MSG
; Error @ 3 report
BGNMSG @DZLB ; DZ11 loopback error
pDZLB::
PRINTB @DZLB, L@NBR
      MOV L@NBR, -(SP)
      MOV @DZLB, -(SP)
      MOV @2, -(SP)
      MOV SP, R0
      TRAP C@PNTB
      ADD @6, SP
PRINTX @CKDZAD
      MOV @CKDZAD, -(SP)
      MOV @1, -(SP)
      MOV SP, R0
      TRAP C@PNTX
      ADD @4, SP
PRINTX @DZDIAG
      MOV @DZDIAG, -(SP)
      MOV @1, -(SP)
      MOV SP, R0
      TRAP C@PNTX
      ADD @4, SP
      JSR PC, CHKMAX ; check if too many errors
ENDMSG
L10006:
      TRAP C@MSG
; Error @ 4 report
BGNMSG @NMDE ; MD fail to echo error
pNMDE::
PRINTB @NMDE, L@NBRx, L@NBRy
      MOV L@NBRy, -(SP)
      MOV L@NBRx, -(SP)
      MOV @NMDE, -(SP)
      MOV @3, -(SP)
      MOV SP, R0
      TRAP C@PNTB
      ADD @10, SP
PRINTX @CKDPSW
      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					
2000 005410			; Error # 5 report		
005410			BGNMSG	#NILE	; No echo received on line x error
2001 005410			#NILE::		
005410			PRINTB	#NILE, LNNBRx	
005410	013746	002274	MOV	LNNBRx, -(SP)	
005414	012746	003602	MOV	#NILE, -(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					
2007 005462			; Error # 6 report		
005462			BGNMSG	#PYCF	; Both lines have priority error
2008 005462			#PYCF::		
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					
2014 005530			; Error # 7 report		
005530			BGNMSG	#MD2E	; Echo on both lines error
2015 005530			#MD2E::		
005530			PRINTB	#MD2E, LNNBRx, LNNBRy	
005530	013746	002276	MOV	LNNBRy, -(SP)	
005534	013746	002274	MOV	LNNBRx, -(SP)	
005540	012746	003416	MOV	#MD2E, -(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	#FSW		; MD fail to switch to line i error
	005566			#FSW::			
2021	005566			PRINTB	#FSW, NEWLNMB		
	005566	013746	002316		MOV	NEWLNB, -(SP)	
	005572	012746	003343		MOV	#FSW, -(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	#DEC1L, NEWLNMB		
	005612	013746	002316		MOV	NEWLNB, -(SP)	
	005616	012746	004304		MOV	#DEC1L, -(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	#CKFSW		
	005636	012746	004523		MOV	#CKFSW, -(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	#FSWF		; MD echo on wrong line error
	005664			#FSWF::			
2029	005664			PRINTB	#FSWF, OLDLNMB, NEWLNMB		
	005664	013746	002316		MOV	NEWLNB, -(SP)	
	005670	013746	002312		MOV	OLDLNMB, -(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, OLDLNMB, NEWLNMB		
	005714	013746	002316		MOV	NEWLNB, -(SP)	
	005720	013746	002312		MOV	OLDLNMB, -(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		

GLOBAL ERROR REPORT SECTION

	005744	012746	004706	MOV	#CKGSCF, -(SP)	
	005750	012746	000001	MOV	#1, -(SP)	
	005754	010600		MOV	SP, R0	
	005756	104415		TRAP	CIPNTX	
	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	CIMSG	
2034						
2035						
2036						
2037						
2038						

REPORT CODING SECTION

.SBTTL REPORT CODING SECTION

2041

2042

2043

2044

2045

2046

2047

2048

2049 005772

005772

2050

2056

2057

2064

2065 005772

005772

005772 104425

2066

2067

; **

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

; --

BGNRPT

L1RPT::

ENDRPT

L10015:

TRAP C1RPT

INITIALIZE SECTION

2070
2071
2072
2073
2074
2075
2076
2077
2078
2079

005774
005774

```
.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          READDEF @EF.START          ; "START",
2115 006040 012700 000040          MOV     @EF.START,RO
2116 006044 104447          TRAP     C$REFG
2117 006046          BCOMPLETE      FtUNI1 ;
2118 006046 103422          BCS      FtUNI1
2119
2120 006050          READDEF @EF.RESTART        ; "RESTART" commands,
2121 006050 012700 000037          MOV     @EF.RESTART,RO
2122 006054 104447          TRAP     C$REFG
2123 006056          BCOMPLETE      FtUNI1 ;
2124 006056 103416          BCS      FtUNI1
2125
2126 006060          READDEF @EF.PWR            ; or POWER UP :
2127 006060 012700 000034          MOV     @EF.PWR,RO
2128 006064 104447          TRAP     C$REFG
2129 006066          BCOMPLETE      FtUNI1 ; start with first unit (@ 0)
2130 006066 103412          BCS      FtUNI1
2131
2132 006070          READDEF @EF.CONTINUE       ; If this a "continue" command,
2133 006070 012700 000036          MOV     @EF.CONTINUE,RO
2134 006074 104447          TRAP     C$REFG
2135 006076          BNCUMPLETE      contI1 ; then exit
2136 006076 103002          BCC      contI1
2137 006100          EXIT      INIT            ; (no re-initialization)
2138 006100 104432          TRAP     C$EXIT
2139 006102 001440          .WORD     L10016-.
2140
2141          contI1: READDEF @EF.NEW            ; If this is not a new pass,
2142 006104 012700 000035          MOV     @EF.NEW,RO
2143 006110 104447          TRAP     C$REFG
2144 006112          BNCUMPLETE      nxUNI1 ; then get next unit
2145 006112 103003          BCC      nxUNI1
2146
2147 006114 012737 177777 002216 FtUNI1: MOV     @-1, UUT      ;
2148
2149 006122 005237 002216          nxUNI1: INC     UUT          ; Point to next unit
2150 006126 023737 002216 002012    CMP     UUT, L$UNIT      ; If there isn't any,
2151 006134 002161          BGE     aborI1          ; then end-of-pass
2152

