

PDP11

INTER TEST PROGRAM
MD-11-DZITA-C

EP-DZITA-C-DL-A

COPYRIGHT © 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZITA-C-D
PRODUCT NAME: INTERPROCESSOR TEST PROGRAM (ITEP)
PROGRAM DATE: OCTOBER 1976
MAINTAINER: DIAGNOSTICS

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1974, 1975, 1976, BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR FIELD SERVICE PERSONEL. IT WILL VERIFY THE PROPER OPERATION OF A COMPLETE COMMUNICATION LINK FROM ONE PDP-11 SYSTEM TO ANOTHER OR TO A COMMUNICATION TEST CENTER.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

A. PDP-11 SYSTEM WITH AT LEAST 4K OF CORE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE

THIS PROGRAM AND ALL OVERLAYS ARE ASSEMBLED IN ABSOLUTE FORMATS. THE ABS LOADER IS USED TO LOAD THE PROGRAM AND OVERLAYS.

LOAD THE ITEP PROGRAM AND THE APPROPRIATE OVERLAY FOR THE TYPE OF INTERFACE YOU WISH TO TEST.

4.0 OPERATING PROCEDURES.

A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED

1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
 2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.
- *THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

B. CONSOLE DIALOGUE PARAMETER INPUT (CURRENT VALUES FOR PARAMETERS ARE FOUND IN OVERLAY)
DN11 AND DN11BB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THIS LISTING.

1. THE PROGRAM WILL TYPEOUT THE NAME OF THE VARIABLE OVERLAY.
 - A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A CARAGE RETURN
 - B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
 - C. IF YOU WISH TO SETUP A DN11BB, TYPE IN DMB.

IF DN OR DMB WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS, VECTOR, ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DNBB.

2. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
 - A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
 - B. TYPEIN ACTUAL BUS ADDRESS
3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
 - A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS
 - B. TYPEIN ACTUAL VECTOR ADDRESS
4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY

NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.

 - A. TYPE A CAR. RETURN TO USE DEFAULT VALUE

- B. TYPEIN ACTUAL VALUE
5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE OVERLAY. (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPEIN ACTUAL VALUE
6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE OVERLAY.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. ENTER ACTUAL VALUE
7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
THE DN-11 WILL USE PARAM #3 AS THE # TO DIAL.
IF USING A MODEM WITHOUT AUTOMATIC HANDSHAKING,
THE NUMBER MUST TERMINATE WITH A
"END-OF-NUMBER" CHARACTER (:).
B. ENTER ACTUAL VALUE.
8. THE PROGRAM WILL RETURN TO STEP B1 IF THIS SETUP
WAS FOR DN11 OR DN11BB.
9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.
- NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING
NEW VALUES, THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT
RESTARTS OF THE PROGRAM.

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR (INTERFACE SERVICE ROUTINE) SPECIFICATION
 SWR14=1 SETUP DN-11BB ISR
 SWR13=1 SETUP DN-11 ISR
 SWR=000000=SETUP VARIABLE ISR (OVERLAY) (NOT DN-11 OR DN11BB)
 SET APPROPRIATE SWITCHES AND HIT CONTINUE.
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
 SETUP SEQUENCE IS: DN11, DN11-BB THEN VARIABLE ISR. (FOR EACH ENTRY SET SWITCHES AND THEN HIT CONT.)
 - A. HALT FOR BUS ADDRESS OF INTERFACE
 - B. HALT FOR VECTOR ADDRESS OF INTERFACE
 - C. HALT FOR PRIORITY OF INTERFACE (200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.)
 - D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - E. HALT FOR INTERFACE PARAM #2 (DN11 AND DNBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THIS LISTING)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DNBB.
 3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
 - A. PRESS CONTINUE TO START TESTING

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. (CR) IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY (CR).
3. ↑U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR

SW14=1 SINGLE PASS

SW14 HAS NO EFFECT IF SW04=0

SW13=1 INHIBIT ERROR TYPEOUTS

SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS

IF SW12=0 AND SW04=1 END PASS IS TYPED

AND TRANSMITTED/RECEIVED DATA IS TYPED.

SW11=1 USE PREVIOUSLY SPECIFIED DATA

SW10=1 DATA SELECT (WITH SW09)

SW09=1 DATA SELECT (WITH SW10)

00=1 GET DATA FROM OPERATOR

01=1 TEST MESSAGE #1 (\$A QUICK BROWN FOX)

10=1 TEST MESSAGE #2 (\$B NUMERICS)

11=1 TEST MESSAGE #3 (\$C COMTEST/QUICK BROWN FOX/NUMERICS)

SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)

SW07=1 DO NOT TEST RECEIVED DATA

SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*

SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*

* IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.

SW04=1 RETURN TO MONITOR FOR END PASS

WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.

SW03=1 INTERNAL LOOPBACK MODE

SW02=1 EXTERNAL LOOPBACK MODE

SW01=1 ONE-WAY-IN MODE

SW00=1 ONE-WAY-OUT MODE

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE. TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN ↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377) SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).
I.E. ABCD↑ 000 123 377↑ EFG (CAR.RETURN)

A TYPICAL SWITCH SETTING FOR HALF-DUPLEX=003150 THIS SETTING USES INTERNAL LOOPBACK MODE, LOOPS IN OVERLAY, MONITORS TRANSMITTED AND RECEIVED DATA ON THE CONSOLE TTY, AND TESTS RECEIVED DATA USING TEST MESSAGE #3.

A TYPICAL SWITCH SETTING FOR FULL-DUPLEX=003144 THIS SETTING IS THE SAME AS ABOVE EXCEPT IT USES THE EXTERNAL LOOPBACK MODE.

ALL STANDARD MESSAGES (TEST MESSAGES 1-3) ARE PRECEDED BY 2 FILL CHARACTERS(177), AND ARE FOLLOWED BY A CR(015), LF(012), RECEIVE TERMINATING CHARACTER(001), 4 FILLS(177), AND A TRANSMIT TERMINATING CHARACTER(000). DURING TRANSMISSION, WHEN A 000 CHARACTER IS SEEN THE TRANSMISSION IS STOPPED. DURING RECEPTION, WHEN A 001 CHARACTER IS RECEIVED, THE RECEIVER IS SHUT OFF. IF THE MESSAGE WAS INPUTED BY THE OPERATER, THE TERMINATING CHARACTERS ARE ADDED.

G01

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10 (SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) (").
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR GO TO STEP 1 (SW04=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA (SW07=0)
3. RETURNS TO MONITOR FOR "END PASS" (SW04=1) OR GO TO STEP 1 (SW04=0)

ONE-WAY-OUT MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAITS FOR CLEAR TO SEND
3. TRANSMITS SELECTED DATA
4. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR GO TO STEP 1 (SW04=0)

E. THE OVERLAY IS THEN ENTERED AND A CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY.

IF ONE-WAY-IN OR INTERNAL LOOPBACK MODES ARE SELECTED.
THE OVERLAY WILL SET DATA TERMINAL READY AND WAIT FOR DATA.

IF ONE-WAY-OUT OR EXTERNAL LOOPBACK MODES WERE SELECTED.
THE OVERLAY WILL SET DATA TERMINAL READY AND REQUEST TO SEND.
THE OVERLAY WILL THEN WAIT FOR CLEAR TO SEND BEFORE ATTEMPTING TO TRANSMIT DATA.

F. IF SW04=0 THE OVERLAY WILL CONTINUE TO TRANSMIT/RECEIVE DATA.

IF SW04=1 THE OVERLAY WILL RETURN TO THE MONITOR AND TYPE "END PASS".

IF BOTH SW04=1 AND SW14=1, THE PROGRAM WILL REQUEST NEW INTERFACE PARAMS AFTER ONE PASS OF THE SELECTED TEST MODE.

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING CHARACTERS ON THE CONSOLE TTY.

LINE FEED = RESTART PROGRAM AT LOCATION 200.

QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)
THEN TYPE EITHER:

#DXXXXXX TO PRINTOUT THE 8 WORDS
AT LOC XXXXXX.

#BXXXXXX TO PRINTOUT THE 16 BYTES
AFTER LOC XXXXXX.

#C TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.
CARRIAGE RETURN = RESTART AT REQUEST FOR NEW OPERATIONAL SWITCHES.

5.0 PROGRAM AND/OR OPERATOR ACTION

IF THE OPERATOR WISHES TO MANUALLY EXAMINE THE TRANSMIT OR RECEIVE BUFFERS, DO THE FOLLOWING: TO FIND THE STARTING ADDRESS OF THE RECEIVE BUFFER, LOAD ADDRESS 11020 AND EXAMINE. TO FIND THE STARTING ADDRESS OF THE TRANSMIT BUFFER, LOAD ADDRESS 11022 AND EXAMINE.

5.1 NORMAL HALTS SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

THE ONLY ERROR REPORT FROM THE CONTROL PROGRAM OCCURS IF THE INTERFACE SPECIFIED IS NOT LOADED.

THE ERROR REPORTS FROM THE VARIOUS INTERFACE SERVICE ROUTINES ARE AS DEFINED IN THEIR DOCUMENTS

7.0 RESTRICTIONS

THE OPERATION OF THIS PROGRAM REQUIRES COORDINATION BETWEEN THE OPERATOR AND THE OPERATOR OF ANOTHER PDP-11 SYSTEM UNLESS ONE OF THE SYSTEMS IS ALWAYS OPERATING IN A FIXED MODE. THE FOLLOWING TABLE LISTS THE VALID COMBINATIONS:

NOTE: ONLY ONE MODE MAY BE SELECTED AT A TIME.

CPU #1	CPU #2
ONE-WAY-OUT	ONE-WAY-IN
ONE-WAY-IN	ONE-WAY-OUT
EXTERNAL-LOOPBACK	INTERNAL-LOOPBACK
INTERNAL-LOOPBACK	EXTERNAL-LOOPBACK
EXTERNAL-LOOPBACK	EXTERNAL-LOOPBACK (FULL-DUPLEX)

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTIONS APPLY:

IF RUNNING IN FULL DUPLEX MODE BOTH SYSTEMS
MUST BE IN EXTERNAL LOOP BACK MODE.

