

GS3WD

GS3WD/DZ11 LGC DIAG
CZDZGAO

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

FEB 1985
digital
Made In USA



CZDZGAO GS340/DZ11 LGC DIAG
PROGRAM DOCUMENT

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

7
8
9
10 .rem 0

11 IDENTIFICATION
12 -----

13 PRODUCT CODE: AC-T918A-MC
14
15 PRODUCT NAME: CZDZGAO GS340/DZ11 LGC DIAG
16
17 PRODUCT DATE: JULY 1984
18
19 MAINTAINER: CSS ANNECY
20
21 AUTHOR: Jean-Christophe PINASA
22
23
24
25
26

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

31 NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
32 SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
33 AFFILIATED COMPANIES.
34

35 COPYRIGHT (C) 1979, 1984 BY DIGITAL EQUIPMENT CORPORATION
36

37 THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
38
39

40 DIGITAL PDP UNIBUS MASSBUS
41 DEC DECUS DECTAPE

PROGRAM DOCUMENT

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

Table of contents

1.0 Introduction

1.1 Program abstract

1.2 Hardware description

1.3 Hardware configuration

1.4 Diagnostic description

2.0 Hardware requirements

3.0 Preliminary program requirements

4.0 General program considerations

4.1 Diagnostic supervisor

4.2 Execution time

4.3 XXDP.

4.4 Memory management

5.0 Program load media

6.0 Operating instructions

6.1 Loading and starting procedures

6.1.1 Loading procedures

6.1.2 Starting procedures

6.1.3 Steps for quick and simple execution

6.2 Initial dialogue

6.3 Program options

6.3.1 START command

6.3.2 RESTART command

6.3.3 CONTINUE command

6.3.4 PROCEED command

6.3.5 ADD command

6.3.6 DROP command

6.3.7 PRINT command

6.3.8 DISPLAY command

6.3.9 FLAGS command

6.3.10 ZFLAGS command

6.3.11 Control characters

6.3.12 Hardware parameters

6.3.13 Software parameters

6.3.14 Extended discussion of p-table dialogue

7.0 Tests descriptions

8.0 Error information

8.1 Error reporting

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-W0 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-W0 ;
- 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-HD logic module.

The GS03-HD 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 GS3MD/DZ11 LGC DIAG

The filename is : ZDZGAO.BIN

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

- Diagnostic test (mode 0)

This part will check all the GS03-HD 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-HD 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

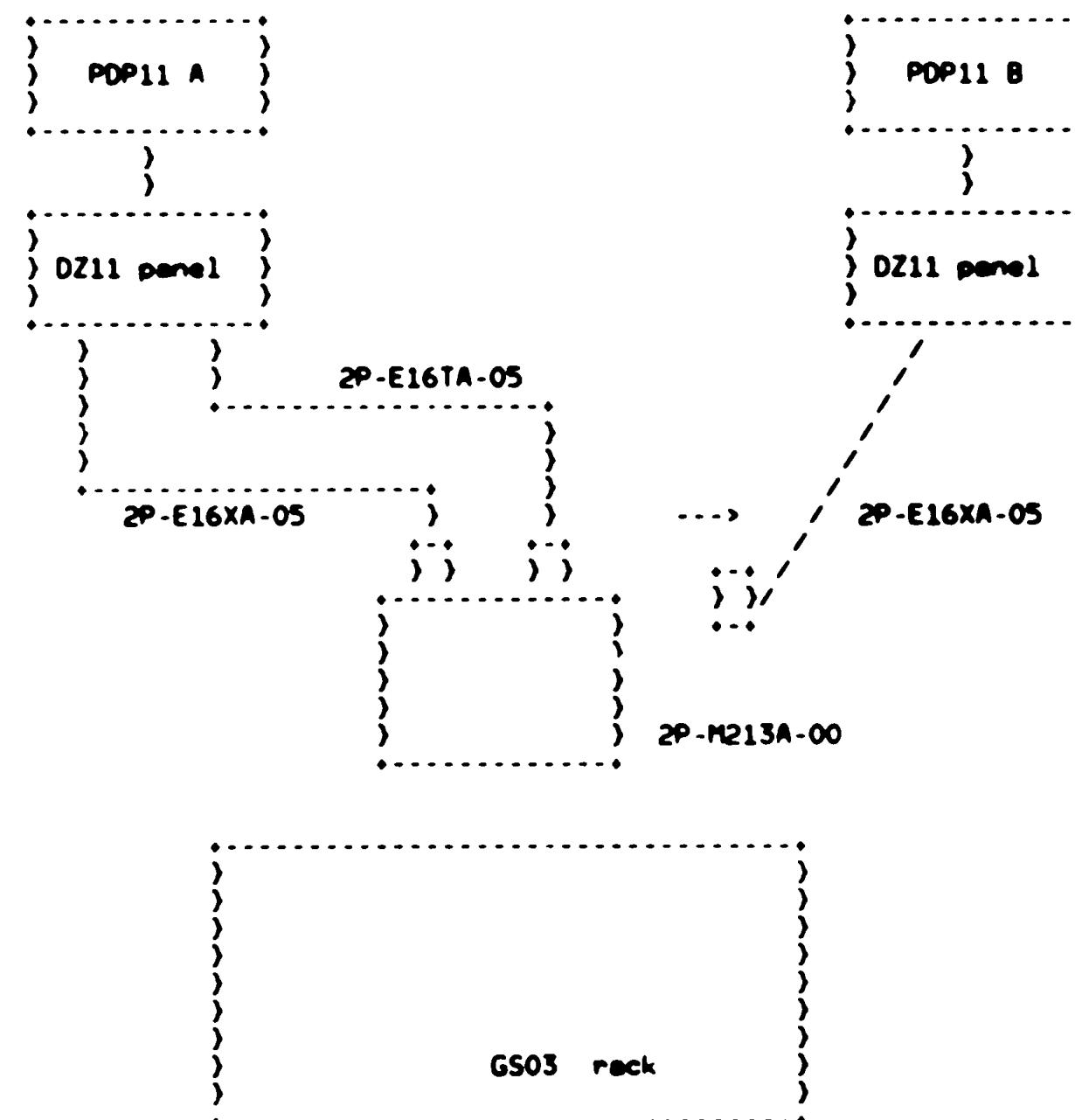
222

223

224

225

Example: Diagnostic running on PDP11 A.



PROGRAM DOCUMENT

227

228

229

230

231

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

232

233

234

235

236

237

- Disable the highest priority commands by :

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

- 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 ZDZGAO 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

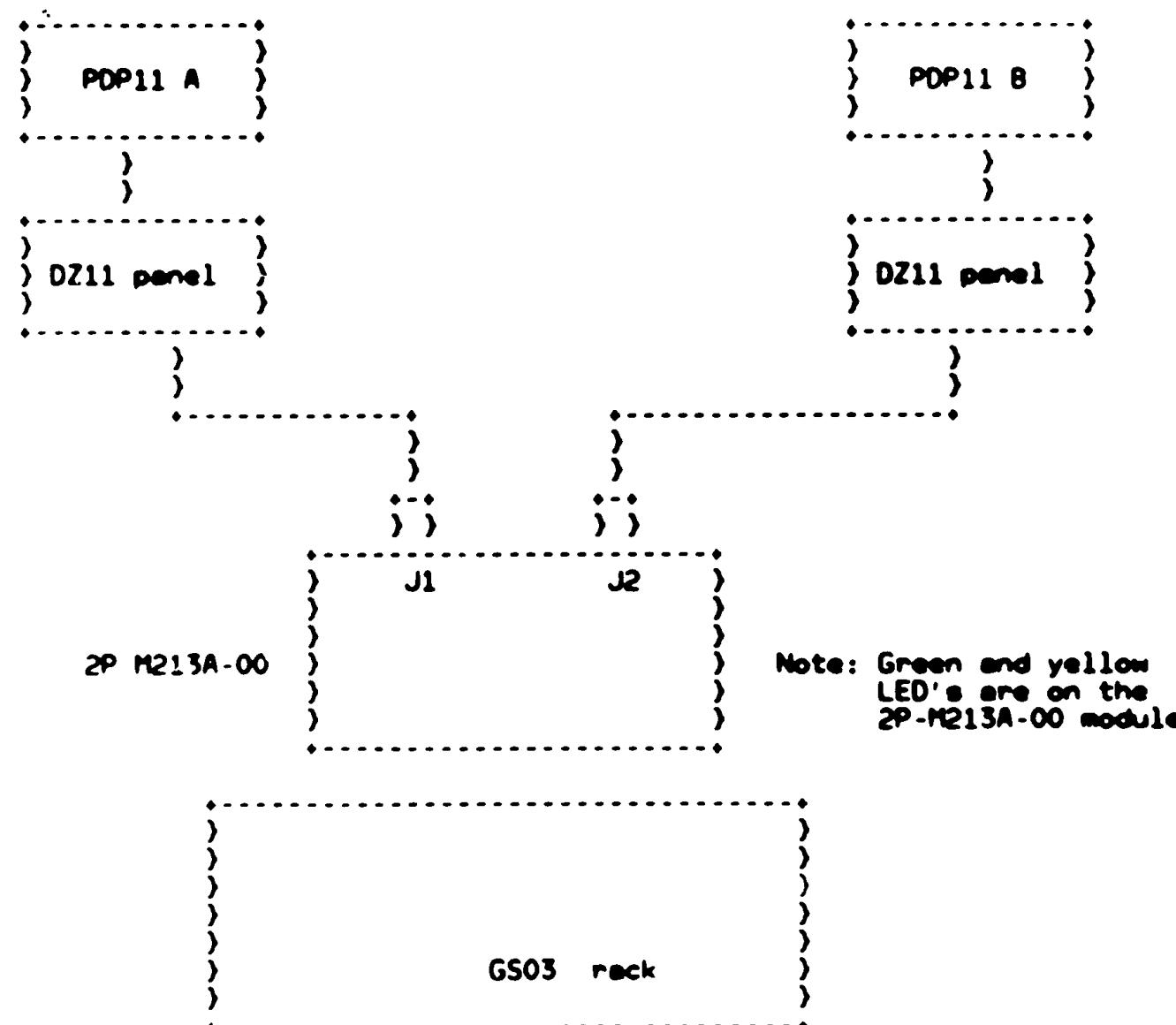
306

307

1.3.2 Installation test (MODE 1)

This test will allow to check GS03-WD 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-WD.

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

- Running ZDZGAO 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 ZDZGAO 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

2.0 Hardware requirements

374

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

375

376

377

378

379

380

381

382

WARNING :

383

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.