```

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    C1GPHRD
      006144 010001         MOV    R0, R1
2140 006146 BCOMPLETE gtPHI1      ;
      006146 103413         BCS    gtPHI1

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    C1PNTF
      006170 062706 000006   ADD    #6, SP
2143 006174 BR    nxUNI1          ; then get next unit
2144
2145 006176 011137 002262   gtPHI1: MOV    (R1), DZCSR#      ; Get address of DZ11 CSR
2146
2147 006202 011137 002264   MOV    (R1), DZRBUF#      ; Get address of DZ11 RBUF/LPR
2148 006206 062737 000002 002264   ADD    #2, DZRBUF#      ; (DZRBUF# = DZLPR#)
2149
2150 006214 011137 002266   MOV    (R1), DZTCR#      ; Get address of DZ11 TCR
2151 006220 062737 000004 002266   ADD    #4, DZTCR#
2152
2153 006226 012137 002270   MOV    (R1) +, DZTDR#      ; Get address of DZ11 TDR
2154 006232 062737 000006 002270   ADD    #6, DZTDR#
2155
2156 006240 011137 002252   MOV    (R1), DZrVCC#      ; Get address of DZ11 receiver interrupt
2157 006244 011137 002254   MOV    (R1), DZrVCS#      ; vector
2158 006250 062737 000002 002254   ADD    #2, DZrVCS#
2159
2160 006256 011137 002256   MOV    (R1), DZtVCC#      ; Get address of DZ11 transmitter interrupt
2161 006262 062737 000004 002256   ADD    #4, DZtVCC#      ; vector
2162 006270 012137 002260   MOV    (R1) +, DZtVCS#
2163 006274 062737 000006 002260   ADD    #6, DZtVCS#
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 #RUNG0#, UUT      ; 'Running on unit <UUT> in mode 0...'
      006320 013746 002216   MOV    UUT, -(SP)
      006324 012746 006740   MOV    #RUNG0#, -(SP)
      006330 012746 000002   MOV    #2, -(SP)
      006334 010600         MOV    SP, R0
      006336 104417         TRAP    C1PNTF
      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    C1PNTF

```

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      ; Compute program variables accordingly :
2182 006442 013701 002234 contI2: MOV DZPTY, R1      ; Load DZ11
2183 006446 072127 000005      ASH  #5, R1      ; bus priority
2184 006452 010137 002232      MOV  R1, TXPSW    ; into TXPSW
2185
2186 006456 005037 002224      CLR  ERRCNT      ;
2187 006462 004737 006506      JSR  PC, sblNI1    ; Compute "LNMBRi" and "LNMAPI" from "LNMAP"
2188
2189 006466 005737 002306      TST  sbaOK        ; If wrong "LNMAP" format,
2190 006472 001402      BEQ  sborI1              ; then abort pass
2191
2192 006474      EXIT  INIT
      006474 104432      TRAP C#EXIT
      006476 001044      .WORD L10016-.
2193
2194 006500      sborI1: DOCLN                      ;CLEAN UP AND ABORT PASS
      006500 104444      TRAP C#DOCLN
2195 006502      EXIT  INIT
      006502 104432      TRAP C#EXIT
      006504 001036      .WORD L10016-.
2196
2197

```

INITIALIZE SECTION

```

2200      ; subroutine to compute line map, number and parameters for lines x and y out
2201      ; of the DZ11
2202      ;
2203      ; Input parameter :
2204      ;
2205      ;     LNMAP
2206      ;
2207      ; Output parameters :
2208      ;     - sBAOK = 1 <=> successful ;
2209      ;     - if successful, line numbers (0..7) in LNNBRx, LNNBRy
2210      ;       line bitmaps in LNMAP, LNMAPx, LNMAPy
2211      ;       line parameters TLPRx, TLPRy
2212      ;
2213      ; Side effects :
2214      ;     - LNMAP is not modified ;
2215      ;     - LNNBR is left undefined.
2216      ;
2217      ;
2218      ;
2219      ;
2220      ;
2221      ;
2222      ;
2223      ;
2224      ;
2225      ;
2226      ;
2227      ;
2228      ;
2229      ;
2230      ;
2231      ;
2232      ;
2233      ;
2234      ;
2235      ;
2236      ;
2237      ;
2238      ;
2239      ;
2240      ;
2241      ;
2242      ;
2243      ;
2244      ;
2245      ;
2246      ;
2247      ;
2248      ;
2249      ;
2250      ;

```

2216	006506	005037	002306	sBLNI1:	CLR	sBAOK		
2217	006512	005037	002302		CLR	LNMAPx		
2218	006516	005037	002304		CLR	LNMAPy		
2220	006522	113702	002500		MOVB	LNMAP, R2		
2221	006526	112701	000001		MOVB	#001, R1		
2222	006532	005037	002272		CLR	LNNBR		
2224	006536	130102		nxBII1:	BITB	R1, R2		
2225	006540	001005			BNE	RLMPI1		; If found, then store value in LNMAPx-y
2227	006542	005237	002272		INC	LNNBR		; else increment line number and
2228	006546	106301			ASLB	R1		; shift set bit in R1 left one position
2229	006550	103372			BCC	nxBII1		; as long as no overflow occurs
2230	006552	000424			BR	erLMI1		
2232	006554	040102		RLMPI1:	BIC	R1, R2		; Clear bit in R2 that has just been found set
2234	006556	105737	002302		TSTB	LNMAPx		; If LNMAPx has already been assigned a value.
2235	006562	001045			BNE	RLMPI2		; then assign one to LNMAPy now
2237	006564	110137	002302		MOVB	R1, LNMAPx		; Store
2238	006570	013737	002272		MOV	LNNBR, LNNBRx		; results
2239	006576	013737	002236		MOV	TLPR0, TLPRx		; into line x
2240	006604	063737	002274		ADD	LNNBRx, TLPRx		; parameters
2242	006612	005737	002214		TST	TMODE		; If mode 0 and LNMAP format was given right.
2244	006616	001747			BEQ	nxBII1		; then now R2 = LNMAPy.
2246								; <- This is just to check for right format.
2247	006620	005702			TST	R2		; If mode 1.
2248	006622	001442			BEQ	succI1		; then only one line should be specified
2250	006624			erLMI1:	PRINTF	#WGLMP1, LNMAP, TMODE		
	006624	013746	002214		MOV	TMODE, -(SP)		
	006630	013746	002300		MOV	LNMAP, -(SP)		
	006634	012746	007357		MOV	#WGLMP1, -(SP)		
	006640	012746	000003		MOV	#3, -(SP)		
	006644	010600			MOV	SP, R0		
	006646	104417			TRAP	C\$PNTF		