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.

EXAMPLE:

SWITCHES 14,13,7,4 SHOULD BE THE SAME
ON BOTH CPU'S

J01

8.0 MISCELLANEOUS

ITP WAS CHECKED OUT USING THE FOLLOWING BELL TELEPHONE MODEMS.
201A (HALF-DUPLEX SYNCHRONOUS 2000 BAUD)
202C (HALF-DUPLEX ASYNCHRONOUS 1200 BAUD)
103A (FULL-DUPLEX ASYNCHRONOUS 110 BAUD)

9.0 PROGRAM DESCRIPTION

THE INTERPROCESSOR TEST PROGRAM (ITP) PROVIDES THE LINKAGE BETWEEN THE OPERATOR AND THE VARIOUS INTERFACE SERVICE ROUTINES (OVERLAY) WHICH PERFORM THE ACTUAL DATA MOVEMENT AND VERIFICATION TO AND FROM THE COMMUNICATION LINK. IN ADDITION, ITP CONTAINS VARIOUS INTERRUPT AND SUB ROUTINES WHICH ARE USED BY THE OVERLAY'S.

9.1 TRAP CATCHER

THIS IS A SERIES OF JUMP AND HALT INSTRUCTIONS PLACED IN ALL UNUSED VECTORS TO CATCH UNEXPECTED INTERRUPTS.

9.2 SWITCH REGISTER INPUT ROUTINE (MANIN:)

THIS ROUTINE IS ENTERED ONLY WHEN SWITCH 15 IS SET WHEN PROGRAM IS STARTED AT LOCATION 200. IT ACCEPTS PARAMETERS FOR THE ISR'S FROM THE CONSOLE SWITCHES AT A SERIES OF HALTS. AS SPECIFIED IN OPERATING INSTRUCTIONS.

9.3 PARAMETER INPUT ROUTINE (GETIT:)

THIS ROUTINE SOLICITS PARAMETERS FROM THE OPERATOR ON THE CONSOLE DEVICE AND PLACES THEM IN THE SPECIFIED ISR'S PARAMETER TABLE.
NOT USED OPTIONAL PARAMETER WORDS ARE INDICATED BY THE PRESENCE OF A NEGATIVE VALUE IN THE ISR'S TABLE. THIS SECTION OF CODE UTILIZES SUB/ROUTINE 'GETANY' WHICH PRINTS OUT THE WORD POINTED TO BY THE ADDRESS IN REGISTER 0.
IT THEN INPUTS A WORD OR CARRIAGE RETURN FROM THE OPERATOR. IF ONLY A CARRIAGE RETURN IS TYPED, THE PARAMETER IS LEFT AS IT IS, OTHERWISE IT IS REPLACED BY THE OPERATORS TYPE IN AND THE POINTER IN REGISTER 0 IS INCREMENTED TO THE NEXT WORD.

9.4 TTY INTERRUPT (TTYINT:)

THE TTY INTERRUPT IS USED TO INTERRUPT THE EXECUTION OF A TEST IN ORDER TO RESTART (TYPE A LINE FEED) OR TO SPECIFY NEW OPERATIONAL SWITCHES (TYPE A CARRIAGE RETURN)

9.5 SET SWITCH OPTIONS (SWRSET:)

THE PROGRAM WILL HALT (MANUAL PARAMETER ENTRY) OR WAIT FOR A CARRIAGE RETURN (TTY CONTROL) AT THIS POINT TO PERMIT THE OPERATOR TO SETUP THE OPERATIONAL SWITCH SETTINGS. THE TEST MODE(SW00-SW03) AND TEST DATA(SW08-SW11) MAY BE CHANGED ONLY AT THIS POINT. ALL OTHER SWITCHES MAY BE CHANGED WHILE A TEST IS RUNNING. IF NEW VARIABLE DATA IS SPECIFIED, THIS ROUTINE WILL REQUEST THAT THE DATA BE ENTERED AND UTILIZES THE 'GETSTR' SUB/ROUTINE TO INPUT THE DATA FROM THE OPERATOR.

9.6 SETUP TIMER (SUTIME:)

THE PROGRAM LOOKS FOR AND UTILIZES EITHER THE LINE CLOCK OR REAL TIME CLOCK IF EITHER IS PRESENT ON THE SYSTEM. A BUS ERROR(NO RESPONSE) IS USED TO INDICATE THE ABSENCE OF A CLOCK. IF NEITHER EXISTS, THE PROGRAM WILL STILL RUN BUT IS SUBJECT TO WAITING IN UNENDING LOOPS.

9.7 THE INTERFACE SERVICE ROUTINES (ISR'S) ARE ENTERED AT THIS POINT.

9.8 END OF PASS (SEOP:)

THIS SECTION OF CODE WILL PRINT "END OF PASS XXXXXX" AND THEN SENSE FOR SW14. IF SWITCH 14 IS RESET THE OVERLAY'S ARE REENTERED. IF SWITCH 14 IS SET THE PROGRAM CHECKS TO SEE IF IT WAS LOADED BY A MONITOR (LOCATION 42 NOT EQUAL 0) AND IF IT WAS, CONTROL IS RETURNED TO THE MONITOR. OTHERWISE THE PROGRAM REQUESTS NEW PARAMETERS.

9.10 HALT HANDLER (SHLT:)

THIS ROUTINE IS USED TO SENSE THE OPERATIONAL SWITCHES AND PROVIDE ERROR CONTROL. IT WILL PRINTOUT THE ADDRESS OF THE ERROR HLT IF SWITCH 13 (DELETE ERROR TIMEOUTS) IS DOWN (NOT SET)

L01

- 9.11 READ A CHARACTER ROUTINE (\$READC:)
THIS ROUTINE GETS A CHARACTER FROM THE TTY AND PLACES IT ON THE STACK
- 9.12 READ A STRING ROUTINE (\$READS)
THIS ROUTINE GETS A STRING OF CHARACTERS FROM THE TTY AND PLACES THEM IN A BUFFER SPECIFIED BY THE ADDRESS FOLLOWING THE SUB/ROUTINE CALL.
THE ROUTINE WILL ALSO ACCEPT OCTALLY REPRESENTED CHARACTERS WHEN THEY ARE PRECEDED AND FOLLOWED BY UP ARROWS, AND SPERATED BY SPACES OR COMMAS.
- 9.13 OCTAL INPUT ROUTINE(\$ACCEPT:)
THIS ROUTINE READS AN OCTALLY REPRESENTED WORD FROM THE TTY AND PLACES IT IN THE LOCATION INDICATED BY THE ADDRESS FOLLOWING THE SUB/ROUTINE CALL.
- 9.14 CLOCK INTERRUPT ROUTINE (TIMER:)
THIS ROUTINE IS ENTERED ON INTERUPTS FROM EITHER THE LINE CLOCK OR REAL TIME CLOCK EVERY 16 MILLISECONDS IF EITHER IS PRESENT.
IT WILL INCREMENT LOCATION 'TIME:' IN THE OVERLAY'S PARAMETER TABLE EVERY SECOND.
- 9.15 BINARY TO OCTAL ROUTINE (\$B2O16)
THIS ROUTINE WILL PRINTOUT THE OCTAL REPRESENTATION OF A WORD ON THE STACK.
- 9.16 POWER DOWN ROUTINE (\$PWRDN:)
THIS ROUTINE SAVES THE STATUS OF THE MACHINE WHEN POWER IS LOST.
- 9.17 POWER UP ROUTINE (\$PWRUP:)
THIS ROUTINE RESTORES THE STATE OF THE MACHINE WHEN POWER IS RESTORED AND RESTARTS AT ADDRESS 200.
- 9.18 VARIABLE INTERFACE SERVICE ROUTINE (VISR:)
THESE LOCATIONS ARE RESERVED FOR AND WILL BE OVERLAID BY THE VARIABLE ISR'S.
THE FIRST 2 WORDS CONTAIN A 3 CHARACTER ISR NEMONIC FOLLOWED BY A ZERO CHARACTER.
THE NEXT 3 WORDS CONTAIN THE BUS ADDRESS, VECTOR ADDRESS AND PRIORITY.
THE NEXT 2 WORDS MAY CONTAIN OPTIONAL PARAMETERS. THEY WILL CONTAIN ALL ONES IF THEY ARE NOT REQUIRED
THE NEXT WORD MAY CONTAIN THE ADDRESS OF AN INPUT BUFFER IF THE ISR REQUIRES AN ASCII PARAMETER. IT WILL CONTAIN

ALL ONES IF THE PARAMETER IS NOT REQUIRED.
LOCATION 'CLOCK:' WILL BE INCREMENTED EVERY SECOND WHILE
THE TEST IS BEING RUN IF THERE IS A LINE CLOCK OR REAL TIME CLOCK
ON THE SYSTEM. IT MAY BE USED AS A ELAPSED TIMER BY THE ISR.

10.0 PARAMETERS FOR THE DM11BB AND THE DN11

10.1 DM11BB PARAMETERS

PARAM#1 IS LOADED INTO THE CONTROL AND STATUS REGISTER OF THE DM11BB
TO SELECT THE LINE NUMBER IN OCTAL (BITS 0-3). ALL OTHER BITS MUST BE 0'S.
THIS IS THE ONLY PARAMETER USED BY THE DM11BB.

10.2 DN11 PARAMETERS

ONLY PARAM#3 IS USED BY THE DN11, IT CONTAINS THE NUMBER THE DN WILL DIAL.