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

3.0 Preliminary program requirements

399

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

400

401

402

403

404

405

406

4.0 General program considerations

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

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

FZDZGAO GS3WD/DZ11 LGC DIAG

MACRO M1200 03-AUG-84 15:01 PAGE 11-1

SEQ 0011

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

6.0 Operating instructions

436

437

438

6.1 Loading and starting procedures

439

440

441

6.1.1 Loading procedures

442

443

444

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

445

446

447

6.1.2 Starting procedures

448

449

450

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

451

452

453

6.1.3 Steps for quick and simple execution

454

455

456

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

457

458

459

460

461

462

463

- 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 DIAG. RUN-TIME SERVICES
471 ZDZGAO-A-0
472 CZDZGAO GS3WD.DZ11 LGC DIAG
473 UNIT IS GS03WD MODULE
474 RESTART ADDR. 147670
475 DR>START
476
477 CHANGE HW (L) ? Y
478
479 # UNITS (D) ? 1
480
481 UNIT 0
482 CSR (0) 160100 ? 160340 : The CSB address is 160340 (range =
483 : 160010-163776)
484 VECTOR (0) 300 ? 460 : Vector address is 460 (range = 300-777)
485 BR (0) 5 ? 6 : BR interrupt level is 6 (range = 4-7)
486 ACTIVE LINES (0) 3 ? <CR> : Defines the line(s) of the DZ11 connected
487 : to the GS03-WD (octal bitmap format :
488 : range = 0-377)
489 WHICH MODE (0) 0 ? <CR> : Here (default value) : lines 0 and 1
490 : mode 0 = hardware test
491 : : caution : connect cables as
492 : : described in the diagnostic
493 : : header and in the option desc.
494 : mode 1 = installation test
495 : : with visual inspection of LED's
496 : : caution : in this mode, the
497 : : diagnostic will run continuously.
498 : To stop it, type "ctrl C".
499 : See header or option description for
500 : more details.
501 Running on unit 0 in mode 0 : pass-time is 2 minutes on the PDP11/70.
502 Only tests 1, 2 and 3 are active in this mode.
503
504
505 Example: Running "CZDZGAO 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 GS3WD.DZ11 LGC DIAG
UNIT IS GS03WD 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

C2

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:

HOE halt on error, causing command mode to be entered when an error is encountered

LOE loop on error, causing the diagnostic to loop continuously within the smallest defined block of coding (segment, subtest, or test) containing the error

IER inhibit error reporting

IBE inhibit basic error reports

IXE inhibit extended error reports

PROGRAM DOCUMENT

- 586 • PRI direct all messages to a line printer
587 PNT print number of test being executed
588 BOE bell on error
589 •• UAM run in unattended mode, bypassing manual intervention tests
590 ISR inhibit statistical reports
591 ADR execute autodrop code
592 IDU inhibit dropping of units by diagnostic
593 LOT loop on test
594 •• EVL evaluate
595
596 • NOT TO BE USED if a line printer is not available.
597 •• Of no use in this diagnostic.

599
600 The flags named or equated to 1 are set, those equated to 0
601 are cleared. A flag not specified is cleared. If the flags
602 switch is not given all flags are cleared. See example at
603 end of 6.3.1.5.

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

604
605 <INCR> is a decimal number indicating how often (in terms of
606 passes) it is desired that the end of pass message be
607 printed. The default is at the end of every pass. See
608 example at end of 6.3.1.5.

6.3.1.5 Effect of start command

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

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

PROGRAM DOCUMENT

636
637
638
639
640

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.

641
642
643
644
645
646
647
648
649

When the question "8 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.

650
651
652

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

653
654
655
656
657
658
659
660
661

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.

662
663

6.3.2 RESTART command

664
665

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

666
667

6.3.2.1 TESTS, PASS, and FLAGS switches

668
669

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

670
671

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

672
673

<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.

674
675

676
677

678
679

680
681

682
683

684
685

686
687

688
689

690

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

CONTINUE/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 cell. 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 isr (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

6.3.10 ZFLAGS command

859

860

ZFL(AGS)

861

862

863

864

865

6.3.10.1 Effect of ZFLAGS command

866

All flags are cleared.

867

868

869

870

6.3.11 Control Characters

871

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

872

873

874

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.

875

876

877

878

879

880

881

882

883

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 MW (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 NNN" 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

SEQ 0023

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 (D) ? 16

UNIT 1

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

UNIT 21

<QUESTION 1> ?
<QUESTION 2> ? 7-11..13-15
<QUESTION 3> ? 77

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

The second time thru the series, tables 16 thru the end are going to be affected (note that this piece of information is printed out for the 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 ***** TEST 2 *****
1102
1103 • Test active only in mode 0 :
1104
1105 • Purpose : check that characters are echoed back
1106 from the GS03-WD.
1107
1108 • Assumption : the previous test ran successfully.
1109
1110
1111 • Description :
1112 The two lines out of the DZ11 are arbitrarily
1113 named line x and line y.
1114 A first attempt will be made to receive echo
1115 back from the GS03-WD on line x. If it is not
1116 successful, another attempt will be made to
1117 receive echo on line y. If this cannot be
1118 achieved either, a hard error warning will be
1119 printed.
1120
1121
1122 • Note :
1123 This diagnostic detects that the GS03-WD switches
1124 to one line by receiving echoed characters back
1125 from the GS03-WD on that line.
1126 This is why, before other tests check correct
1127 switching, this test first checks that echo can
1128 be received back from the GS03-WD, on at least
1129 one line.
1130
1131 • Error message :
1132 • 04 - "No echo received back from the GS03-WD on
1133 either line 0 <line number> or 0 <line number>"
1134 "Check cabling and dip switch E18 (must be OFF)"
1135 *****

PROGRAM DOCUMENT

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

PROGRAM DOCUMENT

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

1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214 TEST 4
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227

• Test active only in mode 1 :
• Purpose : installation test.
• Assumption : all previous tests ran successfully.
• Description :
This test activates the line into the GS03-HD 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-HD should then turn on.
The associated RED LED should turn off after one
full GS03-HD 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 cell, 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 HWD ERR 00005 ON UNIT 00 TST 003 SUB 000 PC: 011046
NO ECHO ON ONE LINE

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

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

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

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

PROGRAM DOCUMENT

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

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 ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1322 ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1323 ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1324 ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1325
1326
1327

PROGRAM HEADER

```

1330 .SBTTL PROGRAM HEADER
1331
1332
1333
1334
1335
1336 002000
1337
1338
1339
1357
1358 002000      HEADER ZDZGAO, A, O, 270..0
       002000      ;DIAGNOSTIC NAME
       002000      .ASCII /Z/
       002001      .ASCII /D/
       002002      .ASCII /Z/
       002003      .ASCII /G/
       002004      .ASCII /A/
       002005      .ASCII /O/
       002006      .BYTE 0
       002007      .BYTE 0
       002010      L$REV::   .WORD 132          ;REVISION LEVEL
       002010      101
       002011      L$DEPO::  .ASCII /A/
       002011      060          .ASCII /O/          ;0
       002012      L$UNIT::  .WORD 104          ;NUMBER OF UNITS
       002012      000001
       002014      L$TML::   .WORD 132          ;LONGEST TEST TIME
       002014      000416
       002016      L$HPCP::  .WORD 011562        ;POINTER TO H.W. QUES.
       002016      011562
       002020      L$SPCP::  .WORD 000000        ;POINTER TO S.W. QUES.
       002020      000000
       002022      L$HPTP::  .WORD 002144        ;PTR. TO DEF. H.W. PTABLE
       002022      002144
       002024      L$STP::   .WORD 000000        ;PTR. TO S.W. PTABLE
       002024      000000
       002026      L$ADP::   .WORD 012076        ;DIAG. END ADDRESS
       002026      012076
       002030      L$STA::   .WORD 000000        ;RESERVED FOR APT STATS
       002030      000000
       002032      L$CO::    .WORD 000000
       002032      000000
       002034      L$DTYP::  .WORD 000000        ;DIAGNOSTIC TYPE
       002034      000000
       002036      L$APT::   .WORD 000000        ;APT EXPANSION
       002036      000000
       002040      L$DTP::   .WORD 002132        ;PTR. TO DISPATCH TABLE
       002040      002132
       002042      L$PRIOR:: .WORD 000000        ;DIAGNOSTIC RUN PRIORITY
       002042      000000
       002044      L$ENVI::  .WORD 000000        ;FLAGS DESCRIBE HOW IT WAS SETUP
       002044      000000
       002046      L$EXP1::  .WORD 000000        ;EXPANSION WORD
       002046      000000
       002050      L$MREV::  .WORD 003           ;SVC REV AND EDIT #
       002050      003          .BYTE 000000      ;REVISION

```

PROGRAM HEADER

002051	003	.BYTE	C\$EDIT	
002052		L\$EF:::		;DIAG. EVENT FLAGS
002052	000000	.WORD	0	
002054	000000	.WORD	0	
002056		L\$SPC:::		
002056	000000	.WORD	0	
002060		L\$DEVP:::		
002060	002324	.WORD	L\$DVTYPE	; POINTER TO DEVICE TYPE LIST
002062		L\$REPP:::		
002062	000000	.WORD	0	;PTR. TO REPORT CODE
002064		L\$EXP4:::		
002064	000000	.WORD	0	
002066		L\$EXP5:::		
002066	000000	.WORD	0	
002070		L\$AUT:::		
002070	007730	.WORD	L\$AU	;PTR. TO ADD UNIT CODE
002072		L\$DUT:::		
002072	007650	.WORD	L\$DU	;PTR. TO DROP UNIT CODE
002074		L\$LUN:::		
002074	000000	.WORD	0	;LUN FOR EXERCISERS TO FILL
002076		L\$DESP:::		
002076	002156	.WORD	L\$DESC	;POINTER TO DIAG. DESCRIPTION
002100		L\$LOAD:::		
002100	104035	EMT	E\$LOAD	;GENERATE SPECIAL AUTOLOAD EMT
002102		L\$ETP:::		
002102	000000	.WORD	0	;POINTER TO ERRRTBL
002104		L\$ICP:::		
002104	005774	.WORD	L\$INIT	;PTR. TO INIT CODE
002106		L\$CCP:::		
002106	007636	.WORD	L\$CLEAN	;PTR. TO CLEAN-UP CODE
002110		L\$ACP:::		
002110	007544	.WORD	L\$AUTO	;PTR. TO AUTO CODE
002112		L\$PRT:::		
002112	002122	.WORD	L\$PROT	;PTR. TO PROTECT TABLE
002114		L\$TEST:::		
002114	000000	.WORD	0	;TEST NUMBER
002116		L\$DLY:::		
002116	000000	.WORD	0	;DELAY COUNT
002120		L\$HIME:::		
002120	000000	.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
002122

BGNPROT
L\$PROT::

1387

1388 002122 000000
1389 002124 177777
1390 002126 177777

0
-1
-1

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

1391

1392

1406

1407

1408 002130
1409

ENDPROT

DISPATCH TABLE

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

.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

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

DEFAULT HARDWARE P-TABLE

1444
1445
1446
1447 // THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1448 // THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1449 // IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
1450 // AND IS USED AS A " TEMPLATE" FOR BUILDING THE P-TABLE
1451
1452
1453
1454 002142 .enabl AMA
002142 000005 BGNHW DFPTBL
002144 .WORD L10001-L\$HW/2
002144
L\$HW:::
DFPTBL:::
1455
1465
1466
1467 002144 160100 .word 160100 : DZ11 CSR address
1468 002146 000300 .word 300 : DZ11 vector address
1469 002150 000005 .word 5 : interrupt priority level (5)
1470 002152 000003 .word 3 : bitmap of lines out of DZ11 into GS03-WD
1471 002154 000000 .word 0 : diagnostic test mode (0)
1472 : or installation test mode (1) selector
1473
1474 002156 ENDHW
002156
1475 L10001:

GLOBAL EQUATES SECTION

1478

.SECTL GLOBAL EQUATES SECTION

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490

1491

1492

1493

1494

1514 002156

EQUALS

: BIT DEFINITIONS

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.
000037 EF.RESTART-- 31.
000036 EF.CONTINUE-- 30.
000035 EF.NEW-- 29.
000034 EF.PWR-- 28.