INITIALIZE SECTION

```

2251      006650 062706 000010      ADD      #10 SP
2252      006654      PRINTF      #WGLMP2      ; 'Wrong number of DZ11 lines...'
      006654 012746 007467      MOV      #WGLMP2, -(SP)
      006660 012746 000001      MOV      #1, -(SP)
      006664 010600      MOV      SP, R0
      006666 104417      TRAP      CIPNTF
      006670 062706 000004      ADD      #4, SP
2253      006674 000207      RTS      PC
2254
2255      006676 105702      RLMP12: TSTB      R2      ; Check that no more than 2 lines were
2256      006700 001351      BNE      erLMT1      ; specified
2257
2258      006702 110137 002304      MOVB      R1, LNMARy      ; Store
2259      006706 013737 002272 002276      MOV      LNMAR, LNMARy      ; results
2260      006714 013737 002236 002242      MOV      TLPR0, TLPRy      ; into line y
2261      006722 063737 002276 002242      ADD      LNMARy, TLPRy      ; parameters
2262
2263      006730 012737 000001 002306      succ11: MOV      #1, sbAOK      ;
2264      006736 000207      RTS      PC
2265      ; end sbLNI1
2266

```

INITIALIZE SECTION

```

2280
2281
2282 006740      045      116      045  RUNGOa: .ASCIIZ  /#M#ARunning on unit #D2#A in mode 0 : pass-time is 2 minutes on the POP11 7
0.
2283 007056      045      116      045  RUNGOa: .ASCIIZ  /#M#AOnly tests 1, 2 and 3 are active in this mode./
2284 007141      045      116      045  RUNGLa: .ASCIIZ  /#M#ARunning on unit #D2#A in mode 1 : type "ctrl C" to stop./
2285 007236      045      116      045  RUNGLb: .ASCIIZ  /#M#AOnly tests 1 and 4 are active in this mode./
2286 007316      045      116      045  NOTAV:  .ASCIIZ  /#M#AUnit #D2#A is not available./
2287 007357      045      116      045  WGLMP1: .ASCIIZ  /#M#AMrong number of DZ11 lines in bitmap "#D3#A" for mode #D1#A test./
2288 007467      045      116      045  WGLMP2: .ASCIIZ  /#M#AChange Hardware P-table to correct.#N/
2289
2290
2291
2292
2293 007542      ENDINIT
      007542      L10016:
      007542 104411      TRAP      C#INIT
2294
2295
2296
2297

```

AUTODROP SECTION

.SBTTL AUTODROP SECTION

```

2300
2301
2302
2303
2304
2305
2306
2307
2308
2309 007544
      007544
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322 007544 013701 002262
2323 007550 012705 000004
2324
2325
2326 007554 012737 007606 000004
2327 007562 012737 000340 000006
2328
2329
2330 007570 005711
2331 007572 000240
2332 007574 062701 000002
2333 007600 005305
2334 007602 011372
2335 007604 000405
2336
2337
2338 007606 062706 000004
2339 007612
      007612 013700 002216
      007616 104451
2340
2341 007620 013737 002226 000004
2342 007626 013737 002230 000006
2343
2344
2345
2346
2347 007634
      007634
      007634 104461
2348
2349
2350
2351

```

```

      .EVEN
      BGNAUTO
L10017:
; Check if DZ11 responds
      MOV     DZCSR0, R1
      MOV     04, R5
; 4 DZ11 registers to be tested
; Set up timeout trap :
      MOV     021, 004
      MOV     0MAXPRI, 006
; address for timeout error trap handler
; priority level 7 in trap PSW to lock out
; other interrupts
10: TST      (R1)
      NOP
      ADD     02, R1
      DEC     R5
      BNE     10
      BR      30
; next register
; Decrement register count
; and branch back if not last register
; time out error trap handler :
20: ADD     04, SP
      DDDU    UUT
      MOV     UUT, R0
      TRAP    C100DU
; Pop old PC, PSW
; Drop unit under test
30: MOV     SAVE4, 004
      MOV     SAVE6, 006
; Restore original timeout vector
;
      ENDAUTO
L10017: TRAP    C100DU

```

CLEANUP CODING SECTION

.SBTTL CLEANUP CODING SECTION

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

2354
 2355
 2356
 2357
 2358
 2359
 2360
 2361
 2362
 2363
 2363
 2364
 2385
 2386
 2387
 2388
 2389
 2390
 2391
 2392
 2393

007636
 007636

007636
 007636 104433
 007640
 007640 013700 002256
 007644 104436

007646
 007646
 007646 104412

BGNCLN
 L10020:

BRESET ; bus reset
 TRAP C0RESET
 CLRVEC DZTVCCa ; Clear transmit interrupt vector
 MOV DZTVCCa,R0
 TRAP C0CVEC