NO1

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 13
DZITAC.P11 04-AUG-76 11:27

```

;BASIC DEFINITIONS
;*****
;INITIAL ADDRESS OF THE STACK POINTER
STACK= 1070

```

```

;*****
;EQUIV ENT.MLT

```

```

;BASIC DEFINITION OF ERROR CALL

```

```

;REGISTER DEFINITION

```

```

R0= X0
R1= X1
R2= X2
R3= X3
R4= X4
R5= X5
R6= X6
R7= X7
MODE= X0
R6=SP
R7=PC

```

```

;GENERAL REGISTER
;GENERAL REGISTER
;GENERAL REGISTER
;GENERAL REGISTER
;GENERAL REGISTER
;GENERAL REGISTER
;GENERAL REGISTER
;GENERAL REGISTER

```

```

;SWITCH DEFINITION

```

```

SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW9= 1000
SW8= 400
SW7= 200
SW6= 100
SW5= 40
SW4= 20
SW3= 10
SW2= 4
SW1= 2
SW0= 1
;EQUIV SW39,SW9
;EQUIV SW38,SW8
;EQUIV SW37,SW7
;EQUIV SW36,SW6
;EQUIV SW35,SW5
;EQUIV SW34,SW4
;EQUIV SW33,SW3
;EQUIV SW32,SW2
;EQUIV SW31,SW1
;EQUIV SW30,SW0

```

```

PRTY0= 0
PRTY1= 40
PRTY2= 100
PRTY3= 140
PRTY4= 200
PRTY5= 240

```

```

001070

```

```

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200
000201
000202
000203
000204
000205
000206
000207
000208
000209
000210
000211
000212
000213
000214
000215
000216
000217
000218
000219
000220
000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275
000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330
000331
000332
000333
000334
000335
000336
000337
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365
000366
000367
000368
000369
000370
000371
000372
000373
000374
000375
000376
000377
000378
000379
000380
000381
000382
000383
000384
000385
000386
000387
000388
000389
000390
000391
000392
000393
000394
000395
000396
000397
000398
000399
000400
000401
000402
000403
000404
000405
000406
000407
000408
000409
000410
000411
000412
000413
000414
000415
000416
000417
000418
000419
000420
000421
000422
000423
000424
000425
000426
000427
000428
000429
000430
000431
000432
000433
000434
000435
000436
000437
000438
000439
000440
000441
000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453
000454
000455
000456
000457
000458
000459
000460
000461
000462
000463
000464
000465
000466
000467
000468
000469
000470
000471
000472
000473
000474
000475
000476
000477
000478
000479
000480
000481
000482
000483
000484
000485
000486
000487
000488
000489
000490
000491
000492
000493
000494
000495
000496
000497
000498
000499
000500
000501
000502
000503
000504
000505
000506
000507
000508
000509
000510
000511
000512
000513
000514
000515
000516
000517
000518
000519
000520
000521
000522
000523
000524
000525
000526
000527
000528
000529
000530
000531
000532
000533
000534
000535
000536
000537
000538
000539
000540
000541
000542
000543
000544
000545
000546
000547
000548
000549
000550
000551
000552
000553
000554
000555
000556
000557
000558
000559
000560
000561
000562
000563
000564
000565
000566
000567
000568
000569
000570
000571
000572
000573
000574
000575
000576
000577
000578
000579
000580
000581
000582
000583
000584
000585
000586
000587
000588
000589
000590
000591
000592
000593
000594
000595
000596
000597
000598
000599
000600
000601
000602
000603
000604
000605
000606
000607
000608
000609
000610
000611
000612
000613
000614
000615
000616
000617
000618
000619
000620
000621
000622
000623
000624
000625
000626
000627
000628
000629
000630
000631
000632
000633
000634
000635
000636
000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
000648
000649
000650
000651
000652
000653
000654
000655
000656
000657
000658
000659
000660
000661
000662
000663
000664
000665
000666
000667
000668
000669
000670
000671
000672
000673
000674
000675
000676
000677
000678
000679
000680
000681
000682
000683
000684
000685
000686
000687
000688
000689
000690
000691
000692
000693
000694
000695
000696
000697
000698
000699
000700
000701
000702
000703
000704
000705
000706
000707
000708
000709
000710
000711
000712
000713
000714
000715
000716
000717
000718
000719
000720
000721
000722
000723
000724
000725
000726
000727
000728
000729
000730
000731
000732
000733
000734
000735
000736
000737
000738
000739
000740
000741
000742
000743
000744
000745
000746
000747
000748
000749
000750
000751
000752
000753
000754
000755
000756
000757
000758
000759
000760
000761
000762
000763
000764
000765
000766
000767
000768
000769
000770
000771
000772
000773
000774
000775
000776
000777
000778
000779
000780
000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
000846
000847
000848
000849
000850
000851
000852
000853
000854
000855
000856
000857
000858
000859
000860
000861
000862
000863
000864
000865
000866
000867
000868
000869
000870
000871
000872
000873
000874
000875
000876
000877
000878
000879
000880
000881
000882
000883
000884
000885
000886
000887
000888
000889
000890
000891
000892
000893
000894
000895
000896
000897
000898
000899
000900
000901
000902
000903
000904
000905
000906
000907
000908
000909
000910
000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935
000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975
000976
000977
000978
000979
000980
000981
000982
000983
000984
000985
000986
000987
000988
000989
000990
000991
000992
000993
000994
000995
000996
000997
000998
000999

```

```

000000
000040
000100
000140
000200
000240

```

586	000300	PRTY6= 300		
587	000340	PRTY7= 340		
588				
589				
590	100000			
591	040000			
592	020000			
593	010000			
594	004000			
595	002000			
596	001000			
597	000400			
598	000200			
599	000100			
600	000040			
601	000020			
602	000010			
603	000004			
604	000002			
605	000001			
606				
607				
608				
609				
610				
611				
612				
613				
614				
615				
616				
617				
618	000004			
619	000010			
620	000014			
621	000014			
622	000014			
623	000020			
624	000024			
625	000030			
626	000034			
627				
628				
629				
630	000000			
631				
632				
633	000174			
634	000174			
635	176			
636				
637	000200			
638				
639	000200	000137	003254	
640	000204	000137	004146	

```

; MISCELLANEOUS BIT ASSIGNMENT
BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

; VECTOR ADDRESSES
ERRVEC= 4
RESVEC= 10
TBITVEC= 14
TRTVEC= 14
BPTVEC= 14
IOTVEC= 20
PMRVEC= 24
EMTVEC= 30
TRAPVEC= 34
.EQUIV R4,CSR
.EQUIV R4,RCR

.=0
;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS
DISPREG:0
SWREG: 0

.=200
JMP @#BEGIN ;JUMP TO STARTING ADDRESS OF PROGRAM
JMP @#SWPRT ;RESTART AT 204. DO THE RESTART.

```

```

641                                     ;*****
642                                     ;.=1100
643
644                                     ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
645                                     ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
646                                     ;NOTE1: NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
647                                     ;NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
648
649
650                                     ;CALL:
651                                     ;1) USING A TRAP INSTRUCTION
652                                     ;      TYPE      ,MESADR      ;MESADR IS FIRST ADDRESS OF AN ASCII STRING
653                                     ;OR
654                                     ;      TYPE
655                                     ;      MESADR
656
656 001100 010046                                $TYPE:  MOV      RD, -(SP)      ;SAVE RD
657 001102 017600 000002                        MOV      22(SP), RD      ;GET ADDRESS OF ASCII STRING
658 001106 062766 000002 000002                ADD      #2, 2(SP)      ;ADJUST RETURN PC
659 001114 112046                                1$:  MOV     (RD)+, -(SP)    ;PUSH CHARACTER TO BE TYPED ONTO STACK
660 001116 001003                                BNE      2$              ;BR IF IT ISN'T THE TERMINATOR
661 001120 005726                                TST      (SP)+           ;IF TERMINATOR POP IT OFF THE STACK
662 001122 012600                                MOV      (SP)+, RD       ;RESTORE RD
663 001124 000002                                RTI                          ;RETURN
664 001126 004737 001160                        2$:  JSR      PC, 5$          ;GO TYPE THIS CHARACTER
665 001132 122726 000012                        3$:  CMP     #12, (SP)+      ;CHECK IF THE CHAR. TYPED WAS A LINE FEED
666 001136 001366                                BNE      1$             ;GO GET NEXT CHAR. IF NOT LINE FEED
667 001140 013746 001524                        MOV      NULL, -(SP)    ;GET # OF FILLER CHARS. NEEDED
668                                         ;AND THE NULL CHAR.
669 001144 105366 000001                        4$:  DECB    1(SP)        ;DOES A NULL NEED TO BE TYPED?
670 001150 002770                                BLT      3$             ;BR IF NO--GO POP THE NULL OFF OF STACK
671 001152 004737 001160                        JSR      PC, 5$          ;GO TYPE A NULL
672 001156 000772                                BR       4$             ;LOOP
673 001160 105777 000334                        5$:  TST     2TPS          ;WAIT UNTIL PRINTER IS READY
674 001164 100375                                BPL      5$             ;LOAD CHAR TO BE TYPED INTO DATA REG.
675 001166 116677 000002 000326                MOV     2(SP), 2TPB
676 001174 000207                                RTS      PC
677