: START COMMAND WAS ISSUED
: RESTART COMMAND WAS ISSUED
: CONTINUE COMMAND WAS ISSUED
: A NEW PASS HAS BEEN STARTED
: 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	HOE..	100000	
1515	000340	MAXPRI ..	340	: Highest processor priority : 7
1516	000020	CSRCLR ..	000020	: DZ11 CSR device clear bit set
1517	040040	MSETIE ..	040040	: DZ11 CSR Master Scan Enable and Transmitter
1518	000050	MSEMAI ..	000050	: Interrupt Enable bits set
1519	010000	RCVRON ..	010000	: DZ11 CSR Master Scan Enable and MAIntenance
1520	174000	RBUFCTL ..	174000	: loopback mode bits set
1521	000100	DLAYarg ..	100	: DZ11 LPR Receiver On bit set
1522	177754	DLAY2s ..	-20.	: DZ11 RBUF mask to get line number after BIC
1523	177622	DLAY11s ..	-110.	: argument providing a rough 0.1 second delay
1524				: when used with the DLAY macro on the 11/70
1525				: 2 seconds delay to wait for echo
1526				: 11 seconds delay to wait for echo
1527				
1528				
1529				
1530				
1531				
1532				
1533				;oooooooooooooooooooooooooooo
1534				;* PROGRAM EVENT FLAG DEFINITIONS
1535				;oooooooooooooooooooooooooooo
1536				
1537				
1538				

GLOBAL DATA SECTION

```

1541          .SBTTL GLOBAL DATA SECTION
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558 002156
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584 002212 000000
1585 002214 000000
1586
1587
1588
1589
1590
1591 002216 000000
1592
1593 002220 000000
1594
1595 002222 000005
1596 002224 000000
1597
1598 002226 000000
1599 002230 000000
1600

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

;oooooooooooooooooooooooooooooooooooooooooooo
;* STORAGE FOR DEVICE REGISTERS
;oooooooooooooooooooooooooooooooooooooooooooo
        DESCRIPT      <CZDZGA0 GS3WD.DZ11 LGC DIAG>
L8DESC::      .ASCIZ /CZDZGA0 GS3WD.DZ11 LGC DIAG/
103      132      104
132      107      101
060      040      107
123      063      127
104      056      104
132      061      061
040      114      107
103      040      104
111      101      107
000

.EVEN

;ERRTABL

;oooooooooooooooooooooooooooooooooooooooooooo
;* PROGRAM CONTROL FLAGS
;oooooooooooooooooooooooooooooooooooooooooooo
        FTIME: .word 0           ; boolean to record first initialization
        TMODE: .word 0           ; Test mode

;oooooooooooooooooooooooooooooooooooooooooooo
;* PROGRAM CONTROL PARAMETERS
;oooooooooooooooooooooooooooooooooooooooooooo
        UUT: .word 0             ; Unit under test
        SWPRTY: .word 0          ; Switch priority (line x, y or none)
        MAXERR: .word 5           ; max error count before dropping unit
        ERRCNT: .word 0           ; error count
        SAVE4: .word 0             ; temporary storage for timeout trap
        SAVE6: .word 0             ; vector

```

GLOBAL DATA SECTION

1602 ;oooooooooooooooooooooooooooo
1603 ;o MISCELLANEOUS STORAGE
1604 ;oooooooooooooooooooooooooooo
1605
1606 002232 000000 TXPSW: .word 0 ; transmitter interrupt vector PSM
1607
1608 002234 000000 DZPTY: .word 0 ; DZ11 priority
1609 002236 011070 TLPRO: .word 011070 ; predefined parameter description for lines into the
1610 ; GS03-WD:
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           ;***** PRIMARY REG ADRS STORAGE FOR THIS UNIT *****
1629           ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1630
1631           ;=====
1632           ;* POINTERS TO DZ11 VECTORS AND REGISTERS
1633           ;=====
1634
1635
1636 002252 000000   DZrVCCa: .word 0      ; DZ11 receiver interrupt vector PC address
1637 002254 000000   DZrVCSe: .word 0      ; DZ11 receiver interrupt vector PSW address
1638 002256 000000   DZtVCCa: .word 0      ; DZ11 transmitter interrupt vector PC address
1639 002260 000000   DZtVCSe: .word 0      ; DZ11 transmitter interrupt vector PSW address
1640 002262 000000   DZCSRa: .word 0       ; DZ11 control status register address
1641 002264          DZRBUFFa:             ; 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   LNBR: .word 0        ; Line
1651 002274 000000   LNBRx: .word 0       ; numbers
1652 002276 000000   LNBRy: .word 0       ; (0..7)
1653 002300 000000   LNMAP: .word 0       ; Line
1654 002302 000000   LNMAPx: .word 0      ; bitmaps
1655 002304 000000   LNMAPy: .word 0      ; (0..377)
1656
1657           ;***** STACK USED FOR SUBROUTINE LINKAGE *****
1658
1659
1660
1661           ;=====
1662           ;* SUBROUTINE LINKAGE PARAMETERS
1663           ;=====
1664
1665 002306 000000   sbAOK: .WORD 0       ; Subroutine execution report
1666
1667 002310 000000   LNTSTD: .WORD 0       ; Number of line echo is to be expected on
1668
1669 002312 000000   OLDLNNB: .WORD 0      ; Number of the line the GS03-WD is switched to
1670           ; when calling subroutine sbSW31
1671
1672 002314 000000   NEWLNMP: .WORD 0      ; Bitmap and
1673 002316 000000   NEWLNNB: .WORD 0      ; Number of the line the GS03-WD is to be
1674           ; switched to if subroutine sbSW31 succeeds
1675
1676 002320 000000   ADDR: .word 0        ; Parameters
1677 002322 000000   UNIT: .word 0        ; for error reports
1678
1679

```

GLOBAL TEXT SECTION

1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693 002324
002324 107 123 060
002327 063 127 104
002332 040 115 117
002335 104 125 114
002340 105 000

.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
;-----

DEVTYPE <GS03WD MODULE>

LDEVTYPE:
.ASCIZ /GS03WD MODULE/

.EVEN

1694
1695
1696
1703
1704
1705

GLOBAL SUBROUTINES

```
1708          .SBTTL GLOBAL SUBROUTINES
1709
1710          -----
1711          ; MACRO'S NEEDED TO CALL SUBROUTINES
1712          -----
1713
1714          ; macro to wait a few ms
1715
1716          ; Call sequence :      DLAY    D          0 < D < 177777
1717
1718
1719
1720          .macro DLAY    D
1721          MOV     #176630, DLAYC1
1722          MOV     #D, DLAYC2
1723          JSR     #BWTG2
1724
1725
1726          .endm
```

GLOBAL SUBROUTINES

```

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

```

GLOBAL SUBROUTINES

```

1772          ; Routine to drop unit after 5 errors
1773
1774          ; Call sequence : JSR    PC, CHKMAX
1775
1776
1777 002400      CHKMAX: INLOOP      ; If looping on error
1778 002400      TRAP   C$INLP
1779 002402      BCOMPLETE    1$      ; then exit
1780 002402      BCS    1$
1781 002404      RFLAGS   RO
1782 002404      TRAP   C$RFLA
1783 002406      032700  000040      BIT    #IDU, RO      ; If dropping of units is inhibited
1784 002412      001026      BNE    1$      ; then exit
1785 002414      005237  002224      INC    ERRCNT      ; Update error count
1786 002420      023737  002224      CMP    ERRCNT, MAXERR ; If there aren't too many errors
1787 002426      003420      BLE    1$      ; then exit
1788 002430      PRINTF   #TMNYERS, MAXERR, UUT
1789 002430      MOV     UUT,-(SP)
1790 002434      013746  002216      MOV     MAXERR,-(SP)
1791 002440      013746  002222      MOV     #TMNYERS,-(SP)
1792 002444      012746  002472      MOV     #3,-(SP)
1793 002450      010600      MOV     SP,RO
1794 002452      104417      TRAP   C$PNTF
1795 002454      062706  000010      ADD    #10,SP
1796          ; else print 'Maximum error count
1797          ; of <maxerr> exceeded for unit <UUT>'
1798          ; and drop unit
1799
1800 002460      DOOU    UUT
1801 002460      013700  002216      MOV    UUT,RO
1802 002464      104451      TRAP   C$DOOU
1803
1804
1805 002466      DOCLN   TRAP   C$DCLN      ; Abort subpass
1806 002466      104444      RTS    PC
1807
1808
1809 002470      000207      1$:    RTS    PC
1810
1811
1812 002472      045       116       045       TMNYERS:.ASCIZ  .nlist
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
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
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2580
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2620
2631
2632
2633
2634
2635
2636
2637
2638
2639
2630
2641
2642
2643
2644
2645
2646
2647
2648
2649
2640
2651
2652
2653
2654
2655
2656
2657
2658
2659
2650
2661
2662
2663
2664
2665
2666
2667
2668
2669
2660
2671
2672
2673
2674
2675
2676
2677
2678
2679
2670
2681
2682
2683
2684
2685
2686
2687
2688
2689
2680
2691
2692
2693
2694
2695
2696
2697
2698
2699
2690
2701
2702
2703
2704
2705
2706
2707
2708
2709
2700
2711
2712
2713
2714
2715
2716
2717
2718
2719
2710
2721
2722
2723
2724
2725
2726
2727
2728
2729
2720
2731
2732
2733
2734
2735
2736
2737
2738
2739
2730
2741
2742
2743
2744
2745
2746
2747
2748
2749
2740
2751
2752
2753
2754
2755
2756
2757
2758
2759
2750
2761
2762
2763
2764
2765
2766
2767
2768
2769
2760
2771
2772
2773
2774
2775
2776
2777
2778
2779
2770
2781
2782
2783
2784
2785
2786
2787
2788
2789
2780
2791
2792
2793
2794
2795
2796
2797
2798
2799
2790
2801
2802
2803
2804
2805
2806
2807
2808
2809
2800
2811
2812
2813
2814
2815
2816
2817
2818
2819
2810
2821
2822
2823
2824
2825
2826
2827
2828
2829
2820
2831
2832
2833
2834
2835
2836
2837
2838
2839
2830
2841
2842
2843
2844
2845
2846
2847
2848
2849
2840
2851
2852
2853
2854
2855
2856
2857
2858
2859
2850
2861
2862
2863
2864
2865
2866
2867
2868
2869
2860
2871
2872
2873
2874
2875
2876
2877
2878
2879
2870
2881
2882
2883
2884
2885
2886
2887
2888
2889
2880
2891
2892
2893
2894
2895
2896
2897
2898
2899
2890
2901
2902
2903
2904
2905
2906
2907
2908
2909
2900
2911
2912
2913
2914
2915
2916
2917
2918
2919
2910
2921
2922
2923
2924
2925
2926
2927
2928
2929
2920
2931
2932
2933
2934
2935
2936
2937
2938
2939
2930
2941
2942
2943
2944
2945
2946
2947
2948
2949
2940
2951
2952
2953
2954
2955
2956
2957
2958
2959
2950
2961
2962
2963
2964
2965
2966
2967
2968
2969
2960
2971
2972
2973
2974
2975
2976
2977
2978
2979
2970
2981
2982
2983
2984
2985
2986
2987
2988
2989
2980
2991
2992
2993
2994
2995
2996
2997
2998
2999
2990
3001
3002
3003
3004
3005
3006
3007
3008
3009
3000
3011
3012
3013
3014
3015
3016
3017
3018
3019
3010
3021
3022
3023
3024
3025
3026
3027
3028
3029
3020
3031
3032
3033
3034
3035
3036
3037
3038
3039
3030
3041
3042
3043
3044
3045
3046
3047
3048
3049
3040
3051
3052
3053
3054
3055
3056
3057
3058
3059
3050
3061
3062
3063
3064
3065
3066
3067
3068
3069
3060
3071
3072
3073
3074
3075
3076
3077
3078
3079
3070
3081
3082
3083
3084
3085
3086
3087
3088
3089
3080
3091
3092
3093
3094
3095
3096
3097
3098
3099
3090
3101
3102
3103
3104
3105
3106
3107
3108
3109
3100
3111
3112
3113
3114
3115
3116
3117
3118
3119
3110
3121
3122
3123
3124
3125
3126
3127
3128
3129
3120
3131
3132
3133
3134
3135
3136
3137
3138
```

CZDZGAO GS3WD/DZ11 LGC DIAG MACRO M1200 03-AUG-84 15:01 PAGE 53

SEQ 0048

GLOBAL SUBROUTINES

1808 ; service routine to transmit in interrupt mode :
1809 002562 BGNSRV svTXG1
002562 svTXG1:::
1810 002562 112777 000101 177500 MOVB *A, BDZTDRa ;
1811
1812 002570 ENDSRV
002570 L10002:
002570 000002 RTI
1813

GLOBAL SUBROUTINES