ENDCLN
 L10020: TRAP C0CLEAN

```

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

BGNDU
L$DU::

.EVEN

PRINTF  @DROPD, R0      ; DRS has put # of unit to be dropped in R0
MOV     R0, -(SP)
MOV     @DROPD, -(SP)
MOV     @2, -(SP)
MOV     SP, R0
TRAP    C$PRINTF
ADD     @6, SP

EXIT    DU
.WORD   J$JMP
.WORD   L10021-2-.

DROPD:  .nlist  BEX
        .ASCIZ  /%sA unit %D2sA dropped./
        .list   BEX
        .EVEN

ENDDU

L10021: TRAP    C$DU

```

ADD UNIT SECTION

.SBTTL ADD UNIT SECTION

2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481

007730
007730

007730

007730 010046
007732 012746 007756
007736 012746 000002
007742 010600
007744 104417
007746 062706 000006

007752
007752 000167
007754 177750

007756 045 116
010004
010004
010004 104452

/////////
// 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.
/////////

L1AU:: BGN AU

.EVEN

PRINTF #ADDED, R0 ; DRS has put # of unit to be added in R0
MOV R0, -(SP)
MOV #ADDED, -(SP)
MOV #2, -(SP)
MOV SP, R0
TRAP C\$PRINTF
ADD #6, SP

EXIT DU
.WORD J\$JMP
.WORD L10021-2-.

.nlist BEX
.ASCIZ /#NAUnit #D2#A added./
.list BEX
.EVEN

END AU

L10022:
TRAP C\$AU

HARDWARE TESTS

.SBTTL HARDWARE TESTS

2484
2485
2486
2487
2488
2489
2490
2491
2492
2499
2505
2506
2507
2513
2514
2515
2527
2528
2529
2530
2536

010006

; START OF CODE BLOCK WHICH IS USED AS DATA
ROMMAP:;...

; TEST TO ...
;--

; BGNTST

; EXIT TST

; .EVEN
; ENDTST

HARDWARE TESTS

```

2585 010100          ENDSEG
      010100          100001:
      010100 104405      TRAP      C0ESEG

2586
2587 010102          BGNSEG
      010102 104404      TRAP      C0BSEG
2588 010104 012737 010124 000004 contD1: MOV      #11, R04      ; address for CSR read error trap handler
2589
2590 010112 032777 000020 172142      BIT      @CSRCLR, BDZCSR; Test CLR
2591 010120 000240      NOP
2592 010122 000423      BR        contD2
2593
2594      ; CSR read error handler ; DEVICE FATAL ERROR
2595 010124 062706 000004      11: ADD      #4, SP      ; Pop old PC, PSW
2596 010130 013737 002262 002320      MOV      DZCSR, ADDR      ; Report address location
2597 010136 013737 002216 002322      MOV      UUT, UNIT      ; Report unit number
2598 010144          ERROF      1, BUSTIM, pCSR;
      010144 104455      TRAP      C0EROF
      010146 000001      .WORD      1
      010150 003214      .WORD      BUSTIM
      010152 005026      .WORD      pCSR
2599 010154 013737 002226 000004      MOV      SAVE4, R04
2600 010162 013737 002230 000006      MOV      SAVE6, R06
2601 010170          DOCLN          ; Abort pass
      010170 104444      TRAP      C0DCLN

2602
2603 010172 013737 002226 000004 contD2: MOV      SAVE4, R04
2604 010200 013737 002230 000006      MOV      SAVE6, R06
2605 010206          ENDSEG
      010206          100011:
      010206 104405      TRAP      C0ESEG

2606 010210          ENDSUB
      010210          L10024:
      010210 104403      TRAP      C0ESUB

2607
2608 010212          BGNSUB      ; Start of subtest 2
      010212          T1.2:
      010212 104402      TRAP      C0BSUB

2609      ; Initialize DZ11 :
2610 010214 012777 000020 172040      MOV      @CSRCLR, BDZCSR; Set CLR bit of DZ11 CSR
2611 010222 012701 177754      MOV      @DLAY2, R1      ; Set up 2 seconds delay
2612
2613 010226 032777 000020 172026 nCLDD1: BIT      @CSRCLR, BDZCSR; 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          ERROF      2, DZINER, pDZIN; then there's a problem
      010262 104455      TRAP      C0EROF
      010264 000002      .WORD      2
      010266 003162      .WORD      DZINER
      010270 005152      .WORD      pDZIN
2620 010272          DOCLN          ; Abort pass
      010272 104444      TRAP      C0DCLN

2621
2622      ; test transmission on line x :

```

HARDWARE TESTS

```

2623
2624 010274 013777 002240 171762 contD3: MOV TLPRx, BDZLPRa ; Load parameters for line x
2625 010302 113777 002302 171756 MOVB LNMAPx, BDZTCRa ; Enable transmission on line x
2626
2627 010310 004737 010426 JSR PC, sbTED1 ; Transmit and test echo on line x
2628
2629 010314 005737 002306 TST sbAOK ; If normal,
2630 010320 001010 BNE contD4 ; then go on testing line y
2631
2632 010322 013737 002274 002272 MOV LNMBRx, LNMBR ; else report DEVICE FATAL error
2633 010330 ERROF 3, DZLBER, pDZLB;
010330 104455 TRAP C#EROF
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, BDZLPRa
2643 010362 113777 002304 171676 MOVB LNMAPy, BDZTCRa
2644
2645 010370 004737 010426 JSR PC, sbTED1 ; Transmit and test echo on line y
2646
2647 010374 005737 002306 TST sbAOK ; 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 LNMBRy, LNMBR ; else report DEVICE FATAL error
2652 010414 ERROF 3, DZLBER, pDZLB;
010414 104455 TRAP C#EROF
010416 000003 .WORD 3
010420 003125 .WORD DZLBER
010422 005240 .WORD pDZLB
2653 010424 DOCLN ; Abort pass
010424 104444 TRAP C#DCLN
2654