```

```

678 ;*****
679 ;      DEBUG DUMP ROUTINE
680 ;*****
681      . = 1200
682 001200 013746 011042 DMPHLT: MOV     FLAG, -(SP)
683 001204 042737 000070 011042 BIC     #70, FLAG      ; INIT FLAGS
684 001212 104400 001532          TYPE    ,ASTRK      ; TYPE #
685 001216 104402          GETCHR
686 001220 012637 001314          MOV     (SP)+, 65
687 001224 104400 001314          TYPE    65
688 001230 122737 000127 001542 CMPB     #127, $CHAR      ; W? FOR WORD
689 001236 001004          BNE     15
690 001240 052737 000010 011042 BIS      #BIT3, FLAG      ; SET FLAG BIT
691 001246 000430          BR      35
692 001250 122737 000102 001542 15:    CMPB     #102, $CHAR      ; B? FOR BYTE
693 001256 001004          BNE     25
694 001260 052737 000020 011042 BIS      #BIT4, FLAG
695 001266 000420          BR      35
696 001270 122737 000103 001542 25:    CMPB     #103, $CHAR      ; C? FOR CONTINUE
697 001276 001014          BNE     35
698 001300 012637 011042          MOV     (SP)+, FLAG
699 001304 052737 000040 011042 BIS      #BIT5, FLAG
700 001312 000413          BR      DUMP
701 001314 000000          65:    000000
702 001316 104400 001526          45:    TYPE    ,QUES
703 001322 104400 001536          TYPE    ,CRLF
704 001326 000724          BR      DMPHLT
705 001330 005037 001544          35:    CLR      WORK
706 001334 005726          TST     (SP)+
707 001336 104406 001544          ACCEPT  ,WORK
708 001342 012700 001476 DUMP:    MOV     DMPHLT, R0      ; INIT DUMP LIST
709 001346 062710 000020          ADD     #20, (R0)      ; BUMP ADDRESS
710 001352 032737 000040 011042 BIT      #BIT5, FLAG
711 001360 001005          BNE     L1
712 001362 013737 001544 001476 MOV     WORK, DMPHLT
713 001370 001001          BNE     L1
714 001372 022020          CL?     (R0)+, (R0)+      ; SKIP 1ST TWO ENTRIES
715
716 001374 012001          L1:    MOV     (R0)+, R1      ; GET ADDR OF DATA FROM LIST
717 001376 001700          BEQ     DMPHLT      ; BR IF END OF LIST
718 001400 104400 001536          TYPE    ,CRLF
719 001404 010146          MOV     R1, -(SP)      ; PUSH ADDR ON STACK
720 001406 004037 006336          JSR     R0, $B2016      ; PRINT OUT ADDRESS
721
722 001412 032737 000010 011042 BIT      #BIT3, FLAG
723 001420 001014          BNE     L3
724 001422 012702 000020          MOV     #20, R2      ; SET WORD COUNTER = 8
725 001426 005046          L2:    CLR      -(SP)
726 001430 112116          MOVB     (R1)+, (SP)
727 001432 104400 007132          TYPE    MSG00
728 001436 004037 006324          JSR     R0, $B20CT
729 001442          .BYTE    3
730 001443          .BYTE    1
731 001444 005302          DEC     R2      ; DECREMENT WORD COUNTER
732 001446 001367          BNE     L2      ; BR IF NOT = 0
733 001450 000751          BR      L1      ; GET NEXT ENTRY

```

734						
735	001452	012702	000010	L3:	MOV	#10,R2
736	001456	012146		IS:	MOV	(R1)+,-(SP)
737	001460	104400	007132		TYPE	MSG00
738	001464	004037	006336		JSR	R0,\$B2016
739	001470	005302			DEC	R2
740	001472	001371			BNE	IS
741	001474	000737			BR	L1
742	001476	000000		DNPLST:	0	
743	001500	000000			0	
744	001502	000001		.RX:	.BLKW	1
745	001504	000001		.TX:	.BLKW	1
746	001506	000000			0	
747	001510	000000			0	
748	001512	000000			0	
749	001514	177560		TKS:	177560	
750	001516	177562		TKB:	177562	
751	001520	177564		TPS:	177564	
752	001522	177566		TPB:	177566	
753	001524	000000		NULL:	.WORD	0

; DECREMENT THE WORD COUNT
 ; GET NEXT ENTRY
 ; RESERVED FOR SW. REG
 ; END OF TABLE FOR SW. REG
 ; TTY KEYBOARD STATUS REG. ADDRESS
 ; TTY KEYBOARD DATA BUFFER REG. ADDRESS
 ; TTY PRINTER STATUS REG. ADDRESS
 ; TTY PRINTER BUFFER REG. ADDRESS
 ; CONTAINS NULL CHARACTER FOR FILLS

```

*****
CONSTANTS AND WORKING STORAGE
*****

```

```

754
755
756
757
758
759 001526 037440 000040
760 001532 005015 000052
761 001536 015
762 001537 012 000

```

```

QUES: .ASCIZ " "
ASTRK: .ASCIZ (15)(12)"*"
CRLF: .ASCII (15)
LF: .ASCIZ (12)

```

```

763 001542
764 001544
765 001542 000000
766 001544 000000
767 001546 000000
768 001550 000000
769 001552 000000
770 001554 000000
771 001556 000000
772 001560 000000
773 001562 000000

```

```

.EVEN
$WORK=WORK

```

```

$CHAR: 0
WORK: 0
WORK1: 0
WORK2: 0
WORK3: 0
WORK4: 0
WORK5: 0
WORK6: 0
FLAGS: 0

```

```

: DATA PATTERN ADDRESS TABLE AND PATTERNS

```

```

775 001564 002604
776 001566 003004
777 001570 003076
778 001572 003126
779 001574 002604
780 001576 003004
781 001600 003076
782 001602 003126

```

```

DAT: .WORD VDB ; ADDRESS OF VARIABLE DATA BUFFER
      .WORD DP1 ; ADDRESS OF MESSAGE 1
      .WORD DP2 ; ADDRESS OF MESSAGE 2
      .WORD DP3 ; ADDRESS OF MESSAGE 3
      .WORD DP4 ; ADDRESS OF MESSAGE 4
      .WORD DP5 ; ADDRESS OF MESSAGE 5
      .WORD DP6 ; ADDRESS OF MESSAGE 6
      .WORD DP7 ; ADDRESS OF MESSAGE 7

```

```

: RECEIVER DATA BUFFER STARTS HERE..

```

```

785 001604 000400
786 002604
787 002604 000100

```

```

IBUF: .BLKW 400
VDB: .BLKW 100 ; VARIABLE DATA BUFFER

```

```

789 003004 177 177
003006 040444 052040 042510
003076 177 177

```

```

DP1: .BYTE 177,177
      .ASCIZ 'SA THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.'(15)(12)(001)(177)(177)(177)(1
      .EVEN
DP2: .BYTE 177,177

```

```

003100 041044 030040 031061
003126 003126
003126 177 177

```

```

      .ASCIZ 'SB 0123456789'(15)(12)(001)(177)(177)(177)(177)
      .EVEN
DP3: .BYTE 177,177

```

```

003130 041444 041440 046517
003230 030040 031061 032063
003254

```

```

      .ASCII 'SC COM-TEST MAYNARD THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG'
      .ASCIZ ' 0123456789'(15)(12)(001)(177)(177)(177)(177)
      .EVEN

```

```

790 002604
791 003004
792 003076
793 003126
794
795

```

```

DP4=VDB ; SPARE
DP5=DP1 ; SPARE
DP6=DP2 ; SPARE
DP7=DP3 ; SPARE

```

```

796
797
798
799 003254
800 003254 012706 001070
801 003260 012703 000024
802 003264 012723 006660
803 003270 012723 000340
804 003274 012723 005020
805 003300 012723 000340
806 003304 012723 006532
807 003310 012723 000340
808 003314 104420
809 003316 005037 004762
810 003322 012737 003254 003512
811 003330 005037 001562
812 003334 000005
813
814
815
816 003336 022777 100000 005500
817 003344 001063
818
819
820
821
822 003346 012701 010144
823 003352 005000
824 003354 000000
825 003356 017737 005462 001562
826 003364 032777 020000 005452
827 003372 001402
828 003374 004737 003434
829
830 003400 012701 010366
831 003404 032737 040000 001562
832 003412 001402
833 003414 004737 003434
834
835 003420 012701 011004
836 003424 004737 003434
837
838 003430 000137 004230
839
840 003434 011100
841 003436 000000
842 003440 017721 005400
843 003444 011100
844 003446 000000
845 003450 017721 005370
846 003454 011100
847 003456 000000
848 003460 017721 005360
849 003464 011100
850 003466 100410
851 003470 000600

```

```

*****
START OF PROGRAM
*****
BEGIN:
      MOV      #STACK, SP      ; SETUP THE STACK POINTER
      MOV      #24, R3
      MOV      #SPWRDN, (R3)+  ; POWER FAILURE VECTOR
      MOV      #340, (R3)+    ; LEVEL 7
      MOV      #SHL, (R3)+    ; EMT VECTOR FOR HLT(ERROR) ROUTINE
      MOV      #340, (R3)+    ; LEVEL 7
      MOV      #STRAP, (R3)+  ; TRAP VECTOR FOR TRAP CALLS
      MOV      #340, (R3)+    ; LEVEL 7
      SUSWR
      CLR      $PASS          ; CLEAR THE PASS COUNT
      MOV      #BEGIN, $LPADR ; INITILIZE THE LOOP ADDRESS FOR SCOPE
      CLR      FLAGS          ; RESET FLAGS
      RESET
*****
      GET PARAMETERS FROM OPERATOR
*****
      CMP      #100000, $SWR  ; MANUAL INPUT??
      BNE      GETIT          ; BR IF NO
*****
      SWITCH REG INPUT ROUTINE
*****
MAININ: MOV      #DN+4, R1      ; PRESET POINTER FOR DN-11
      CLR      R0              ; CLEAR DISPLAY
      HALT
      MOV      $SWR, FLAGS     ; SAVE ISR REQUEST INDICATORS
      BIT      #20000, $SWR    ; IS DN11 SETUP REQUESTER?
      BEQ      .+6             ; BR IF NO
      JSR      PC, MANBA       ; GO SETUP DN11
*****
      MOV      #DMB+4, R1      ; PRESET DMB ISR ADDRESS
      BIT      #40000, FLAGS   ; IS DMB SETUP REQUESTED?
      BEQ      .+6             ; BR IF NO
      JSR      PC, MANBA       ; GO SETUP DN11-B
*****
      MOV      #VISR+4, R1     ; PRESET VARIABLE ISR ADDRESS
      JSR      PC, MANBA       ; GO SETUP VARIABLE ISR
*****
      JMP      SWRSET          ; GO GET OPERATIONAL SWITCHES
*****
MANBA: MOV      (R1), R0        ; DISPLAY BUS ADDR
      HALT
      MOV      $SWR, (R1)+     ; HALT FOR BUS ADDR
      MOV      (R1), R0        ; DISPLAY VECTOR ADDR
      HALT
      MOV      $SWR, (R1)+     ; HALT FOR VECTOR ADDR
      MOV      (R1), R0        ; DISPLAY PRIORITY
      HALT
      MOV      $SWR, (R1)+     ; HALT FOR PRIORITY
      MOV      (R1), R0        ; DISPLAY PARAM #1
      BMI      MANINX          ; BR IF PARAM NOT REQUIRED
      HALT
      ; HALT FOR PARAM #1