```

1816          : subroutine to initialize DZ11 for interrupt mode transmission
1817
1818
1819
1820 002572    : Calling sequence : JSR PC, abIDG1
1821          ; abIDG1: SETVEC DZTVCCa, #evTXG1, TXPSW
1822          002572    MOV TXPSW,-(SP)
1823          002576    MOV #evTXG1,-(SP)
1824          002602    MOV DZTVCCa,-(SP)
1825          002606    MOV #3,-(SP)
1826          002612    TRAP C$VEC
1827          002614    ADD #10,SP
1828          ; Set up transmitter interrupt vector
1829
1830          002620    012777  000020  177434    MOV #CSRCLR, $DZCSRa; Set CLR bit of DZ11 CSR
1831
1832          002626    032777  000020  177426  nCLDG1: BIT #CSRCLR, $DZCSRa; Test CLR
1833          002634    BREAK ; Authorize "control-C" abort
1834          002634    104422  TRAP C$BRK
1835          ~002636    001373  BNE nCLDG1      ; Wait until CSRCLR = 0
1836          3
1837          9 002640    000207  RTS PC
1838          ; end abIDG1
1839
1840

```

GLOBAL SUBROUTINES

L4

```

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, #DZCSR0; Enable interrupt mode transmission
1851 002654  012701  177622      MOV    #Delay11s, R1 ; Set up 11 seconds delay
1852
1853 002660  105777  177376      nRDNG1: TSTB   #DZCSR0      ; 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   DLAYorg    :
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
1876
1877          : end sbTEG1

```

GLOBAL SUBROUTINES

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

GLOBAL ERROR REPORT SECTION

1898 .SBTTL GLOBAL ERROR REPORT SECTION

1899

1900 ;//////////
1901 ;// THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
1902 ;// THAT ARE USED IN MORE THAN ONE TEST.
1903 ;//////////

1904

1905 .nlist BEX

1906 002754 105 103 110 FAISWF: .ASCIZ /ECHO ON WRONG LINE/
1907 002777 106 101 111 FAISWT: .ASCIZ /FAIL TO SWITCH TO/
1908 003021 105 103 110 WD2ECH: .ASCIZ /ECHO ON BOTH LINES/
1909 003044 120 122 111 PTYCFL: .ASCIZ /PRIORITY CONFLICT/
1910 003066 116 117 040 NO1LEC: .ASCIZ /NO ECHO ON ONE LINE/
1911 003112 116 117 040 NO1DEC: .ASCIZ /NO WD ECHO/
1912 003125 104 132 061 DZLBER: .ASCIZ /DZ11 INTERNAL LOOPBACK ERROR/
1913 003162 104 132 061 DZINER: .ASCIZ /DZ11 INITIALIZATION FAULT/
1914 003214 102 125 123 BUSTIM: .ASCIZ /BUS TIMEOUT/
1915 .EVEN

1916

1917 ;*****
1918 ;* BASIC ERROR REPORTS MESSAGES :
1919 ;*****

1920

1921 003230 045 116 045 FSWF: .ASCIZ /MKAEcho from GS03-WD received on wrong line # \$01#A (expected : # \$01#A)./
1922 003343 045 116 045 FSMT: .ASCIZ /MKAGS03-WD failed to switch to line # \$01/
1923 003416 045 116 045 WD2E: .ASCIZ /MKAEcho from GS03-WD received on both lines # \$01#A and # \$01/
1924 003515 045 116 045 PYCF: .ASCIZ /MKABoth lines have switch priority over each other./
1925 003602 045 116 045 NILE: .ASCIZ /MKANO echo received back from GS03-WD on line # \$01/
1926 003667 045 116 045 NWDE: .ASCIZ /MKANO echo received back from GS03-WD on either line # \$01#A or # \$01/
1927 003776 045 116 045 DZLB: .ASCIZ /MKADZ11 internal loop back malfunction on line # \$01/
1928 004064 045 116 045 DZIN: .ASCIZ /MKADZ11 failed to reset./
1929 004116 045 116 045 CSRw: .ASCIZ /MKAUncsuccessful attempt to write to DZ11 CSR at address #06/
1930 004213 045 116 045 CSRp: .ASCIZ /MKAUncsuccessful attempt to read DZ11 CSR at address #06/
1931 .EVEN

1932

1933 ;*****
1934 ;* EXTENDED ERROR REPORTS MESSAGES :
1935 ;*****

1936

1937 004304 045 116 045 NOECIL: .ASCIZ /MKANO echo received back from GS03-WD on line # \$01/
1938 004371 045 116 045 STECIL: .ASCIZ /MKAEcho is still being received on line # \$01#A when actually transmitting

1939 004523 045 116 045 CKFMSW: .ASCIZ /MKCheck FORCE, MANUAL switches, priority setting and cables./
1940 004622 045 116 045 CKDPSW: .ASCIZ /MKCheck cabling and dip switch E18 (must be OFF)./
1941 004706 045 116 045 CKGSCF: .ASCIZ /MKCheck GS03 configuration./
1942 004744 045 116 045 DZDIAG: .ASCIZ /MKARun DZ11 diagnostic./
1943 004775 045 116 045 CKDZAD: .ASCIZ /MKCheck DZ11 address./
1944 .list BEX
1945 .EVEN

1946

1947

1948

1949

1950

1951

1952

1953

GLOBAL ERROR REPORT SECTION

SEQ 0053

1956
1957
1958
1959
1960
1961
1962
1963 005026 ; MACRO'S NEEDED TO REPORT ERRORS
005026
1964 005026 , Error # 0 report
005026 BGNSG PCSRr ; CSR read error
005026 PCSRr:: PRINTB PCSRr, ADDR
005026 013746 002320 MOV ADDR,-(SP)
005032 012746 004213 MOV @CSRr,-(SP)
005036 012746 000002 MOV #2,-(SP)
005042 010600 MOV SP, R0
005044 104414 TRAP C;PNTB
005046 062706 000006 ADD #6, SP

1965 005052 PRINTX #CXDZAD
005052 012746 004775 MOV #CXDZAD,-(SP)
005056 012746 000001 MOV #1,-(SP)
005062 010600 MOV SP, R0
005064 104415 TRAP C;PNTX
005066 062706 000004 ADD #4, SP
1966 005072 004737 002400 JSR PC, CHKMAX ; check if too many errors

1967 005076 ENMSG L10003:
005076 104423 TRAP C;MSG

1968
1969
1970 005100 ; Error # 1 report
005100 BGNSG PCSRW ; CSR write error
005100 PCSRW:: PRINTB PCSRW, ADDR
005100 013746 002320 MOV ADDR,-(SP)
005104 012746 004116 MOV @CSRW,-(SP)
005110 012746 000002 MOV #2,-(SP)
005114 010600 MOV SP, R0
005116 104414 TRAP C;PNTB
005120 062706 000006 ADD #6, SP

1972 005124 PRINTX #CXDZAD
005124 012746 004775 MOV #CXDZAD,-(SP)
005130 012746 000001 MOV #1,-(SP)
005134 010600 MOV SP, R0
005136 104415 TRAP C;PNTX
005140 062706 000004 ADD #4, SP
1973 005144 004737 002400 JSR PC, CHKMAX ; check if too many errors

1974 005150 ENMSG L10004:
005150 104423 TRAP C;MSG

1975
1976
1977 005152 ; Error # 2 report
005152 BGNSG PDZIN ; DZ11 initialization error
005152 PDZIN:: PRINTB PDZIN
005152 012746 004064 MOV #DZIN,-(SP)
005156 012746 000001 MOV #1,-(SP)
005162 010600 MOV SP, R0
005164 104414 TRAP C;PNTB
005166 062706 000004 ADD #4, SP

GLOBAL ERROR REPORT SECTION

C5

1979	005172		PRINTX	ACKDZAD		
	005172	012746	004775	MOV	ACKDZAD, -(SP)	
	005176	012746	000001	MOV	#1, -(SP)	
	005202	010600		MOV	SP, RO	
	005204	104415		TRAP	C8PNTX	
	005206	062706	000004	ADD	#4, SP	
1980	005212		PRINTX	ODZDIAG		
	005212	012746	004744	MOV	ODZDIAG, -(SP)	
	005216	012746	000001	MOV	#1, -(SP)	
	005222	010600		MOV	SP, RO	
	005224	104415		TRAP	C8PNTX	
	005226	062706	000004	ADD	#4, SP	
1981	005232	004737	002400	JSR	PC, CHKMAX	; check if too many errors
1982	005236		ENDMSG			
	005236		L10005:	TRAP	C8MSG	
1983						
1984						
1985	005240					
	005240		BGMMSG	ODZLB		; DZ11 loopback error
1986	005240		POZLB::			
	005240	013746	002272	PRINTB	ODZLB, LNBR	
	005244	012746	003776	MOV	LNBR, -(SP)	
	005250	012746	000002	MOV	ODZLB, -(SP)	
	005254	010600		MOV	#2, -(SP)	
	005256	104414		MOV	SP, RO	
	005260	062706	000006	TRAP	C8PNTB	
1987	005264		PRINTX	ACKDZAD		
	005264	012746	004775	MOV	ACKDZAD, -(SP)	
	005270	012746	000001	MOV	#1, -(SP)	
	005274	010600		MOV	SP, RO	
	005276	104415		TRAP	C8PNTX	
	005300	062706	000004	ADD	#4, SP	
1988	005304		PRINTX	ODZDIAG		
	005304	012746	004744	MOV	ODZDIAG, -(SP)	
	005310	012746	000001	MOV	#1, -(SP)	
	005314	010600		MOV	SP, RO	
	005316	104415		TRAP	C8PNTX	
	005320	062706	000004	ADD	#4, SP	
1989	005324	004737	002400	JSR	PC, CHKMAX	; check if too many errors
1990	005330		ENDMSG			
	005330		L10006:	TRAP	C8MSG	
1991						
1992						
1993	005332					
	005332		BGMMSG	PNMDE		; WD fail to echo error
1994	005332		PNMDE::			
	005332	013746	002276	PRINTB	PNMDE, LNBRx, LNBRy	
	005336	013746	002274	MOV	LNBRy, -(SP)	
	005342	012746	003667	MOV	LNBRx, -(SP)	
	005346	012746	000003	MOV	PNMDE, -(SP)	
	005352	010600		MOV	#3, -(SP)	
	005354	104414		MOV	SP, RO	
	005356	062706	000010	TRAP	C8PNTB	
1995	005362		PRINTX	ACKDPSW		
	005362	012746	004622	MOV	ACKDPSW, -(SP)	

GLOBAL ERROR REPORT SECTION

D5

005366	012746	000001	MOV	01,-(SP)	
005372	010600		MOV	SP, R0	
005374	104415		TRAP	C8PNTX	
005376	062706	000004	ADD	#4, SP	
1996	005402	004737	JSR	PC, CHKMAX	
1997	005406			; check if too many errors	
	005406				
	005406	104423			
1998			ENDMSG		
1999			L10007:		
2000	005410		TRAP	C8MSG	
2001	005410			; Error # 5 report	
2001	005410		BGNMSG	BNILE	
	005410	013746		; No echo received on line x error	
	005414	012746	BNILE::		
	005420	012746	PRINTB	BNILE, LNNBRx	
	005424	010600	MOV	LNNBRx,-(SP)	
	005426	104414	MOV	BNILE,-(SP)	
	005430	062706	MOV	#2,-(SP)	
2002	005434	000006	PRINTX	MOV	SP, R0
	005434	012746	MOV	C8PNTB	
	005440	004523	ADD	#6, SP	
	005440	012746	JSCKFMSW		
	005444	000001	MOV	0CKFMSW,-(SP)	