```

HARDWARE TESTS

```

2657      ; subroutine to transmit one character in maintenance loopback mode
2658      ; and check for echo
2659      ;
2660      ; Output parameter : sbAOK = 1 <=> success
2661      ;
2662 010426 005037 002306      sbTED1: CLR      sbAOK      ;
2663
2664 010432 012777 000050 171622      MOV      @MSEMAI, @DZCSRa; Enable maintenance loopback mode transmission
2665 010440 012701 177754      MOV      @DLAY2a, R1      ; Set up 2 seconds delay
2666
2667 010444 005777 171612      nTRYD1: TST      @DZCSRa      ; Wait
2668 010450 100414      BMI      contD7      ; for
2669
2670 010452      DLAY      DLAYarg      ; CSR Transmit Ready bit
2671 010472 005201      INC      R1      ; to set
2672 010474 001363      BNE      nTRYD1      ; If time-out
2673 010476 000137 010610      JMP      PBLMD1      ; then there's a problem
2674
2675 010502 112777 000101 171560      contD7: MOVB     @'A, @DZTDRA      ; Load character into Transmit Data Register
2676 010510 012701 177754      MOV      @DLAY2a, R1      ; Set up 2 seconds delay
2677
2678 010514 105777 171542      nRDND1: TSTB     @DZCSRa      ; REPEAT Wait
2679 010520 100021      BPL      contD8      ; UNTIL echo received
2680
2681 010522 017737 171536 002250      MOV      @DZRBUFFa, ECHO      ; Read received data
2682 010530 122737 000101 002250      CMPB     @'A, ECHO      ; If data received differs from data sent.
2683 010536 001024      BNE      PBLMD1      ; then there is a problem
2684
2685 010540 000240      NOP
2686 010542 000240      NOP
2687 010544 017737 171514 002250      MOV      @DZRBUFFa, ECHO      ; Try and read more data
2688 010552 100416      BMI      PBLMD1      ; If silo is not empty, there is a problem
2689
2690 010554 012737 000001 002306      MOV      @1, sbAOK      ; else All is OK
2691 010562 000207      RTS      PC      ;
2692
2693 010564      contD8: DLAY      DLAYarg      ; Wait
2694 010604 005201      INC      R1      ; routine
2695 010606 001342      BNE      nRDND1
2696
2697 010610 000207      PBLMD1: RTS      PC      ; When delay is elapsed or a problem arises.
2698      ; return with sbAOK = 0
2699      ; end sbTED1
2700
2701 010612      ENDSUB
2702 010612      L10025:      TRAP      C#ESUB
2703 010614      ENDTST
2704 010614      L10023:      TRAP      C#ETST
2705 010614 104401      .EVEN

```

HARDWARE TESTS

2708 010616

2709

2710

2711

2712

2713

2714

2715

2716

2717

2718

2719

2720

2721

2722

2723

2724

2725

2726

2727

2728

2729

2730

2731

2732

2733

2734

2735

2736

2737

2738

2739

2740

2741 C10616

2742

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 *****

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, sbIDG1      ; Initialize DZ11 for interrupt mode
2752                                ; transmission
2753
2754              ; Test echo on line x :
2755
2756 010634 013777 002240 171422      MOV      TLPRx, BDZLPRa ; Load parameters for line x
2757 010642 113777 002302 171416      MOVB     LNMAPx, BDZTCRa ; Enable transmission on line x
2758
2759 010650 013737 002274 002310      MOV      LNMBRx, LNTSTD ; Transmit
2760 010656 004737 002642      JSR      PC, sbTEG1      ; and test echo on line x
2761
2762 010662 005737 002306      TST      sbACK      ; 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, BDZLPRa
2768 010676 113777 002304 171362      MOVB     LNMAPy, BDZTCRa
2769
2770 010704 013737 002276 002310      MOV      LNMBRy, LNTSTD ; Transmit
2771 010712 004737 002642      JSR      PC, sbTEG1      ; and test echo on line y
2772
2773 010716 005737 002306      TST      sbACK      ; 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, pNMDE; Report error
      010724 104456      TRAP      C#ERRHRD
      010726 000004      .WORD      4
      010730 003112      .WORD      NOWDEC
      010732 005332      .WORD      pNMDE

2779
2780 010734 012777 000020 171320      endL1: MOV      #CSRCLR, BDZCSRa; 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

```

HARDWARE TESTS

2787 010750

2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842

BADHEAD

```
***** TEST3 *****  
;*   
;* Test active only in mode 0 :  
;*   
;* Purpose : switch the GS03-MD back and forth.  
;*   
;* Assumptions :  
;* - all previous tests ran successfully ;  
;* - WATCHDOG FUNCTION has priority (cf. note).  
;*   
;* Description :  
;* This test is the implementation of the following  
;* algorithm :  
;*   
;* Repeat twice, swapping lines x and y, the se-  
;* quence :  
;* - Try and switch GS03-MD to line x ;  
;* - Try and switch GS03-MD from line x to  
;* line y ;  
;* - Try and switch GS03-MD back from line y  
;* to line x ;  
;*   
;* Note :  
;* This diagnostic assumes that the switches are set  
;* to give the WATCHDOG FUNCTION priority. This  
;* means that the front panel switches should all be  
;* in the center position and the relay modules  
;* should all be configured for the same priority  
;* (see Option Description for details).  
;*   
;* IMPROPER SETTINGS CAN CAUSE UNEXPECTED ERRORS.  
;* WHICH WILL NOT NECESSARILY BE DIAGNOSED AS SUCH.  
;*   
;* Error messages :  
;* #5 - "No echo received back from GS03-MD on line  
;* # <line number>"  
;* "Check FORCE, MANUAL switches, priority  
;* setting and cables".  
;* #6 - "Both lines have switch priority over each  
;* other."  
;* "Check GS03 configuration."  
;* #7 - "Echo from GS03-MD received on both lines  
;* # <line number> and # <line number>."  
;* #8 - "GS03-MD failed to switch to line # <line  
;* number>"  
;* "No echo received back from GS03-MD on line  
;* # <line number>"  
;* "Check FORCE, MANUAL switches, priority  
;* setting and cables".  
;* #9 - "Echo from the GS03-MD received on wrong  
;* line # <line number> (expected : # <line  
;* number>)."  
;* "Echo is still being received on line  
;* # <line number> when actually transmitting  
;* on line # <line number> only."  
;* "Check GS03 configuration."
```

HARDWARE TESTS

2843

2844 010750

2845

1.
BADHEAD

..... TESTS

HARDWARE TESTS

```

2848 010750      BGNTST
      010750      TS::
2849              ; Initialization :
2850 010750 005737 002214      TST      TMODE      ; If mode 1, then skip this test
2851 010754 001402      BFG      contG1
2852 010756      EXIT      TST
      010756 104432      TRAP      C!EXIT
      010760 000522      .WORD      L10027-.