```

852	003472	017721	005346		MOV	2SWR,	(R1)+	
853	003476	011100			MOV	(R1),	RO	; DISPLAY PARAM #2
854	003500	100403			BMI	MANINX		; BR IF PARAM NOT REQUIRED
855	003502	000000			HALT			; HALT FOR PARAM #2
856	003504	017721	005334		MOV	2SWR,	(R1)+	
857	003510	000207		MANINX:	RTS	PC		
858	003512	000000		SLPADR:	0			
859								
860								
861								
862								
863	003514	012700	011000					
864	003520	104400	007134					
865	003524	104400	011000					
866	003530	104400	001526					
867								
868	003534	104404	001544		GETSTR	WORK		; READIN OPERATOR'S RESPONSE
869	003540	123727	001544	000015	CMPB	WORK	#15	; IS IT CAR. RET?
870	003546	001431			BEQ	GETBA		; BR IF YES
871								
872	003550	012700	010140		MOV	DN,	RO	; PRESET DN ISR ADDR
873	003554	023737	010140	001544	CMP	DN,	WORK	; IS IT DN?
874	003562	001004			BNE	DMBTST		; BR IF NO
875	003564	052737	020000	001562	BIS	#20000,	FLAGS	; SET DN11 FLAG
876	003572	000417			BR	GETBA		; GO GET DN11 PARAMS
877	003574				DMBTST:			
878	003574	012700	010362		MOV	DMB,	RO	; PRESET DM11B ISR
879	003600	023737	010362	001544	CMP	DMB,	WORK	; IS IT DM?
880	003606	001004			BNE	NOISR		; BR IF NO
881	003610	052737	040000	001562	BIS	#40000,	FLAGS	; SET DM11-B FLAG
882	003616	000405			BR	GETBA		; GO GET DM11-B PARAMS
883								
884	003620				NOISR:			
885	003620	104400	007157		TYPE	,MSG02		; <15><12> ISR NOT LOADED!
886								
887	003624	000000			HALT			; HALT
888	003626	000137	003254		JMP	BEGIN		; TRY AGAIN
889	003632	010004			GETBA:	MOV	RO,	R4
890	003634	022020			CMP	(RO)+,	(RO)+	; SAVE POINTER
891	003636	104400	007202		TYPE	,MSG03		; INCREMENT ISR POINTER
892	003642	004737	004764		JSR	PC,GETANY		; <15>' 2' BUS ADDRESS=
893								; GET THE BUS ADDR
894	003646	104400	007221		GETVA:	TYPE	,MSG04	; <15><12> VECTOR ADDRESS=
895	003652	004737	004764		JSR	PC,GETANY		; GET THE VECTOR ADDR.
896								
897	003656	104400	007243		GETPRI:	TYPE	,MSG05	; <15><12> PRIORITY=
898	003662	004737	004764		JSR	PC,GETANY		; GET THE PRIORITY
899								
900	003666	005710			TST	(RO)		; PARAM #1 REQUIRED?
901	003670	100412			BMI	GETP3		; BR IF NO
902	003672	104400	007257		GETPRM:	TYPE	,MSG06	; <15><12> PARAMS #1=
903	003676	004737	004764		JSR	PC,GETANY		; GET PARAM
904								
905	003702	005710			TST	(RO)		; PARAM #2 REQUIRED?
906	003704	100404			BMI	GETP3		; BR IF NO
907	003706	104400	007274		TYPE	,MSG07		; <15><12> PARAMS #2=

```

908 003712 004737 004764          JSR      PC,GETANY      ;GET PARAM
909
910 003716 016437 000016 003734 GETP3: MOV      16(R4), ARIA ;IS ASCII PARAM REQUIRED
911 003724 100424          BMI      GETEX              ;BR IF NO
912 003726 104400 007311          TYPE      ,MSG08        ;<15><12> ASCII PARAM=
913 003732 104400          TYPE      ;PRINTOUT ASCII PARAM
914 003734 000000          ARIA:    0
915 003736 104400 001526          TYPE      ,QUES
916
917 003742 104404          GETSTR              ;GET ASCII INPUT AND
918 003744 001604          IBUF              ;PUT IT HERE
919 003746 012702 001604          MOV      #IBUF, R2        ;SETUP POINTER
920 003752 122712 000015          CMPB     #15, (R2)        ;WAS NEW DATA ENTERED?
921 003756 001407          BEQ      GETEX              ;BR IF NO
922
923 003760 013703 003734          MOV      ARIA, R3          ;SETUP DEST. POINTER
924 003764 112223          MOVB     (R2)+, (R3)+          ;MOV INPUT TO DEST.
925 003766 122712 000015          CMPB     #15, (R2)        ;LAST DIGIT?
926 003772 001374          BNE      .-6                ;LOOP IF NO
927 003774 105013          CLRB     (R3)                ;INSERT ALL ZERO CHAR
928
929
930 003776 020427 011000          GETEX:  CMP      R4, #VISR    ;WAS THIS THE VARIABLE ISR
931 004002 001461          BEQ      SWPNT              ;BR IF YES
932 004004 000137 003514          JMP      GETIT          ;GET ANOPHER
933
934          ;TTY INTERRUPTS HERE WHEN MODULE IS RUNNING.
935
936 004010 017701 175502          TTYINT: MOV      @TKB,R1      ;CLEAR TTY BUFFER
937 004014 042777 000100 175472          BIC      #100,@TKS      ;RESET INT. ENABLE
938 004022 042701 177700          BIC      #177700,R1      ;STRIP JUNK
939 004026 022701 000007          CMP      #7,R1
940 004032 001013          BNE      2$
941 004034 022737 000176 011044          CMP      #SWREG,SWR
942 004042 001006          BNE      1$
943 004044 052737 000001 011042          BIS      #BIT0,FLAG
944 004052 052777 000100 175434          BIS      #100,@TKS
945 004060 000002          1$:      RTI
946 004062 022701 000077          2$:      CMP      #'?', R1      ;IS IT ?
947 004066 001014          BNE      NOQ              ;BR IF NO
948 004070 012700 001476          MOV      #DMPLST,R0      ;SETUP DUMP LIST
949 004074 012710 001604          MOV      #IBUF, (R0)      ;TO PRINTOUT INPUT BUFFER
950 004100 017737 004714 001502          MOV      @IRDA,.RX      ;IF SWITCH REG. =0 PRINT RX BUFFER.
951 004106 017737 004710 001504          MOV      @IXDA,.TX      ;PRINT TX BUFFER
952 004114 000137 001374          JMP      L1          ;AND GO PRINT IT
953
954 004120 022701 000012          NOQ:    CMP      #12, R1      ;IS IT LINE FEED?
955 004124 001002          BNE      RSTART          ;BR IF NO
956 004126 000137 003254          JMP      BEGIN        ;RESTART
957
958 004132 104400 010135          RSTART: TYPE      ,MFILL
959 004136 000005          RESET
960 004140 105227 000000          INCB     #0          ;DELAY HERE FOR AWILE
961 004144 001375          BNE      .-4
962 004146 022737 000176 011044          SWPNT:  CMP      #SWREG,SWR
963 004154 001007          BNE      XSWPNT

```

K02

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 23
 DZITAC.P11 04-AUG-76 11:27 INTERPROCESSOR TEST PROGRAM

```

964 004156 052737 000002 011042 BIS #BIT1,FLAG
965 004164 104400 007330 TYPE ,MSG09
966 004170 104422 SETSWI
967 004172 000417 BR REST
968 004174 104400 007330 XSWPNT: TYPE MSG09 ;(<15><12> SET SWITCHES
969 004200 105777 175310 TSTB @TKS ;WAIT FOR TTY INPUT
970 004204 100375 BPL -4 ;LOOP
971 004206 017702 175304 MOV @TKB,R2 ;RESET DONE FLAG
972 004212 017746 004626 MOV @SWR, -(SP)
973 004216 004037 006336 JSR R0,$@2016 ;PRINTOUT SWITCHES
974 004222 104400 001536 TYPE ,CALF
975 004226 000401 BR +4 ;SKIP OVER HALT
976 *****
977 SET SWITCH OPTIONS
978 *****
979
980 004230 000000 SWRSET: HALT ;HALT FOR SWITCH SETUP
981 ;SW00=ONE WAY OUT
982 ;SW01=ONE WAY IN
983 ;SW02=EXTERNAL LOOPBACK
984 ;SW03=INTERNAL LOOPBACK
985 ;SW04=LOOP ON DATA
986 ;SW05=MONITOR INPUT
987 ;SW06=MONITOR OUTPUT
988 ;SW07=NO DATA COMPARE
989 ;SW08=EXTERNAL DATA
990 ;SW09=DATA SELECT
991 ;SW10=DATA SELECT
992 ;SW11=DATA SELECT
993 ;SW12=
994 ;SW13=INHIBIT ERROR TYPEOUTS
995 ;SW14=LOOP ON TEST
996 ;SW15=HALT ON ERROR
997
998 004232 012737 004240 003512 REST: MOV #RESTR, $LPADR ;SETUP LOOP
999 004240 017701 004600 RESTR: MOV @SWR,R1
1000 004244 000301 SWAB R1
1001 004246 032777 000017 004570 BIT #17, @SWR ;WAS SOME MODE SELECTED?
1002 004254 001003 BNE +10 ;BR IF YES
1003 004256 104400 007617 TYPE ,MSG21 ;(<15><12>NO MODE SELECTED.
1004 004262 000723 BR RSTART ;GO GET SWITCH REGISTER.
1005 004264 042701 177761 BIC #177761, R1 ;STRIP JUNK
1006 004270 046137 001564 011022 MOV DAT(1), IXDA ;SETUP INIT DATA ADDR FROM TABLE
1007 004276 005701 TST R1 ;VARIABLE DATA SPECIFIED?
1008 004300 001010 BNE SUXCC ;BR IF NO
1009 004302 032777 000400 004534 BIT #400, @SWR ;USE EXTERNAL DATA?
1010 004310 001004 BNE SUXCC ;BR IF YES
1011 *****
1012 GET VARIABLE DATA
1013 *****
1014 004312 104400 007352 TYPE ,MSG10 ;(<15><12> ENTER DATA <15><12>
1015 004316 104404 GETSTR ;GET DATA
1016 004320 002604 VDB ;AND PUT IT HERE
1017 004322 012737 001604 011020 SUXCC: MOV #IBUF, IRDA ;SETUP READ BUFFER ADDR
1018
1019 004330 032777 000400 004506 BIT #400, @SWR ;EXTERNAL DATA?