	005444	010600	MOV	01,-(SP)	
	005446	104415	TRAP	SP, R0	
	005450	062706	ADD	C8PNTX	
2003	005454	000004	JSR	#4, SP	
2004	005460	004737	PC, CHKMAX	; check if too many errors	
	005460				
	005460	104423			
2005			ENDMSG		
2006			L10010:		
2007	005462		TRAP	C8MSG	
2008	005462			; Error # 6 report	
2008	005462		BGNMSG	PPYCF	
	005462	012746		; Both lines have priority error	
	005466	003515	PPYCF::		
	005466	012746	PRINTB	PPYCF	
	005472	000001	MOV	PPYCF,-(SP)	
	005472	010600	MOV	01,-(SP)	
	005474	104414	MOV	SP, R0	
	005476	062706	TRAP	C8PNTB	
2009	005502	000004	ADD	#4, SP	
	005502	012746	JSCKGSCF		
	005506	004706	MOV	0CKGSCF,-(SP)	
	005506	012746	MOV	01,-(SP)	
	005512	000001	MOV	SP, R0	
	005512	010600	TRAP	C8PNTX	
	005514	104415	ADD	#4, SP	
	005516	062706	JSR	PC, CHKMAX	
2010	005522	000004		; check if too many errors	
2011	005526	004737			
	005526		ENDMSG		
	005526	104423			
2012			L10011:		
2013			TRAP	C8MSG	
2014	005530			; Error # 7 report	
2014	005530		BGNMSG	PHD2E	
	005530			; Echo on both lines error	
	005530		PHD2E::		
	005530	013746	PRINTB	PHD2E, LNNBRx, LNNBRy	
	005534	013746	MOV	LNNBRy,-(SP)	
	005534	002274	MOV	LNNBRx,-(SP)	
	005540	012746	MOV	PHD2E,-(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	JSR	PC, CHKMAX	; check if too many errors
2017	005564				
	005564				
	005564	104423	ENDMSG	TRAP C;MSG	
2018					
2019					
2020	005566		: Error # 8 report		
	005566		BGNMSG	#FSWT	
			#FSWT:::		; WD fail to switch to line i error
2021	005566		PRINTB	#FSWT, NEWLNBS	
	005566	013746	MOV	NEWLNBS,-(SP)	
	005572	012746	MOV	#FSWT,-(SP)	
	005576	012746	MOV	#2,-(SP)	
	005602	010600	MOV	SP, R0	
	005604	104414	TRAP	C;PNTB	
	005606	062706	ADD	#6, SP	
2022	005612		PRINTX	#NOECIL, NEWLNBS	
	005612	013746	MOV	NEWLNBS,-(SP)	
	005616	012746	MOV	#NOECIL,-(SP)	
	005622	012746	MOV	#2,-(SP)	
	005626	010600	MOV	SP, R0	
	005630	104415	TRAP	C;PNTX	
	005632	062706	ADD	#6, SP	
2023	005636		PRINTX	#CKFMSW	
	005636	012746	MOV	#CKFMSW,-(SP)	
	005642	012746	MOV	#1,-(SP)	
	005646	010600	MOV	SP, R0	
	005650	104415	TRAP	C;PNTX	
	005652	062706	ADD	#6, SP	
2024	005656	004737	JSR	PC, CHKMAX	; check if too many errors
2025	005662		ENDMSG		
	005662		L10013:		
	005662	104423	TRAP	C;MSG	
2026					
2027					
2028	005664		: Error # 9 report		
	005664		BGNMSG	#FSWF	
			#FSWF:::		; WD echo on wrong line error
2029	005664		PRINTB	#FSWF, OLDDLNBS, NEWLNBS	
	005664	013746	MOV	NEWLNBS,-(SP)	
	005670	013746	MOV	OLDDLNBS,-(SP)	
	005674	012746	MOV	#FSWF,-(SP)	
	005700	012746	MOV	#3,-(SP)	
	005704	010600	MOV	SP, R0	
	005706	104414	TRAP	C;PNTB	
	005710	062706	ADD	#10, SP	
2030	005714		PRINTX	#STECIL, OLDDLNBS, NEWLNBS	
	005714	013746	MOV	NEWLNBS,-(SP)	
	005720	013746	MOV	OLDDLNBS,-(SP)	
	005724	012746	MOV	#STECIL,-(SP)	
	005730	012746	MOV	#3,-(SP)	
	005734	010600	MOV	SP, R0	
	005736	104415	TRAP	C;PNTX	
	005740	062706	ADD	#10, SP	
2031	005744		PRINTX	#CKGSCF	

CZDZGAO GS3WD/DZ11 LGC DIAG

MACRO M1200 03-AUG-80 15:01 PAGE 63-4

SEQ 0057

GLOBAL ERROR REPORT SECTION

005744	012746	004706	MOV	0CKGSCF,-(SP)
005750	012746	020001	MOV	01,-(SP)
005754	010600		MOV	SP, R0
005756	104415		TRAP	C0PNTX
005760	062706	000004	ADD	04, SP
2032	005764	004737	JSR	PC, CHKMAX
2033	005770			i check if too many errors
	005770		ENDMSG	
	005770	104423	L10014:	
			TRAP	C0MSG
2034				
2035				
2036				
2037				
2038				

REPORT CODING SECTION

2041

.SBTTL REPORT CODING SECTION

2042

2043

2044

2045

2046

2047

2048

2049 005772

100
; THE REPORT CODING SECTION CONTAINS THE
; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.

005772

100
BGNRPT
L\$RPT::

2050

2056

2057

2064

2065 005772

100
ENDRPT
L10015:
005772 TRAP C\$RPT
005772 104425

2066

2067

INITIALIZE SECTION

2070
2071
2072
2073
2074
2075
2076
2077 005774
 005774
2078
2079

.S8TTL 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 nFTMII1 ; 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 \$01, FTIME ;
 2111 006024 013737 002226 000004 nFTMII1: MOV SAVE4, \$04 ; else it is restored
 2112 006032 013737 002230 000006 MOV SAVE6, \$06 ;
 2113
 2114 006040 READEF @EF.START ; "START".
 006040 012700 000040 MOV @EF.START, RO
 006044 104447 TRAP C\$REFG
 2115 006046 BCOMPLETE ptUNI1 ;
 006046 103422 BCS ptUNI1
 2116
 2117 006050 READEF @EF.RESTART ; "RESTART" commands.
 006050 012700 000037 MOV @EF.RESTART, RO
 006054 104447 TRAP C\$REFG
 2118 006056 BCOMPLETE ptUNI1 ;
 006056 103416 BCS ptUNI1
 2119
 2120 006060 READEF @EF.PWR ; or POWER UP :
 006060 012700 000034 MOV @EF.PWR, RO
 006064 104447 TRAP C\$REFG
 2121 006066 BCOMPLETE ptUNI1 ; start with first unit (0 0)
 006066 103412 BCS ptUNI1
 2122
 2123 006070 READEF @EF.CONTINUE ; If this a "continue" command.
 006070 012700 000036 MOV @EF.CONTINUE, RO
 006074 104447 TRAP C\$REFG
 2124 006076 BNCOMPLETE contI1 ; then exit
 006076 103002 BCC contI1
 2125 006100 EXIT INIT ; (no re-initialization)
 006100 104432 TRAP C\$EXIT
 006102 001440 .WORD L10016-.
 2126
 2127 006104 contI1: READEF @EF.NEW ; If this is not a new pass.
 006104 012700 000035 MOV @EF.NEW, RO
 006110 104447 TRAP C\$REFG
 2128 006112 BNCOMPLETE nxUNI1 ; then get next unit
 006112 103003 BCC nxUNI1
 2129
 2130 006114 012737 177777 002216 ptUNI1: MOV \$-1, UUT ;
 2131
 2132 006122 005237 002216 nxUNI1: INC UUT ; Point to next unit
 2133 006126 023737 002216 002012 CMP UUT, L\$UNIT ; If there isn't any,
 2134 006134 002161 BGE aborI1 ; then end-of-pass
 2135

INITIALIZE SECTION

2138 : Load hardware parameters for unit under test.
 2139 006136 GPHARD UUT, R1 : Call to DRS to put p-table address in R1
 006136 013700 002216
 006142 104442
 006144 010001
 2140 006146 BCOMPLETE gtPMI1 :
 006146 103413 BCS gtPMI1
 2141
 2142 006150 PRINTF @NOTAV, UUT : If not available.
 006150 013746 002216
 006154 012746 007316
 006160 012746 000002
 006164 010600
 006166 104417
 006170 062706 000006
 2143 006174 000752 BR nxUNII : then get next unit
 2144
 2145 006176 011137 002262 gtPMI1: MOV (R1), DZCSR_a : Get address of DZ11 CSR
 2146
 2147 006202 011137 002264 MOV (R1), DZRBUF_a : Get address of DZ11 RBUF/LPR
 2148 006206 062737 000002 002264 ADD #2, DZRBUF_a : (DZRBUF_a = DZLPR_a)
 2149
 2150 006214 011137 002266 MOV (R1), DZTCR_a : Get address of DZ11 TCR
 2151 006220 062737 000004 002266 ADD #4, DZTCR_a
 2152
 2153 006226 012137 002270 MOV (R1) ., DZTDR_a : Get address of DZ11 TDR
 2154 006232 062737 000006 002270 ADD #6, DZTDR_a
 2155
 2156 006240 011137 002252 MOV (R1), DZrVCC_a : Get address of DZ11 receiver interrupt
 2157 006244 011137 002254 MOV (R1), DZrVC_a : vector
 2158 006250 062737 000002 002254 ADD #2, DZrVC_a
 2159
 2160 006256 011137 002256 MOV (R1), DZtVCC_a : Get address of DZ11 transmitter interrupt
 2161 006262 062737 000004 002256 ADD #4, DZtVCC_a : vector
 2162 006270 012137 002260 MOV (R1) ., DZtVC_a
 2163 006274 062737 000006 002260 ADD #6, DZtVC_a
 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 MOD1II1
 2171
 2172 006320 PRINTF @RUNGO_a, UUT : 'Running on unit <UUT> in mode 0...'
 006320 013746 002216
 006324 012746 006740
 006330 012746 000002
 006334 010600
 006336 104417
 006340 062706 000006
 2173 006344 PRINTF @RUNGO_b, UUT : 'Only tests 1, 2 and 3 are active...'
 006344 013746 002216
 006350 012746 007056
 006354 012746 000002
 006360 010600
 006362 104417 TRAP C8PNTF

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...'
	013746	002216		MOV	UUT,-(SP)
	006376	012746		MOV	@RUNG1a,-(SP)
	006402	012746		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
2183 006446 072127 000005
2184 006452 010137 002232
2185
2186 006456 00503/ 002224
2187 006462 004737 006506
2188
2189 006466 005737 002306
2190 006472 001402
2191
2192 006474      104432
    006474      104444
    006476 001044
2193
2194 006500      104444
    006500      104444
2195 006502      104432
    006502      104432
    006504 001036
2196
2197

contI2: MOV DZPTY, R1      ; Load DZ11
        ASH #5, R1      ; bus priority
        MOV R1, TXPSW     ; into TXPSW

CLR ERRCNT
JSR PC, $0LNII      ; Compute "LNnbr;" and "LNmap;" from "LNmap"

TST sbAOK
BEQ aborII      ; If wrong "LNmap" format,
                  ; then abort pass

EXIT INIT
TRAP C$EXIT
.WORD L10016-.

aborII: DOCLN      ; CLEAN UP AND ABORT PASS
        TRAP C$DOCLN
        EXIT INIT
        TRAP C$EXIT
        .WORD L10016-.