2853
2854 010762 004737 002572      contG1: JSR      PC, sbIDG1      ; Initialize DZ11 for interrupt mode
2855                                ; transmission
2856
2857 010766 013777 002240 171270      MOV      TLPRx, BDZLPRa ; Load parameters for line x
2858 010774 013777 002242 171262      MOV      TLPRy, BDZLPRa ; Load parameters for line y
2859
2860 011002 005037 011500      CLR      FTIMG1
2861
2862 011006 005037 002220      reptG1: CLR      SWPRY      ; Set switch priority to none.
2863
2864              ; Try and switch the GS03-MD to line x
2865
2866 011012 113777 002302 171246      MOVb     LMAPx, BDZTCRa ; Enable transmission on line x
2867
2868 011020 013737 002274 002310      MOV      LMBRx, LNTSTD ; Transmit
2869 011026 004737 002642      JSR      PC, sbTEG1      ; and test for echo on line x
2870
2871 011032 005737 002306      TST      sbACK      ; If successful,
2872 011036 001012      BNE      contG2      ; then go on testing
2873
2874 011040 013737 002274 002272      MOV      LMBRx, LMBR      ; else report error
2875 011046      ERRPRD      S, NO1LEC, pN1LE; and
      011046 104456      TRAP      C!ERRPRD
      011050 000005      .WORD      S
      011052 003066      .WORD      NO1LEC
      011054 005410      .WORD      pN1LE
2876 011056 012777 000020 171176      MOV      @CSRCLR, BDZCSRa; 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      LMBRx, OLDLMNB ; Load
2883 011076 013737 002276 002316      MOV      LMBRy, NEMLMNB ; parameters
2884 011104 013737 002304 002314      MOV      LMAPy, NEMLMNP ;
2885
2886 011112 004737 011252      JSR      PC, sbSWG1      ; 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      LMBRy, OLDLMNB ; Load
2893 011130 013737 002274 002316      MOV      LMBRx, NEMLMNB ; parameters

```

HARDWARE TESTS

```

2894 011136 013737 002302 002314      MOV     LMAPx, NEWLMP ;
2895                                     JSR     PC, abSMG1      ; and test
2896 011144 004737 011252
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     LMAPx, LMAPR ;
2905 011170 013737 002276 002274      MOV     LMAPy, LMAPR ;
2906 011176 013737 002272 002276      MOV     LMAPR, LMAPy ;
2907
2908 011204 013737 002302 002272      MOV     LMAPx, LMAPR ; LMAPR is used as a temporary here
2909 011212 013737 002304 002302      MOV     LMAPy, LMAPx ;
2910 011220 013737 002272 002304      MOV     LMAPR, LMAPy ;
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
      011246 104432      TRAP  C!EXIT
      011250 000232      .WORD L10027-.

```

HARDWARE TESTS

```

2920      ; subroutine to try and switch the GS03-WD from line OLDLNNB to line NEWLNNB
2921      ;
2922      ; Assumption : line OLDLNNB is already alive.
2923      ;
2924 011252 013777 002300 171006  sbSWG1: MOV     LNNAP, BDZTCRa ; Enable transmission on both lines
2925
2926 011260 013737 002316 002310      MOV     NEWLNNB, LNTSTD ; Start transmitting
2927 011266 004737 002642      JSR     PC, sbTEG1 ; and test for echo on line NEWLNNB
2928
2929 011272 005737 002306      /ST     sbACK ; If echo on line NEWLNNB,
2930 011276 001432      BEQ     contG4 ;
2931      ; then
2932 011300 005737 002220      TST     SWPRTY ; begin line NEWLNNB has switch priority
2933 011304 001406      BEQ     contG3 ; so, if OLDLNNB already had it :
2934
2935 011306      ERRHRD 6, PTYCFL, pPYCF ; 'Both lines have switch priority...'
2936 011306 104456      TRAP   CIERHRD
2937 011310 000006      .WORD 6
2938 011312 003044      .WORD PTYCFL
2939 011314 005462      .WORD pPYCF
2940
2941      ; 'Check FORCE, MANUAL...'
2942 011316      ESCAPE  TST
2943 011316 104410      TRAP  CIESCAPE
2944 011320 000162      .WORD L10027-.
2945
2946 011322 012737 000001 002220  contG3: MOV     #1, SWPRTY ; else record that NEWLNNB has priority
2947
2948 011330 013737 002312 002310      MOV     OLDLNNB, LNTSTD ; If there is still echo
2949 011336 004737 002642      JSR     PC, sbTEG1 ; on OLDLNNB,
2950 011342 005737 002306      TST     sbACK ; that means there's echo on both lines :
2951 011346 001406      BEQ     contG4 ;
2952
2953 011350      ERRHRD 7, WD2ECH, pWD2E ; 'Echo from GS03-WD received on both
2954 011350 104456      TRAP   CIERHRD
2955 011352 000007      .WORD 7
2956 011354 003021      .WORD WD2ECH
2957 011356 005530      .WORD pWD2E
2958
2959      ; lines.'
2960 011360      ESCAPE  TST
2961 011360 104410      TRAP  CIESCAPE
2962 011362 000120      .WORD L10027-.
2963
2964      ; end
2965 011364 013777 002314 170674  contG4: MOV     NEWLNNP, BDZTCRa ; Stop transmitting on line OLDLNNB
2966
2967 011372 013737 002316 002310      MOV     NEWLNNB, LNTSTD ; If no echo
2968 011400 004737 002642      JSR     PC, sbTEG1 ; on line
2969 011404 005737 002306      TST     sbACK ; NEWLNNB,
2970 011410 001011      BNE     contG5 ; then :
2971
2972 011412 013737 002316 002272      MOV     NEWLNNB, LNNBR ; 'GS03-WD fail to switch to line <NEWLNNB>'
2973 011420      ERRHRD 8, FAISWT, pFSWT ; 'No echo received on line <NEWLNNB>'
2974 011420 104456      TRAP   CIERHRD
2975 011422 000010      .WORD 8
2976 011424 002777      .WORD FAISWT
2977 011426 005566      .WORD pFSWT
2978
2979      ; 'Check FORCE, MANUAL...'

```

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, abTEG1 ; still echo
2965 011446 005737 002306 TST    abAOK ; on line
2966 011452 001411 BEQ     eabG1 ; 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!ERRRD
      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 eabG1: RTS  PC
2975                ; end abSWG1