```

L02

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 24
 DZITAC.P11 04-AUG-76 11:27 INTERPROCESSOR TEST PROGRAM

1020	004336	001403			BEG	SWRNXT		;BR IF NO
1021	004340	012737	002604	011022	MOV	#VDB, IXDA		;SETUP BUFFER ADDRESS
1022	004346				SWRNXT:			
1023	004346	012737	004010	000060	MOV	#TTYINT, #60		;SETUP TTY VECTOR
1024	004354	012737	000340	000062	MOV	#340, #62		
1025	004362	012777	000100	175124	MOV	#100, #TKS		;AND ENABLE INTERRUPTS
1026	004370	012702	001604		MOV	#IBUF, R2		
1027	004374	005022			CLRIB:	CLR (R2)+		;CLEAR INPUT BUFFER
1028	004376	022702	002004		CMP	#IBUF+200, R2		
1029	004402	001374			BNE	CLRIB		

M02

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 25
 DZITAC.P11 04-AUG-76 11:27 INTERPROCESSOR TEST PROGRAM

```

1030 ;*****
1031 ;      SETUP TIMER      *
1032 ;*****
1033
1034 004404 012737 000060 006322 SUTIME: MOV    #60,    MSEC5 ;PRESET COUNTER
1035 004412 012737 006300 000100      MOV    #TIMER, 100 ;SETUP LINE CLOCK VECTOR
1036 004420 012737 000340 000102      MOV    #340, 102 ;AND PRIORITY
1037 004426 012737 004444 000004      MOV    #NOLC, 4 ;SETUP BUS ERROR VECTOR
1038 004434 052737 000100 177546      BIS    #100, 177546 ;ENABLE LINE CLOCK
1039 004442 000423
1040
1041 ;BUS ERROR RETURNS HERE IF NO LINE CLOCK
1042
1043 004444 012737 006300 000104 NOLC:  MOV    #TIMER, 104 ;SETUP RTC VECTOR
1044 004452 012737 000340 000106      MOV    #340, 106 ;AND PRIORITY
1045 004460 012737 004506 000004      MOV    #15, 4 ;SETUP BUS ERROR VECTOR
1046 004466 012737 003100 172542      MOV    #1600., 172542 ;SET COUNTER BUFFER.
1047 004474 012737 000111 172540      MOV    #111, 172540 ;ENABLE REAL TIME TIME CLOCK
1048 004502 000240      NOP ;WAIT
1049 004504 000402      BR    NORTC ;CONTINUE.
1050 004506 104400 007550      15:  TYPE    ,MSG19 ;<15><12> NO CLOCK AVAILABLE.
1051
1052 004512 000137 004516      NORTC: JMP     .+4 ;SPARE JUMP
1053 004516 012737 000006 000004      MOV    #6,2#4 ;SET TRAP VECTOR
1054 004524 005037 000006      CLR     2#6 ;SET BUS ERROR VECTOR
1055 004530 012706 001070      MOV    #STACK, SP ;SETUP STACK
1056 004534 104414 000000      STPS, PRTY0
1057 004540 012737 006336 011030      MOV    #5B2016,B2016 ;SETUP BIN TO OCT ADDR
1058
1059
1060 ;*****
1061 ;      DO TESTING NOW      *
1062 ;*****
1063 004546 032737 020000 001562      BIT     #20000, FLAGS ;WAS A DN11 SETUP
1064 004554 001402      BEQ     DMCHK ;BR IF NO
1065 004556 004737 010160      JSR     PC,DNGO ;GO TO DN11 ISR
1066
1067 004562 032737 040000 001562 DMCHK: BIT     #40000, FLAGS ;WAS A DM11-B SETUP?
1068 004570 001402      BEQ     VIGO ;BR IF NO
1069 004572 004737 010402      JSR     PC,DMGO ;GO TO DM11-B ISR
1070
1071 ;*****
1072 ;      GOTO THE MODULE AND RUN      *
1073 ;*****
1074
1075 004576 004777 004234      VIGO:  JSR     PC,2ISR+36 ;GO TO ISR

```

```

1076 ;*****
1077 ;      END OF PASS ROUTINE
1078 ;*****
1079 004602 005237 004762 EOP: INC $PASS ;INCREMENT PASS COUNTER
1080 004606 005746 TST -(SP) ;PUSH DOWN AND PROTECT STACK.
1081 004610 104416 KBDIN
1082 004612 032777 010000 004224 BIT #SW12,JSWR ;INHIBIT TYPEOUTS?
1083 004620 001034 BNE 2$ ;BR IF YES
1084 004622 104400 007371 TYPE MSG11 ;<15><12> END OF PASS
1085 004626 013746 004762 MOV $PASS, -(SP)
1086 004632 004037 006336 JSR RD,$B2016 ;PRINTOUT PASS COUNT
1087 004636 104400 001536 TYPE ,CRLF
1088 004642 032700 000002 BIT #OWI,MODE ;SKIP TRANSMIT TYPEOUT IF OWI
1089 004646 001012 BNE 3$ ;BR IF YES
1090 004650 104400 010040 TYPE MSG26 ;TRANSMITTED DATA=
1091 004654 013737 011022 004664 MOV IXDA,4$ ;SET POINTER TO TXBUF
1092 004662 104400 TYPE ;TYPE TXBUFFER
1093 004664 000000 4$: 0
1094 004666 032700 000001 BIT #OWO,MODE ;SKIP RECEIVE TYPEOUT IF OWO
1095 004672 001007 BNE 2$ ;BR IF YES
1096 004674 104400 010066 3$: TYPE MSG27 ;RECEIVED DATA=
1097 004700 013737 011020 004710 MOV IRDA,5$ ;SET POINTER TO RXBUF
1098 004706 104400 TYPE ;TYPE RXBUFFER
1099 004710 000000 5$: 0
1100 004712 032777 040000 004124 2$: BIT #BIT14,JSWR ;LOOP ON TEST?
1101 004720 001005 BNE 1$ ;BR IF NO...
1102 004722 016600 000002 MO 2(SP),RO ;GET RETURN ADDRESS
1103 004726 104414 000000 STPS,PTYO
1104 004732 000110 JMP (RO) ;GO BACK TO MODULE.
1105 004734 012706 001070 1$: MOV #STACK,SP ;RESET THE STACK POINTER.
1106 004740 013700 000042 MOV #42, RO ;GET MONITOR ADDRESS
1107 004744 001404 BEQ $DOAGN ;BR IF NONE
1108 004746 004710 JSR PC,(RO) ;GO TO MONITOR
1109 004750 000240 NOP ;SAVE ROOM FOR
1110 004752 000240 NOP ;ACT-11
1111 004754 000240 NOP
1112 004756 000137 000200 $DOAGN: JMP #200 ;RESTART TEST
1113 004762 000000 $PASS: 0
1114
1115
1116 ;*****
1117 ;      SUBROUTINE TO INPUT OCTAL WORD FROM OPERATOR
1118 ;*****
1119 004764 011046 GETWY: MOV (RO), -(SP) ;PUT WORD ON STACK
1120 004766 004037 006336 JSR RD,$B2016 ;AND TYPE IT
1121 004772 104400 001526 TYPE QUES
1122 004776 011037 001544 MOV (RO),WORK ;PRESET FOR DEFAULT (CR)
1123 005002 104406 001544 ANYMOR: ACCEPT WORK ;OCTAL READIN
1124 005006 013710 001544 MOV WORK, (RO) ;MOVE IT TO ISR
1125 005012 005720 ANYEX: TST (RO) ;BUMP POINTER
1126 005014 000240 NOP
1127 005016 000207 RTS PC ;SUB/ROUTINE EXIT
1128
1129 ;*****
1130 ;      ERROR HLT HANDLER
1131 ;*****