```

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 : - #BAOK = 1 <=> successful ;
 2209 : - if successful, line numbers (0..7) in LNMBRx, LNMBRy
 2210 : line bitmaps in LNMAP, LNMAPx, LNMAPy
 2211 : line parameters TLPRx, TLPRy
 2212 : Side effects :
 2213 : - LNMAP is not modified ;
 2214 : - LNMBR is left undefined.
 2215
 2216 006506 005037 002306
 2217 006512 005037 002302
 2218 006516 005037 002304
 2219
 2220 006522 113702 002500
 2221 006526 112701 000001
 2222 006532 005037 002272
 2223
 2224 006536 130102
 2225 006540 001005
 2226
 2227 006542 005237 002272
 2228 006546 106301
 2229 006550 103372
 2230 006552 000424
 2231
 2232 006554 040102
 2233
 2234 006556 105737 002302
 2235 006562 001045
 2236
 2237 006564 110137 002302
 2238 006570 013737 002272 002274
 2239 006576 013737 002236 002240
 2240 006604 063737 002274 002240
 2241
 2242 006612 005737 002214
 2243
 2244 006616 001747
 2245
 2246
 2247 006620 005702
 2248 006622 001442
 2249
 2250 006624
 006624 013746 002214
 006630 013746 002300
 006634 012746 007357
 006640 012746 000003
 006644 010600
 006646 104417

#BLNI1: CLR #BAOK ;
 CLR LNMAPx
 CLR LNMAPy
 MOV8 LNMAP, R2
 MOVB #001, R1
 CLR LNMBR
 nxBIII: BITB R1, R2
 BNE RLMPI1 ; If found, then store value in LNMAPx-y
 INC LNMBR ; else increment line number and
 ASLB R1 ; shift set bit in R1 left one position
 BCC nxBIII ; as long as no overflow occurs
 BR erLMI1
 RLMPI1: BIC R1, R2 ; Clear bit in R2 that has just been found set
 TSTB LNMAPx ; If LNMAPx has already been assigned a value,
 BNE RLMPI2 ; then assign one to LNMAPy now
 MOV8 R1, LNMAPx ; Store
 MOV LNMBR, LNMBRx ; results
 MOV TLPRO, TLPRx ; into line x
 ADD LNMBRx, TLPRx ; parameters
 TST TMODE ; If mode 0 and LNMAP format was given right,
 BEQ nxBIII ; then now R2 = LNMAPy.
 ; <- This is just to check for right format.
 ; If mode 1,
 TST R2 ; then only one line should be specified
 BEQ succII
 erLMI1: PRINTF #WGLMP1, LNMAP, TMODE
 MOV TMODE, -(SP)
 MOV LNMAP, -(SP)
 MOV #WGLMP1, -(SP)
 MOV #3, -(SP)
 MOV SP, R0
 TRAP C8PNTF

INITIALIZE SECTION

INITIALIZE SECTION

2280
2281
2282 006740 045 116 045 RUNG0: .ASCIZ /list DEX
0. 2283 007056 045 116 045 RUNG0: .ASCIZ //Only tests 1, 2 and 3 are active in this mode./
2284 007141 045 116 045 RUNG1: .ASCIZ //Running on unit #02SA in mode 0 : exec-time is 2 minutes on the PDP11 7
2285 007236 045 116 045 RUNG1b: .ASCIZ //Only tests 1 and 4 are active in this mode./
2286 007316 045 116 045 NOTAV: .ASCIZ //Unit #02SA is not available./
2287 007357 045 116 045 MGLMP1: .ASCIZ //Wrong number of D211 lines in bitmap "#03SA" for mode #01SA test./
2288 007467 045 116 045 MGLMP2: .ASCIZ //Change Hardware P-table to correct./
2289
2290
2291 .EVEN
2292
2293 007542
007542
007542 104411 ENDINIT
L10016:
TRAP C8INIT
2294
2295
2296
2297

AUTODROP SECTION

CE

2300

.SBTTL AUTODROP SECTION

2301

2302

2303

2304

2305

2306

2307

2308

2309 007544

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

007544

EVEN
BDAUTO

L: AUTO::

2310

2317

2318

2319

2320

; Check if DZ11 responds

2321

2322 007544 013701 002262

MOV DZCSR0, R1
MOV #4, RS ; 4 DZ11 registers to be tested

2323 007550 012705 000004

; Set up timeout trap :
 MOV #21, #04 ; address for timeout error trap handler
 MOV #MAXPRI, #06 ; priority level 7 in trap PSW to lock out
 ; other interrupts

2324

2325

2326 007554 012737 007606 000004

MOV #21, #04 ; address for timeout error trap handler
 MOV #MAXPRI, #06 ; priority level 7 in trap PSW to lock out
 ; other interrupts

2327 007562 012737 000340 000006

2328

2329

2330 007570 005711

18: TST (R1)
 NOP
 ADD #2, R1 ; next register
 DEC RS ; Decrement register count
 BNE 18 ; and branch back if not last register
 BR 38

2331 007572 000240

2332 007574 062701 000002

2333 007600 005305

2334 007602 01372

2335 007604 000405

2336

2337

2338 007606 062706 000004

28: ADD #4, SP ; Pop old PC, PSW
 DODU UUT ; Drop unit under test
 MOV UUT, R0
 TRAP C#DODU

2339 007612 013700 002216

007616 104451

2340

2341 007620 013737 002226 000004

38: MOV SAVE4, #04 ; Restore original timeout vector
 MOV SAVE6, #06

2342 007626 013737 002230 000006

2343

2344

2345

2346

2347 007634

007634

007634 104461

ENDAUTO

L10017:
 TRAP CIAUTO

2348

2349

2350

2351

**2354
2355
2356
2357
2358
2359
2360
2361 007636
007636**

.SBTTL CLEANUP CODING SECTION

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

BGNCLN

2362
2363
2363
2384
2385 007636
 007636
2386 007640
 007640
 007644
2387
2388 007646
 007646
 007646
2389
2390
2391
2392
2393

BRESET	: bus reset
TRAP	CORESET
CLRVEC	DZTVCC _a : Clear transmit interrupt vector
MOV	DZTVCC _a ,R0
TRAP	CSCVEC

L10020: ENOCLN TRAP C\$CLEAN

DROP UNIT SECTION

2396
2397
2398
2399
2400
2401
2402
2403 007650
 007650

2404

2405

2406

2415

2416

2417

2418

2419 007650 010046
 007650 012746 007676
 007652 012746 000002
 007656 012746 000000
 007662 010600
 007664 104417
 007666 062706 000006

2420

2421 007672
 007672 000167
 007674 000030

2422

2423

2435

2436

2437 007676 045 116 045 DROPO:
 007726
 007726 104453

2442

2443

2444

2445

2446

.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	#DROPO, R0	; DRS has put # of unit to be dropped in R0
MOV	R0,-(SP)	
MOV	#DROPO,-(SP)	
MOV	#2,-(SP)	
MOV	SP,R0	
TRAP	C\$PNTF	
ADD	#6,SP	

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

.nlist	BEX
.ASCIZ	/**/A unit #02%A dropped./
.list	BEX

.EVEN

ENDDU	
L10021:	TRAP C\$DU

ADD UNIT SECTION

```

2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468 007730          .SBttl ADD UNIT SECTION
2468 007730
2469
2470
2471 007730          // THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2471 007730          // TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
2471 007730          // "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
2471 007730          ///////////////////////////////////////////////////////////////////
2471 010046          L:AU::: BGNAU
2471 007732          .EVEN
2471 012746          PRINTF @ADDED, R0      ; DRS has put # of unit to be added in R0
2471 007736          MOV   R0,-(SP)
2471 012746          MOV   @ADDED,-(SP)
2471 000002          MOV   #2,-(SP)
2471 010600          MOV   SP,R0
2471 104417          TRAP  C$PNTF
2471 062706          ADD   #6,SP
2472
2473 007752          EXIT   DU
2473 000167          .WORD  J$JMP
2473 177750          .WORD  L10021-2-.
2474
2475
2476 007756          045    116    045    ADDED: .nlist  BEX
2476 010004          .ASCIZ /$RMAUnit #02%A added./
2477
2478
2479
2480 010004          .list   BEX
2480 010004          .EVEN
2480 104452          ENDAU
2481
L10022:           TRAP   C$AU

```

HARDWARE TESTS

2484
2485
2486
2487
2488
2489 010006
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536

.SBTTL HARDWARE TESTS

;START OF CODE BLOCK WHICH IS USED AS DATA
ROMMAP:...
; TEST TO ...
;
;
; BGNSTST
;
; EXIT TST
;
.EVEN
; ENDTST

HARDWARE TESTS

2539 010006

BADHEAD
 ;oooooooooooooo TEST1 ooooooooooooooo
 ;
 ; 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 :
 ; 00.1 - Subtest 1 : "Unsuccessful attempt to
 ; write to/read DZ11 CSR"
 ; "Check DZ11 address."
 ;
 ; 02 - Subtest 2 : "DZ11 failed to reset."
 ; "Check DZ11 address."
 ; "Run DZ11 diagnostic."
 ;
 ; 03 - Subtest 2 : "DZ11 internal loopback malfunction
 ; on line # <line number>"
 ; "Check DZ11 address."
 ; "Run DZ11 diagnostic."

2564 010006

BADHEAD
 ;oooooooooooooo TEST1 ooooooooooooooo

2565

2566 010006

010006

2567 010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

010006

01

HARDWARE TESTS

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

HARDWARE TESTS

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 #MSEMA1, SDZCSR0; Enable maintenance loopback mode transmission
 2665 010440 012701 177754 MOV #DLAY2s, R1 ; Set up 2 seconds delay
 2666
 2667 010444 005777 171612 nTRYD1: TST SDZCSR0 ; Wait
 2668 010450 100414 BMI contD7 ; for
 2669
 2670 010452 DLAY INC R1 ; CSR Transmit Ready bit
 2671 010472 005201 BNE nTRYD1 ; to set
 2672 010474 001363 JMP PBLMD1 ; If time-out
 2673 010476 000137 010610 ; then there's a problem
 2674
 2675 010502 112777 000101 171560 contD7: MOVB #'A, SDZTDR0 ; Load character into Transmit Data Register
 2676 010510 012701 177754 MOV #DLAY2s, R1 ; Set up 2 seconds delay
 2677
 2678 010514 105777 171542 nRDND1: TSTB SDZCSR0 ; REPEAT Wait
 2679 010520 100021 BPL contD8 ; UNTIL echo received
 2680
 2681 010522 017737 171536 002250 MOV SDZRBUFa, 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 SDZRBUFa, 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 INC R1 ; Wait
 2694 010604 005201 BNE nRDND1 ; routine
 2695 010606 001342
 2696
 2697 010610 000207 PBLMD1: RTS PC ; When delay is elapsed or a problem arises.
 2698 ; end sbTED1 ; return with sbAOK = 0
 2699
 2700
 2701 010612 ENDSUB
 010612 L10025:
 010612 104403 TRAP C8ESUB
 2702
 2703 010614 ENDTST
 010614 L10023:
 010614 104401 TRAP C8ETST
 .EVEN
 2704
 2705

HARDWARE TESTS

2708 C10616

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 BGNST
 010616
 2746 T2:: : 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 C8EXIT
 010626 000120 .WORD L10026-.
 2750
 2751 010630 004737 002572 contL1: JSR PC, sbIDG1 ; Initialize DZ11 for interrupt mode
 2752
 2753
 2754 : Test echo on line x :
 2755
 2756 010634 013777 002240 171422 MOV TLPRx, \$DZLPRx ; Load parameters for line x
 2757 010642 113777 002302 171416 MOVB LNMAPx, \$DZTCRx ; 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, \$DZLPRy
 2768 010676 113777 002304 171362 MOVB LNMAPy, \$DZTCRy
 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 ERR11: ERRMHD 4. NOWDEC, PNMDE; Report error
 010724 104456 TRAP C8ERRMHD
 010726 000004 .WORD 4
 010730 003112 .WORD NOWDEC
 010732 005332 .WORD PNMDE
 2779
 2780 010734 012777 000020 171320 endl1: MOV #CSRCLR, \$DZCSR ; Shut off DZ11
 2781 010742 ESCAPE TST
 010742 104410 TRAP C8ESCAPE
 010744 000002 .WORD L10026-.
 2782
 2783 010746 ENDTST
 010746 010746 L10026:
 010746 104401 TRAP C8ETST
 .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

;oooooooooooooo TEST3 ooooooooooooooo

;*
;* Test active only in mode 0 :
;*
;* Purpose : switch the GS03-HD 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-HD to line x ;
;* - Try and switch GS03-HD from line x to
;* line y ;
;* - Try and switch GS03-HD 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-HD 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-HD received on both lines
;* # <line number> and # <line number>."
;* #8 - "GS03-HD failed to switch to line # <line
;* number>"
;* "No echo received back from GS03-HD on line
;* # <line number>"
;* "Check FORCE, MANUAL switches, priority
;* setting and cables".
;* #9 - "Echo from the GS03-HD 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."