```

HARDWARE TESTS

2984					
2985	011500	000000	FTING1: .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

3000

3001

3002

3003

3004

3005

3006

3007

3008

3009

3010

3011

3012

3013

3014

3015

3016

3017

3018

3019

3020

3021 011504

3022

3023 011504

011504

3024

3025 011504 005737 002214

3026 011510 001002

3027 011512

011512 104432

011514 000042

3028

3029 011516 004737 002572

3030

3031

3032

3033

3034 011522 042737 010000 002240

3035 011530 013777 002240 170526

3036 011536 113777 002302 170522

3037

3038 011544 012777 040040 170510

3039

3040 011552

011552 104422

3041 011554 000776

3042

3043 011556

011556

011556 104401

3044

3045

BADHEAD

***** TEST4 *****

;*

;* Test active only in mode 1 :

;*

;* Purpose : installation test.

;*

;* Assumption : all previous tests ran successfully.

;*

;* Description :

;* This test activates the line into the GS03-MD in
;* order for the operator to check that the LED's
;* react correctly :

;*

;* The GREEN or YELLOW LED corresponding to this
;* CPU's line into the GS03-MD should then turn on.
;* The associated RED LED should turn off after one
;* full GS03-MD clock pulse after this test begins
;* (which means that the RED clock LED should blink
;* twice at the most before this happens).

;*

;* Error message : none.

;*

BADHEAD

***** TEST4 *****

BGNTST

T4::

; Initialization :

TST

TMODE

; If mode 0, then skip this test

BNE

contA1

EXIT

TST

TRAP

C#EXIT

.WORD

L10030-.

contA1: JSR PC, sbIDG1

; Initialize DZ11 for interrupt mode
; transmission

; Transmit on line x :

BIC

#RCVRON, TLPRx ; Receiver will not be used

MOV

TLPRx, BDZLPRx ; Load parameters for line x

MOVB

LNMARx, BDZTCR; Enable transmission on line x

MOV

#MSETIE, BDZCSR; Enable interrupt mode transmission

loopA1: BREAK

TRAP

C#BRK

BR

loopA1

ENDTST

L10030:

TRAP

C#ETST

.EVEN

HARDWARE PARAMETER CODING SECTION

.SBTTL HARDWARE PARAMETER CODING SECTION

3048

3049

3050

3051

3052

3053

3054

3055

3056

3057

3058

3059

3060

3061 011560

011560 000053

011562

3062

3063 011562

011562 000031

011564 011642

011566 160010

011570 163776

3064 011572

011572 001031

011574 011646

011576 000300

011600 000777

3065 011602

011602 002032

011604 011655

011606 000007

011610 000004

011612 000007

3066 011614

011614 003032

011616 011660

011620 000377

011622 000001

011624 000377

3067 011626

011626 004032

011630 011675

011632 177777

011634 000000

011636 000001

3068

3069 011640

011640 024004

3070

3077

3078

3079 011642

103

123

122

CSR:

.ASCIZ

/CSR/

3080 011646

126

105

103

VECTOR:

.ASCIZ

/VECTOR/

3081 011655

102

122

000

PRIORITY:

.ASCIZ

/PR/

3082 011660

101

103

124

ACLINES:

.ASCIZ

/ACTIVE LINES/

3083 011675

127

110

111

WCHMODE:

.ASCIZ

/WHICH MODE/

3084

.list

BEX

```

////////////////////////////////////
; 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.
////////////////////////////////////

```

```

BGNHRD
.WORD L10031-L#HARD/2
L#HARD::

GPRMA CSR, 0, 0, 160010, 163776, YES
.WORD T#CODE
.WORD CSR
.WORD T#LOLIM
.WORD T#HILIM
GPRMA VECTOR, 2, 0, 300, 777, YES
.WORD T#CODE
.WORD VECTOR
.WORD T#LOLIM
.WORD T#HILIM
GPRMD PRIORITY, 4, 0, 000007, 4, 7, YES
.WORD T#CODE
.WORD PRIORITY
.WORD 000007
.WORD T#LOLIM
.WORD T#HILIM
GPRMD ACLINES, 6, 0, 000377, 1, 377, YES
.WORD T#CODE
.WORD ACLINES
.WORD 000377
.WORD T#LOLIM
.WORD T#HILIM
GPRMD WCHMODE, 10, 0, 177777, 0, 1, YES
.WORD T#CODE
.WORD WCHMODE
.WORD 177777
.WORD T#LOLIM
.WORD T#HILIM

EXIT HRD
.WORD T#CODE

```

HARDWARE PARAMETER CODING SECTION

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

SOFTWARE PARAMETER CODING SECTION

```
3095      .abttl  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
3108      011710      .WORD L10032-L#SOFT/2
3109      011712      L#SOFT::
3110
3111
3112
3113
3114      011712      ENDSFT
3115
3116      L10032:      .EVEN
3117
3118
3119
3120
3121
3122
3123
3124
```