```

1132	005020			SHLT:			
1133	005020	104414	000140		STPS,PRTY3		: LOWER PSM PRIOTITY TO 3
1134	005024	005237	005716	SHLOT:	INC	SERTTL	: INCREMENT ERROR COUNTER
1135	005030	001775			BEQ	SHLOT	: MAKE SURE ITS NOT ZERO
1136	005032	011637	005714		MOV	(SP)	: SAVE ADDRESS OF HLT
1137	005036	162737	000002	005714	SUB	R2,	: AND BACK IT UP
1138	005044	010146			MOV	R1,	: SAVE R1
1139							
1140	005046	032777	020000	003770	BIT	#BIT13, QSWR	: INHIBIT ERP TYPEOUTS?
1141	005054	0C 07C			BNE	TRX	: BR IF YES
1142							
1143	005056	104400	001536		TYPE	CRLF	
1144	005052	117701	000626		MOVB	SHLTAD,R1	: EXTRACT HLT CODE
1145	005056	006301			ASL	R1	: AND ALIGN IT
1146	005070	016137	005256	005100	MOV	ENTAB(R1),.+10	: GET HEADER ADDRESS
1147	005076	104400	005306		TYPE	,END	: AND PRINT HEADER
1148	005102	104400	007410		TYPE	MSG12	: (AT LOC)
1149	005106	013746	005714		MOV	SHLTAD, -(SP)	: GET HLT ADDRESS
1150	005112	004037	006336		JSR	R0, \$B2016	: AND PRINT IT
1151	005116	005701			TST	R1	: HLT CODE = 0?
1152	005120	001446			BEQ	TRX	: BR IF YES
1153							
1154							
1155	005122	022701	000016		CMP	#16,R1	: IS IT HLT+7?
1156	005126	001023			BNE	R1	: BR IF NO
1157	005130	005702			TST	R2	: PRINTOUT BAD DATA?
1158	005132	001406			BEQ	R2	: BR IF NO
1159	005134	104400	007747		TYPE	MSG23	: (BAD DATA=)
1160	005140	110246			MOVB	R2, -(SP)	: GET DATA
1161	005142	004037	006324		JSR	R0, \$B20CT	: AND PRINT IT
1162	005146	003			.BYTE	3	
1163	005147	001			.BYTE	1	
1164	005150	005703		25:	TST	R3	: PRINT OUT GOOD DATA?
1165	005152	001410			BEQ	35	: BR IF NO
1166	005154	104400	007763		TYPE	MSG24	: (GOOD DATA=)
1167	005160	110346			MOVB	R3, -(SP)	: GET DATA
1168	005162	004037	006324		JSR	R0, \$B20CT	: AND PRINT IT
1169	005166	003			.BYTE	3	
1170	005167	001			.BYTE	1	
1171	005170	104400	001536		TYPE	CRLF	
1172	005174	000420		35:	BR	TRX	
1173							
1174							
1175	005176	005702		15:	TST	R2	: PRINTOUT RCV CSR?
1176	005200	001405			BEQ	TR3	: BR IF NO
1177	005202	104400	007421		TYPE	MSG13	: (RCV CSR=)
1178	005206	010246			MOV	R2, -(SP)	: GET DATA
1179	005210	004037	006336		JSR	R0, \$B2016	: AND PRINT IT
1180							
1181	005214	005703		TR3:	TST	R3	: PRINTOUT XMIT CSR?
1182	005216	001407			BEQ	TRX	: BR IF NO
1183	005220	104400	007433		TYPE	MSG14	: (XMIT CSR=)
1184	005224	010346			MOV	R3, -(SP)	: GET DATA
1185	005226	004037	006336		JSR	R0, \$B2016	: AND PRINT IT
1186	005232	104400	001536		TYPE	,CRLF	
1187							

C03

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 28
 DZITAC.P11 04-AUG-76 11:27 INTERPROCESSOR TEST PROGRAM

```

1188 005236 032777 100000 003600 TRX: BIT #BIT15, JSWR ;HALT ON ERROR?
1189 005244 001401 BEQ MLTX ;BR IF NO
1190 005246 000000 HALT
1191
1192 005250 104416 MLTX: KBOIN
1193 005252 012601 MOV (SP)+, R1 ;RESTORE R1
1194 005254 000302 RTI ;AND RETURN TO PROGRAM
1195
1196 005256 005306 ENTAB: ENO
1197 005260 005321 EM1
1198 005262 005331 EM2
1199 005264 005351 EM3
1200 005266 005375 EM4
1201 005270 005410 EM5
1202 005272 005434 EM6
1203 005274 005530 EM7
1204 005276 005553 EM10
1205 005300 005630 EM11
1206 005302 005655 EM12
1207 005304 005677 EM13
1208

```

```

005306 051105 047522 020122 ENO: .ASCIZ "ERROR HALT"
005321 127 044501 044524 EM1: .ASCIZ "WAITING"
005331 127 044501 044524 EM2: .ASCIZ "WAITING TO XMIT"
005351 104 026516 030461 EM3: .ASCIZ "DN-11 NOT AVAILABLE"
005375 104 030516 026461 EM4: .ASCIZ "DN11-ERROR"
005410 047104 030461 041440 EM5: .ASCIZ "DN11 CALL ABANDONED"
005434 041522 020126 052502 EM6: .ASCIZ "RCV BUFFER FULL, END OF MESSAGE CHARACTER(001) WAS NOT FOUND"
005530 040504 040524 041440 EM7: .ASCIZ "DATA COMPARE ERROR"
005553 105 051122 051117 EM10: .ASCIZ "ERROR RCV CSR=CONTENTS OF SELECT 0 REGISTER"
005630 047125 054105 042520 EM11: .ASCIZ "UNEXPECTED INTERRUPT"
005655 116 046530 050040 EM12: .ASCIZ "LXM PRINCIPAL CAR"
005677 116 046530 040440 EM13: .ASCIZ "LXM ALT CAR"
005714 005714 .EVEN
1209 005714 000000 SHLTD: 0
005716 000000 SERTTL: 0

```

```

1210      ;*****
1211      ;      READ A CHAR. ROUTINE
1212      ;*****
1213
1214      :      CALL=   GETCHR   ;INPUT A CHAR FROM TTY
1215      :                      ;RETURNS HERE WITH CHAR ON STACK
1216      $READC: MOV      (SP), -(SP)      ;PUSH THE PC
1217      MOV      4(SP), 2(SP)      ;SAVE THE PS
1218      TSTB     @TKS              ;IS RECEIVE DONE
1219      BPL      -4                ;LOOP IF NO
1220      MOV      @TKB, $CHAR      ;SAVE THE CHAR.
1221      BIC      @177600, $CHAR    ;STRIP JUNK
1222      MOV      $CHAR, 4(SP)      ;PUT CHAR ON STACK
1223      RTI                      ;EXIT
1224      ;*****
1225      ;      READ A STRING ROUTINE
1226      ;*****
1227
1228      :      CALL=   GETSTR   ;INPUT A STRING OF CHARS FROM TTY
1229      :      ADDR   ;TO THIS ADDRESS
1230      :                      ;TERMINATE INPUT WITH LINE FEED
1231      $READS: MOV      (SP), R2      ;SETUP ADDRESS OF INPUT BUFFER
1232      MOV      (R2)+, R1          ;INCREMENT RETURN ADDRESS
1233      MOV      R2, (SP)          ;AND PUT BACK ON STACK
1234
1235      :      GETIC: GETCHR          ;GET A CHAR
1236      TYPE     $CHAR              ;IS IT BINARY DELIMITER
1237      CMPB     @136, (SP)+        ;BR IF NO
1238      BNE      GOTIC
1239
1240      :      OCT:   ACCEPT $WORK      ;GET OCT. CHAR
1241      MOVB     $WORK, (R1)+        ;STORE OCT. CHAR.
1242      CMPB     @136, $CHAR        ;TERMINATOR=BIN. DELIMITER?
1243      BNE      OCT               ;BR IF NO
1244      BR       GETIC
1245
1246      :      GOTIC: MOVB     $CHAR, (R1)+ ;STORE CHAR. IN BUFFER
1247      CMP      @15, $CHAR          ;IS IT END OF INPUT (CAR. RETURN)
1248      BNE      GETIC              ;BR IF NO
1249      MOVB     @12, (R1)+
1250      TYPE     LF                  ;TYPE A LINE FEED
1251      MOVB     @001, (R1)+        ;INSERT RX TERM.
1252      MOVB     @177, (R1)+        ;AND A FILL
1253      MOVB     @177, (R1)+        ;INSERT ANOTHER FILL
1254      MOVB     @177, (R1)+        ;INSERT 3RD FILL
1255      MOVB     @177, (R1)+        ;INSERT 4TH FILL
1256      CLRB     (R1)              ;PUT ZEROS AT END
1257      RTI
1258

```

E03

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 30
 DZITAC.P11 04-AUG-76 11:27 INTERPROCESSOR TEST PROGRAM

```

1259
1260
1261
1262
1263
1264 006102
1265 006102 010046
1266 006104 010146
1267 006106 010246
1268 006110 010346
1269 006112 016600 000010
1270 006116 005001
1271 006120 012702 000006
1272 006124 104402
1273 006126 112603
1274 006130 110337 006276
1275 006134 104400 006276
1276 006140 022703 000025
1277 006144 001451
1278 006146 022703 000015
1279 006152 001427
1280 006154 022703 000040
1281 006160 001433
1282 006162 022703 000136
1283 006166 001430
1284 006170 032703 000110
1285 006174 001011
1286
1287 006176 005302
1288 006200 002407
1289 006202 006301
1290 006204 006301
1291 006206 006301
1292 006210 042703 177770
1293 006214 050301
1294 006216 000742
1295 006220 104400 001526
1296 006224 104400 001536
1297 006230 000732
1298 006232 104400 001537
1299 006236 022702 000006
1300 006242 001002
1301 006244 013001
1302 006246 005740
1303 006250 010130
1304 006252 010066 000010
1305 006256 012603
1306 006260 012602
1307 006262 012601
1308 006264 012600
1309 006266 000002
1310 006270 104400 010130
1311 006274 000710
1312 006276 000 000

*****
ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
CALL:
      ACCEPT .ADDR          ;PUT OCTAL NUMBER IN ADDR
$ACCEPT:
      MOV      R0,-(SP)      ;SAVE R0
      MOV      R1,-(SP)      ;SAVE R1
      MOV      R2,-(SP)      ;SAVE R2
      MOV      R3,-(SP)      ;SAVE R3
      MOV      10(SP),R0     ;GET ADDRESS OF WHERE TO PUT NUMBER
1$:   CLR      R1            ;CLEAR PARTIAL NUMBER
      MOV      #6,R2         ;MAX. # OF DIGITS ALLOWED
2$:   GETCHR          ;GET ONE CHARACTER
      MOVB     (SP)+,R3       ;AND PUT IT IN R3
      MOVB     R3,R6         ;ECHO THE CHARACTER
      TYPE     ,R6
      CMP      #25,R3
      BEQ      #5
      CMP      #15,R3
      BEQ      #5
      CMP      #40,R3
      BEQ      #5
      CMP      #136,R3
      BEQ      #5
      BIT      #110,R3
      SNE      #4
      DEC      R2
      BLT      #4
      ASL      R1
      ASL      R1
      ASL      R1
      BIC      #10<7>,R3
      BIS      R3,R1
      BR       2$
4$:   TYPE     ,QUES
      TYPE     ,CR LF
      BR       1$
5$:   TYPE     ,LF
      CMP      #6,R2
      BEQ      #6
      ENE      ,+6
      MOV      2(R0)+,R1
      TST      -(R0)
7$:   MOV      R1,2(R0)+
      MOV      R0,10(SP)
      MOV      (SP)+,R3
      MOV      (SP)+,R2
      MOV      (SP)+,R1
      MOV      (SP)+,R0
      RTI
8$:   TYPE     ,CTLU
      BR       1$
6$:   .BYTE    0,0          ;STORAGE FOR ASCII CHAR, AND TERMINATOR