CZDZGAO GS340/DZ11 LGC DING

MACRO M1200 03-AUG-04 13:01 PAGE 97-1

B7

SEQ 0079

HARDWARE TESTS

2843

2844 010750

2845

10
BADHEAD

;oooooooooooooo TEST3 ooooooooooooooo

HARDWARE TESTS

2848 010750
 010750
 2849
 2850 010750 005737 002214
 2851 010754 001402
 2852 010756
 010756 104432
 010760 000522
 2853
 2854 010762 004737 002572
 2855
 2856
 2857 010766 013777 002240 171270
 2858 010774 013777 002242 171262
 2859
 2860 011002 005037 011500
 2861
 2862 011006 005037 002220
 2863
 2864 : Try and switch the GS03-MD to line x
 2865
 2866 011012 113777 002302 171246
 2867
 2868 011020 013737 002274 002310
 2869 011026 004737 002642
 2870
 2871 011032 005737 002306
 2872 011036 001012
 2873
 2874 011040 013737 002274 002272
 2875 011046
 011046 104456
 011050 000005
 011052 003066
 011054 005410
 2876 011056 012777 000020 171176
 2877
 2878 011064
 011064 104410
 011066 000414
 2879
 2880 : Try and switch the GS03-MD from line x to line y
 2881
 2882 011070 013737 002274 002312
 2883 011076 013737 002276 002316
 2884 011104 013737 002304 002314
 2885
 2886 011112 004757 011252
 2887
 2888 011116
 011116 104410
 011120 000362
 2889
 2890 : Try and switch the GS03-MD from line y to line x
 2891
 2892 011122 013737 002276 002312
 2893 011130 013737 002274 002316

BGNTST

T3::

; Initialization:

TST	TMODE	: If mode 1, then skip this test
BFO	contG1	
EXIT	TST	
TRAP	C EXIT	
.WORD	L10027-.	

contG1: JSR PC, #IDG1 : Initialize DZ11 for interrupt mode

; transmission

MOV TLPRx, #DZLPRx : Load parameters for line x

MOV TLPRy, #DZLPRy : Load parameters for line y

CLR FTIMG1

reptG1: CLR SWRTY : Set switch priority to none.

; Try and switch the GS03-MD to line x

MOV LNMAPx, #DZTCRx : Enable transmission on line x

MOV LNMBRx, LNTSTD : Transmit

JSR PC, #TEG1 : and test for echo on line x

TST #BAOK : If successful,

BNE contG2 : then go on testing

MOV LNMBRx, LNMBR : else report error

ERRMD 5, NOILEC, SMILE; and

TRAP C#ERRMD

.WORD 5

.WORD NOILEC

.WORD SMILE

MOV #CSRCLR, #DZCSR : shut off DZ11

contG2: ESCAPE TST

TRAP C#ESCAPE

.WORD L10027-.

; Try and switch the GS03-MD from line x to line y

MOV LNMBRx, OLDLNMB : Load

MOV LNMBRy, NEWLNMB : parameters

MOV LNMAPy, NEWLNMP :

JSR PC, #SMG1 : and test

ESCAPE TST

TRAP C#ESCAPE

.WORD L10027-.

; Try and switch the GS03-MD from line y to line x

MOV LNMBRy, OLDLNMB : Load

MOV LNMBRx, NEWLNMB : parameters

HARDWARE TESTS

2894 011136 013737 002302 002314 MOV LNMAPx, NEWLNMP ;
 2895
 2896 011144 004737 011252 JSR PC, #SMG1 ; and test
 2897
 2898 011150
 011150 104410
 011152 000330 ESCAPE TST
 TRAP C&ESCAPE
 .WORD L10027-.
 2899
 2900 011154 005737 011500 TST FTIMG1
 2901 011160 001027 ONE endG1
 2902
 2903 : Swap lines = and y and repeat this test :
 2904 011162 013737 002274 002272 MOV LNMAPx, LNMR ;
 2905 011170 013737 002276 002274 MOV LNMAPy, LNMRx ;
 2906 011176 013737 002272 002276 MOV LNMR, LNMAPy ;
 2907
 2908 011204 013737 002302 002272 MOV LNMAPx, LNMR ; LNMR is used as a temporary here
 2909 011212 013737 002304 002302 MOV LNMAPy, LNMAPx ;
 2910 011220 013737 002272 002304 MOV LNMR, LNMAPy ;
 2911
 2912 011226 012737 000001 011500 MOV #1, FTIMG1
 2913 011234 000137 011006 JMP reptG1
 2914
 2915 : End of test :
 2916 011240 012777 000020 171014 endG1: MOV #CSRCLR, #ZCSR#; Shut off DZ11
 2917 011246 EXIT TST
 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 NEHLNNB
 2921
 2922 : Assumption : line OLDLNNB is already alive.
 2923
 2924 011252 013777 002300 171006 abSMG1: MOV LNMAP, BDZTCR_a ; Enable transmission on both lines
 2925
 2926 011260 013737 002316 002310 MOV JSR NEWLNNB, LNTSTD ; Start transmitting
 2927 011266 004737 002642 PC, abTEG1 ; and test for echo on line NEWLNNB
 2928
 2929 011272 005737 002306 TST abAOK ; If echo on line NEWLNNB,
 2930 011276 001432 BEQ contG4 ;
 2931
 2932 011300 005737 002220 TST SWPRTY ; then begin line NEHLNNB has switch priority
 2933 011304 001406 BEQ contG3 ; so, if OLDLNNB already had it :
 2934
 2935 011306 104456 ERRH_D 6, PTYCFL, pPYCF; 'Both lines have switch priority...'
 011306 104456 TRAP C8ERH_D
 011310 000006 .WORD 6
 011312 003044 .WORD PTYCFL
 011314 005462 .WORD pPYCF
 2936
 2937 011316 ESCAPE TST
 011316 104410 TRAP C8ESCAPE
 011320 000162 .WORD L10027-.
 2938
 2939 011322 012757 000001 002220 contG3: MOV #1, SWPRTY ; else record that NEHLNNB has priority
 2940
 2941 011330 013737 002312 002310 MOV JSR OLDLNNB, LNTSTD ; If there is still echo
 2942 011336 004737 002642 PC, abTEG1 ; on OLDLNNB,
 2943 011342 005737 002306 TST abAOK ; that means there's echo on both lines :
 2944 011346 001406 BEQ contG4
 2945
 2946 011350 104456 ERRH_D 7, WD2ECH, pWD2E; 'Echo from GS03-WD received on both
 011350 104456 TRAP C8ERH_D
 011352 000007 .WORD 7
 011354 003021 .WORD WD2ECH
 011356 005530 .WORD pWD2E
 2947
 2948 011360 ESCAPE TST
 011360 104410 TRAP C8ESCAPE
 011362 000120 .WORD L10027-.
 2949 ; end
 2950
 2951 011364 013777 002314 170674 contG4: MOV NEHLNNP, BDZTCR_a; Stop transmitting on line OLDLNNB
 2952
 2953 011372 013737 002316 002310 MOV JSR NEWLNNB, LNTSTD ; If no echo
 2954 011400 004737 002642 PC, abTEG1 ; on line
 2955 011404 005737 002306 TST abAOK ; NEHLNNB,
 2956 011410 001011 BNE contG5 ; then :
 2957
 2958 011412 013737 002316 002272 MOV NEWLNNB, LNNBR ; 'GS03-WD fail to switch to line <NEHLNNB>'
 2959 011420 104456 ERRH_D 8, FAISWT, pFSWT; 'No echo received on line <NEHLNNB>'
 011420 104456 TRAP C8ERH_D
 011422 000010 .WORD 8
 011424 002777 .WORD FAISWT
 011426 005566 .WORD pFSWT
 2960 ; 'Check FORCE, MANUAL...'

HARDWARE TESTS

2961 011430		ESCAPE	TST	
011430	104410	TRAP	C&ESCAPE	
011432	000050	.WORD	L10027-.	
2962				
2963 011434	013737	002312	002310	contG3: MOV OLDLNMB, LNTSTD ; If there is
2964 011442	004737	002642		JSR PC, abTEG1 ; still echo
2965 011446	005737	002306		TST abACK ; on line
2966 011452	001411			BEQ abG1 ; OLDLNMB :
2967				
2968 011454	013737	002312	002272	MOV OLDLNMB, LNMBR ; 'GS03-WD fail to switch from line <OLDLNMB>'
2969 011462				ERRMRD 9, FAISWF, pFSWF ; 'Echo is still being received'
011462	104456	TRAP C&ERRMRD		
011464	000011	.WORD 9		
011466	002754	.WORD FAISWF		
011470	005664	.WORD pFSWF		
2970				: on line <OLDLNMB>'
2971				: 'Check FORCE, MANUAL...'
2972 011472		ESCAPE	TST	
011472	104410	TRAP	C&ESCAPE	
011474	000006	.WORD	L10027-.	
2973				
2974 011476	000207	abG1: RTS PC		
2975		; end abSHG1		

HARDWARE TESTS

2984
2985 011500 000000 FTIMG1: .WORD 0 ; Boolean value to flag first run through test
2986
2987
2988
2989 011502 ENDTST
011502 L10027:
011502 104401 TRAP CSETST
.EVEN
2990
2991

HARDWARE TESTS

2999 011504

BADHEAD
 ;oooooooooooooo TEST4 ooooooooooooooo
 ;
 ; Test active only in mode 1 :
 ;
 ; Purpose : installation test.
 ;
 ; Assumption : all previous tests ran successfully.
 ;
 ; Description :
 ; This test activates the line into the GS03-WD 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-WD should then turn on.
 ; The associated RED LED should turn off after one
 ; full GS03-WD 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.

3021 011504

BADHEAD
 ;oooooooooooooo TEST4 ooooooooooooooo

3022

3023 011504

011504

3024

011504 005737 002214
 3026 011510 001002
 3027 011512
 011512 104432
 011514 000042

BGNTST

T4::

; Initialization :
 TST TMODE ; If mode 0, then skip this test
 BNE contA1
 EXIT TST
 TRAP C\$EXIT
 .WORD L10030-.