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      .WORD      0
      012100      .WORD      L10035-./2-1
      012102
L10033:
3163
3164 012102      .word      160100
3165 012104      .word      300
3166 012106      .word      5
3167 012110      .word      3
3168 012112      .word      0
3169
3170 012114      ENDP TAB
      012114      L10035:
3171 0'2114      ENDSETUP
3172
3173
3174
3175
3176
3177      000001      .END
```

SYMBOL TABLE

ABORI1	006500	CSRCLE	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	DLAY25	177754 G	F#SUB	000002	LOOPA1	011552
BIT0	000001 G	C#BSEG	000004	DROPO	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	G#PMI1	006176	L#ACP	002110 G
BIT02	000004 G	C#CLK	000062	DZIN	004064	G#CNT0	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#DDU	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#OFST	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#ERRR	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#PREV	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

L1TEST	002114	G	NEMLMN	002316	PRI04	=	000200	G	TLPRO	002236	T11DAT	=	010035	
L1TML	002014	G	NFTMI1	006024	PRI05	=	000240	G	THMYER	002472	T11DU	=	010021	
L1UNIT	002012	G	NDECIL	004304	PRI06	=	000300	G	THODE	002214	T11MAR	=	010031	
L10001	002156		NOTAV	007316	PRI07	=	000340	G	TXPSM	002232	T11MM	=	010001	
L10002	002570		NOWDEC	003112	PTYCFL		003044		T1ARGC	=	000002	T11INI	=	010016
L10003	005076		NO1LEC	003066	PWDE		005530	G	T1CODE	=	024004	T11MSG	=	010014
L10004	005150		NROMD1	010514	PYCF		003315		T1ERRN	=	000011	T11PC	=	000C01
L10005	005236		NROMG1	002660	REUFCT	=	174000	G	T1EXCP	=	000000	T11PRO	=	010000
L10006	005330		NTRYD1	010444	RCVRON	=	010000	G	T1FLAG	=	000041	T11PTA	=	010034
L10007	005406		NIDE	003667	REPTG1		011006		T1FREE	=	012114	T11RPT	=	010015
L10010	005460		NABII1	006536	RLMPI1		006554		T1GMAN	=	000000	T11SEG	=	010001
L10011	005526		NXUNI1	006122	RLMPI2		006676		T1HILI	=	000001	T11SUP	=	010032
L10012	005564		NILE	003602	ROMAP		010006		T1LAST	=	000001	T11SRV	=	010002
L10013	005662		OLDLMN	002312	RUNGOA		006740		T1LOLI	=	000000	T11SUB	=	010025
L10014	005770		O1APTS	=	000000	RUNG08		007056	T1LSYM	=	010000	T11TES	=	010030
L10015	005772		O1AU	=	000001	RUNG1A		007141	T1LTND	=	000004	T1		010006 G
L10016	007542		O1P2R	=	000000	RUNG1B		007236	T1NEST	=	177777	T1.1		010006
L10017	007634		O1BGMS	=	000000	SAVE4		002226	T1NS0	=	000000	T1.2		010212
L10020	007646		O1DU	=	000001	SAVE6		002230	T1NS1	=	000005	T2		010616 G
L10021	007726		O1ERRT	=	000000	SBACK		002306	T1NS2	=	000002	T3		010750 G
L10022	010004		O1GMSM	=	000000	SBIDG1		002572	T1NS3	=	000005	T4		011504 G
L10023	010614		O1POIN	=	000001	SBLMI1		006506	T1PCNT	=	000000	UAM	=	000200 G
L10024	010210		O1SETU	=	000001	SBSMG1		011252	T1PTAB	=	010034	UNIT		002322
L10025	010612		PBLMD1	010610	SBTED1		010426		T1PTHV	=	000001	UUT		002216
L10026	010746		PCSR	005026	SBTEG1		002642		T1PTNU	=	000001	VECTOR		011646
L10027	011502		PCSRM	005130	SBWTG1		002342		T1SAVL	=	177777	WAITG1		002716
L10030	011556		PDZIN	005152	SBWTG2		002362		T1SEGL	=	177777	WCHMOD		011675
L10031	011710		PDZLB	005240	STEC1L		004371		T1SEX0	=	010001	WDZE		003416
L10032	011712		PFSMF	005664	SUCG1		002744		T1SIZE	=	000007	WDZCH		003021
L10033	012102		PFSMT	005566	SUCI1		006730		T1SUBN	=	000000	MGLMP1		007357
L10035	012114		PNT	=	001000	SVCGR1	=	000000	T1TAGL	=	177777	MGLMP2		007467
MAERR	002222		PWDE	005332	SVCINS	=	000000		T1TAGN	=	010036	X1ALMA	=	000000
MAUPRI	=	000340	PWLE	005410	SVCISB	=	000000		T1TEMP	=	000000	X1FALS	=	000040
MOD1I1	006372		PPYCF	005462	SVCTAG	=	000000		T1TEST	=	000004	X1OFFS	=	000400
MSEMAI	=	000050	PRI	=	002000	SVCTST	=	000000	T1TSTM	=	177777	X1TRUE	=	000020
MSETIE	=	040040	PRI0T	011655	SVTXG1	=	002562	G	T1TSTS	=	000001	ZDZGAO	=	002000 G
NCLD01	010226		PRI00	=	000000	SMPRTY		002220	T11AU	=	010022	1LSTIN	=	000000
NCLDG1	002626		PRI01	=	000040	S1LSYM	=	010000	T11AUT	=	010017	1LSTTA	=	000000
NETYG1	002666		PRI02	=	000100	TLPRX		002240	T11CLE	=	010020	1PATCH	=	011712 G
NEMLMN	002314		PRI03	=	000140	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

ZDZGAO.BIN,ZDZGAO-(PIMASA.D.GS.PDP.SSDC)LIBA.MLB/ML,ZDZGAO