3028

3029 011516 004737 002572

contA1: JSR PC, #IDG1 ; Initialize DZ11 for interrupt mode
 ; transmission

3031

3032

; Transmit on line x :

3033

3034 011522 042737 010000 002240

BIC #RCVRON, TLPRx ; Receiver will not be used

3035 011530 013777 002240 170526

MOV TLPRx, #DZLPRx ; Load parameters for line x

3036 011536 113777 002302 170522

MOVB LNMAPx, #DZTCRx ; Enable transmission on line x

3037

3038 011544 012777 040040 170510

MOV #MSETIE, #DZCSRx; Enable interrupt mode transmission

3039

011552 104422
 3041 011554 000776

loopA1: BREAK

TRAP C\$BRK
 BR loopA1

3042

011556 011556
 011556 104401

ENDTST

L10030:

TRAP C\$ETST
 .EVEN

3044

3045

HARDWARE PARAMETER CODING SECTION

3048

.SBTTL HARDWARE PARAMETER CODING SECTION

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

.nlist

BEX

/CSR/

3080 011646 126 105 103 VECTOR: .ASCIZ

/VECTOR/

3081 011655 102 122 000 PRIORITY: .ASCIZ

/BR/

3082 011660 101 103 124 ACLINES: .ASCIZ

/ACTIVE LINES/

3083 011675 127 110 111 WCHMODE: .ASCIZ

/WHICH MODE/

3084

.1list

BEX

HARDWARE PARAMETER CODING SECTION

3085

.EVEN

3086

3087 011710

ENDHWD

.EVEN

011710

L10031:

3088

3089

3090

3091

3092

3093

SOFTWARE PARAMETER CODING SECTION

3095 .sbttl SOFTWARE PARAMETER CODING SECTION
3096
3097
3098 // THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
3099 // THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
3100 // MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
3101 // INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
3102 // MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
3103 // WITH THE OPERATOR.
3104 //
3105
3106
3107 011710 BGNNSFT
011710 000000 .WORD L10032-L\$SOFT/2
011712 L\$SOFT:::
3108
3109
3110
3111
3112 011712 ENDSFT
011712 .EVEN
L10032:
3120
3121
3122
3123

SOFTWARE PARAMETER CODING SECTION

3131
3132 011712
3133 011712
3140
3141 012072

012072 012114
012074 000007
012076
3142 012076
3143
3144

\$PATCH:::
.BLKW 70

LASTAD
.EVEN
.WORD T\$FREE
.WORD T\$SIZE
L\$LAST:::
ENDMOD

SOFTWARE PARAMETER CODING SECTION

3146
3147
3160
3161 012076
3162 012076 BGNSETUP 1
 012076 BGNPTAB
 012100 .WORD 0
 012102 .WORD L10035-. /2-1
 012102
3163
3164 012102 160100 .word 160100
3165 012104 000300 .word 300
3166 012106 000005 .word 5
3167 012110 000003 .word 3
3168 012112 000000 .word 0
3169
3170 012114 ENOPTAB
 012114
3171 0'2114 L10035:
3172
3173
3174
3175
3176
3177 000001 ENOSSETUP
 .END

SYMBOL TABLE

ABORI1	006500	C\$PCLR= 000020 G	DIAGMC= 000000	F\$PWR = 000017	L\$NBR 002272
ACLINE	011660	CSRR 004213	DLAYAR= 000100 G	F\$RPT = 000012	L\$NBRX 002274
ADDED	007756	CSRW 004116	DLAYC1 002244	F\$SEG = 000003	L\$NBRY 002276
ADDR	002320	C\$AU = 000052	DLAYC2 002246	F\$SOFT = 000005	L\$NTSTD 002310
ADR	- 000020 G	C\$AUTO= 000061	DLAY11= 177622 G	F\$SRV = 000010	LOE = 040000 G
ASSEM	- 000010	C\$BK = 000022	DLAY2S= 177754 G	F\$SUB = 000002	LOOPA1 011552
BIT0	- 000001 G	C\$BSEG= 000004	DROP0 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	DZLBR 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\$OCLN= 000044	DZPTY 002234	G\$LOLI= 000001	L\$CLEA 007636 G
BIT08	- 000400 G	C\$ODDU= 000051	DZBUF 002264	G\$NO = 000000	L\$CO 002032 G
BIT09	- 001000 G	C\$ORPT= 000024	DZRVCC 002252	G\$OFFS= 000400	L\$DEPO 002011 G
BIT1	- 000002 G	C\$DU = 000053	DZRVCS 002254	G\$OFSI= 000376	L\$DESC 002156 G
BIT10	- 002000 G	C\$EDIT= 000003	DZTCRA 002266	G\$PRMA= 000001	L\$DESP 002076 G
BIT11	- 004000 G	C\$ERDF= 000055	DZTDRA 002270	G\$PRMD= 000002	L\$DEV 002060 G
BIT12	- 010000 G	C\$ERMR= 000056	DZTVCC 002256	G\$PRML= 000000	L\$DISP 002132 G
BIT13	- 020000 G	C\$ERR0= 000060	DZTVCS 002260	G\$RADA= 000140	L\$DLY 002116 G
BIT14	- 040000 G	C\$ERSF= 000054	ECHO 002250	G\$RADB= 000000	L\$DTDP 002040 G
BIT15	- 100000 G	C\$ERSO= 000057	EF.CON= 000036 G	G\$RADB= 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	I\$BE = 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\$EXPS 002066 G
BUSTIM	003214	C\$GPL0= 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
CXDZAD	004775	C\$INLP= 000020	E\$LOAD= 000035	I\$AUTO= 000041	L\$HTPT 002022 G
CXFMSW	004523	C\$MANI= 000050	FAISWF 002754	I\$CLN = 000041	L\$HW 002144 G
CXGSCF	004706	C\$MEM = 000031	FAISWT 002777	I\$DU = 000041	L\$ICP 002104 G
CONTA1	011516	C\$MSG = 000023	FSWF 003230	I\$HDL = 000041	L\$INIT 005774 G
CONTD1	010104	C\$OPEN= 000034	FSWT 003343	I\$INIT= 000041	L\$LDAP 002026 G
CONTD2	010172	C\$PNTB= 000014	FTIME 002212	I\$MOD = 000041	L\$LAST 012076 G
CONTD3	010274	C\$PNTF= 000017	FTIMG1 011500	I\$MSG = 000041	L\$LOAD 002100 G
CONTD4	010342	C\$PNTS= 000016	FTUNI1 006114	I\$PROT= 000040	L\$LUN 002074 G
CONTD5	010354	C\$PNTX= 000015	F\$AU = 000015	I\$PTAB= 000041	L\$MREV 002050 G
CONTD6	010406	C\$QIO = 000377	F\$AUTO= 000020	I\$PWR = 000041	L\$NAME 002000 G
CONTD7	010502	C\$RDBU= 000007	F\$BGN = 000040	I\$RPT = 000041	L\$PRI0 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	L\$MAP = 002300	L\$SPCP 002020 G
CONTL1	010630	C\$TPRI= 000013	F\$MSG = 000011	L\$MAPX = 002302	L\$SPTP 002024 G
CSR	011642	DFPTBL 002144 G	F\$PROT= 000021	L\$MAPY = 002304	L\$STA 002030 G

SYMBOL TABLE

L\$TEST	002114 G	NEWNN	002316	PRI04	- 000200 G	TLPTR	002236	T\$SDAT	- 010035
L\$TML	002014 G	NFTMII	006024	PRI05	- 000240 G	TMAYER	002472	T\$DU	- 010021
L\$UNLT	002012 G	NOECIL	004304	PRI06	- 000300 G	TMODE	002214	T\$HAR	- 010031
L10001	002156	NOTAV	007316	PRI07	- 000340 G	TXPSW	002232	T\$HM	- 010001
L10002	002570	NONDEC	003112	PTYCFL	003044	T\$RAGC	- 000002	T\$IMI	- 010016
L10003	005076	NOILEC	003066	PWD2E	005530 G	T\$CODE	024004	T\$MSG	- 010014
L10004	005150	NROND1	010514	PYCF	005515	T\$ERRN	- 000011	T\$PC	- 000001
L10005	005236	NRONG1	002640	RBUFCT	- 174000 G	T\$EXCP	- 000000	T\$PRO	- 010000
L10006	005330	NTRYD1	010444	RCVRON	- 010000 G	T\$FLAG	- 000041	T\$PTA	- 010034
L10007	005406	NUOE	003667	REPTG1	011006	T\$FREE	- 012114	T\$RPT	- 010015
L10010	005460	NUBIII	006536	RLMPI1	006534	T\$GMAN	- 000000	T\$SEG	- 010001
L10011	005526	NUUMII	006122	RLMPI2	006676	T\$HILI	- 000001	T\$.SUP	- 010032
L10012	005564	NULE	003602	ROMMAP	010006	T\$LAST	- 000001	T\$SRV	- 010002
L10013	005662	OLDNN	002312	RUNG0A	006740	T\$LOLI	- 000000	T\$SUB	- 010025
L10014	005770	OIAPTS	- 000000	RUNG0B	007056	T\$LSYM	- 010000	T\$TES	- 010030
L10015	005772	OIAU	- 000001	RUNG1A	007141	T\$LTAD	- 000004	T1	- 010006 G
L10016	007542	OIPXR	- 000000	RUNG1B	007236	T\$NEST	- 177777	T1.1	- 010006
L10017	007634	OIGNS	- 000000	SAVE4	002226	T\$NS0	- 000000	T1.2	- 010212
L10020	007646	OIOU	- 000001	SAVE6	002230	T\$NS1	- 000005	T2	- 010616 G
L10021	007726	OISERR	- 000000	SBAOK	002306	T\$NS2	- 000002	T3	- 010750 G
L10022	010004	OIQNSM	- 000000	SBIDG1	002372	T\$NS3	- 000003	T4	- 011504 G
L10023	010614	OIPQIM	- 000001	SBLNII	006506	T\$PCNT	- 000000	UAM	- 000200 G
L10024	010210	OIPSETU	- 000001	SRSMG1	011252	T\$PTAB	- 010034	UNIT	- 002322
L10025	010612	POLM01	010610	SBTED1	010426	T\$PTHV	- 000001	UUT	- 002216
L10026	010746	PCSRR	005026 G	SBTEG1	002642	T\$PTHU	- 000001	VECTOR	- 011646
L10027	011502	PCSRW	005100 G	SBWTG1	002342	T\$SAVL	- 177777	WAITG1	- 0C2716
L10030	011556	PDZIN	005152 G	SBWTG2	002362	T\$SEGL	- 177777	WCWMO0	- 011675
L10031	011710	PDZL0	005240 G	STECL1	004371	T\$SEXO	- 010001	MD2E	- 003416
L10032	011712	PFSMF	005664 G	SUC061	002744	T\$SIZE	- 000007	MD2ECH	- 003021
L10033	012102	PFSMT	005766 G	SUC11	006730	T\$SUBN	- 000000	MGLMP1	- 007357
L10035	012114	PNT	- 001000 G	SVC06L	- 000000	T\$TAGL	- 177777	MGLMP2	- 007467
MAXERR	002222	PNADE	005332 G	SVCINS	- 000000	T\$TAGN	- 010036	X\$ALMA	- 000000
MAXPRI	- 000340 G	PNILE	005410 G	SVCJSB	- 000000	T\$TEMP	- 000000	X\$FALS	- 000040
M0D1I1	006372	PPYCF	005462 G	SVCTAG	- 000000	T\$TEST	- 000004	X\$OFFS	- 000400
MSEMAM	- 000050 G	PRI	- 002000 G	SVCST	- 000000	T\$TSTM	- 177777	X\$TRUE	- 000020
MSETIE	- 040040 G	PRIORT	011655	SVTXG1	002362 G	T\$STS	- 000001	ZDZGAO	- 002000 G
MCLD01	010226	PRI00	- 000000 G	SPRTY	002220	T\$IAU	- 010022	ILSTIN	- 000000
MCLDG1	002626	PRI01	- 000040 G	SLSYM	- 010000	T\$IAUT	- 010017	ILSTTA	- 000000
METYG1	002666	PRI02	- 000100 G	TLPTR	002240	T\$ICLE	- 010020	IPATCH	- 011712 G
NEWNN	002314	PRI03	- 000140 G	TLPTR	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-[PINASA.D.GS.PDP.SSOC]LIBA.MLB/ML,ZDZGAO