```

```

1313 ;*****
1314 ;CLOCK INTERRUPT ROUTINE
1315 ;*****
1316 006300 005337 006322 TIMER: DEC MSECs ;COUNT 60 CYCLES
1317 006304 001005 BNE TIMEX ;BR IF NOT 60
1318 006306 012737 000060 006322 MOV #60 MSECs ;RESTORE COUNT
1319 006314 005237 011032 INC TIME ;INCREMENT SECONDS
1320 006320 TIMEX:
1321 006320 000002 RTI ;RETURN FROM INTERRUPT
1322 006322 000000 MSECs: 0
1323 ;*****
1324 ;BINARY TO OCTAL (ASCII) AND TYPE
1325 ;$B20CT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1326 ;CALL:
1327 ;MOV NUM, -(SP) ;NUMBER TO BE TYPED
1328 ;JSR RO, $B20CT ;CALL FOR TYPEOUT
1329 ;.BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1330 ;.BYTE M ;M=1 OR 0
1331 ; ;1=TYPE LEADING ZEROS
1332 ; ;0=SUPPRESS LEADING ZEROS
1333
1334 ;$B201---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B20CT OR $B2016
1335 ;CALL:
1336 ;MOV NUM, -(SP)
1337 ;JSR RO, $B201
1338
1339 ;$B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1340 ;CALL:
1341 ;MOV NUM, -(SP)
1342 ;JSR RO, $B2016
1343
1344 006324 112037 006531 $B20CT: MOVB (RO)+, $SOMODE+1 ;PICKUP THE NUMBER OF DIGITS TO TYPE
1345 006330 112037 006527 MOVB (RO)+, $OFILL ;GET THE ZERO FILL SWITCH
1346 006334 000406 BR $B201
1347 006336 112737 000001 006527 $B2016: MOVB #1, $OFILL ;SET THE ZERO FILL SWITCH
1348 006344 112737 000006 006531 MOVB #6, $SOMODE+1 ;SET FOR SIX(6) DIGITS
1349 006352 112737 000005 006526 $B201: MOVB #5, $OCNT ;SET THE ITERATION COUNT
1350 006360 010346 MOV R3, -(SP) ;SAVE R3
1351 006362 010446 MOV R4, -(SP) ;SAVE R4
1352 006364 010546 MOV R5, -(SP) ;SAVE R5
1353 006366 113704 006531 MOVB $SOMODE+1, R4 ;GET THE NUMBER OF DIGITS TO TYPE
1354 006372 005404 NEG R4
1355 006374 062704 000006 ADD #6, R4 ;SUBTRACT IT FOR MAX. ALLOWED
1356 006400 110437 006530 MOVB R4, $SOMODE ;SAVE IT FOR USE
1357 006404 113704 006527 MOVB $OFILL, R4 ;GET THE ZERO FILL SWITCH
1358 006410 016605 000010 MOV 10(SP), R5 ;PICKUP THE INPUT NUMBER
1359 006414 005003 CLR R3 ;CLEAR THE OUTPUT WORD
1360 006416 006105 1$: ROL R5 ;ROTATE MSB INTO "C"
1361 006420 000404 BR 3$ ;GO DO MSB
1362 006422 006105 2$: ROL R5 ;FORM THIS DIGIT
1363 006424 006105 ROL R5
1364 006426 006105 ROL R5
1365 006430 010503 MOV R5, R3
1366 006432 006103 3$: ROL R3 ;GET LSB OF THIS DIGIT
1367 006434 105337 006530 DECB $SOMODE ;TYPE THIS DIGIT?
1368 006440 100016 BPL 7$ ;BR IF NO

```

1369	006442	042703	177770	BIC	#177770,R3	:GET RID OF JUNK
1370	006446	001002		BNE	4\$	:TEST FOR 0
1371	006450	005704		TST	R4	:SUPPRESS THIS 0?
1372	006452	001403		BEQ	5\$	:BR IF YES
1373	006454	005204	4\$:	INC	R4	:DON'T SUPPRESS ANYMORE 0'S
1374	006456	052703	000060	BIS	#'0,R3	:MAKE THIS DIGIT ASCII
1375	006462	052703	001040	BIS	#' ,R3	:MAKE ASCII IF NOT ALREADY
1376	006466	110337	006524	MOVB	R3,8\$	:SAVE FOR TYPING
1377	006472	104400	006524	TYPE	8\$	:GO TYPE THIS DIGIT
1378	006476	105337	006526	7\$:	DECB	\$OCNT
1379	006502	003347		BGT	2\$	:COUNT BY 1
1380	006504	002402		BLT	6\$	:BR IF MORE TO DO
1381	006506	005204		INC	R4	:BR IF DONE
1382	006510	000744		BR	2\$	:INSURE LAST DIGIT ISN'T A BLANK
1383	006512	012605		6\$:	MOV	(SP)+,R5
1384	006514	012604		MOV	(SP)+,R4	:GO DO THE LAST DIGIT
1385	006516	012603		MOV	(SP)+,R3	:RESTORE R5
1386	006520	012616		MOV	(SP)+,(SP)	:RESTORE R4
1387	006522	000200		RTS	R0	:RESTORE R3
1388	006524	000		8\$:	.BYTE	:SET THE STACK FOR RETURNING
1389	006525	000		.BYTE	0	:RETURN
1390	006526	000		\$OCNT:	.BYTE	:STORAGE FOR ASCII DIGIT
1391	006527	000		\$OFILL:	.BYTE	:TERMINATOR FOR TYPE ROUTINE
1392	006530	000000		\$OMODE:	0	:OCTAL DIGIT COUNTER
1393						:ZERO FILL SWITCH
1394						:NUMBER OF DIGITS TO TYPE

```

;*****
;TRAP HANDLER
$TRAP:  MOV    R0,-(SP)      ;SAVE R0
        MOV    2(SP),R0    ;GET TRAP ADDRESS
        TST    -(R0)       ;BACKUP BY 2
        MOVB   (R0),R0     ;GET RIGHT BYTE OF TRAP
        MOV    $TRPAD(R0),R0 ;INDEX TO TABLE
        RTS    R0          ;GO TO ROUTINE

```

```

;TRAP TABLE
;ROUTINE
;-----

```

1408	006552	001100		\$TRPAD:	STYPE	
1409		104400				TYPE=TRAP+0
1410	006554	005720		\$READC		GETCHR=TRAP+2
1411		104402		\$READS		GETSTR=TRAP+4
1412	006556	005762		\$ACCEPT		ACCEPT=TRAP+6
1413		104404		\$WAIT		RWAIT=TRAP+10
1414	006560	006102		\$XWAIT		XWAIT=TRAP+12
1415		104406		.STPS		STPS=TRAP+14
1416	006562	006576		.KBDIN		KBDIN=TRAP+16
1417		104410		.SUSWR		
1418	006564	006606				
1419		104412				
1420	006566	007030				
1421		104414				
1422	006570	006720				
1423		104416				
1424	006572	007052				

```

1425      104420
1426 006574 006736
1427      104422
1428
1429
1430
1431
1432
1433 006576 104400 007442
1434 006602 000137 006612
1435
1436 006606 104400 007465
1437
1438 006612 011646
1439 006614 004037 006336
1440
1441 006620 104400 010001
1442
1443 006624 016646 000004
1444 006630 016666 000004 000006
1445 006636 016666 000002 000004
1446 006644 012616
1447 006646 004037 006336
1448
1449 006652 104400 007544
1450
1451
1452 006656 000002
1453
1454
1455
1456
1457
1458
1459
1460
1461 006660 012737 006672 000024
1462 006666 000000
1463 006670 000776
1464
1465
1466
1467
1468
1469
1470
1471
1472 006672 012737 006660 000024
1473 006700 012706 001070
1474 006704 104400 007576
1475 006710 104414 000000
1476 006714 000137 000200
1477
1478 006720 032737 000001 011042
1479 006726 001437
1480 006730 042777 000100 172556

```

SUSWR=TRAP+20
.SETSWI
SETSWI=TRAP+22

SPECIAL PRINTOUT ROUTINES

\$RWAIT: TYPE ,MSG15 ;(15)<(12) WAITING AT LOC (SP)
JMP WAITPO

\$XWAIT: TYPE ,MSG16 ;(15)<(12) WAITING FOR CLEAR TO SEND AT LOC (SP)<(SP)

WAITPO: MOV (SP), -(SP) ;SETUP ADDRESS OF CALL FOR PRINTOUT
JSR R0,\$B2016 ;PRINTOUT ADDRESS

TYPE ,MSG25 ; (DMBB LINE STATUS REG=)

MOV 4(SP), -(SP) ;MOV CSR TO BOTTOM OF STACK
MOV 4(SP), 6(SP) ;MOVE PSW UP A WORD
MOV 2(SP), 4(SP) ;MOVE PC UP A WORD
MOV (SP)+, (SP) ;MOVE CSR UP A WORD
JSR R0,\$B2016 ;GO PRINT CSR

TYPE ,MSG18 ;(15)<(12)
;GIVE IT LINE FEED

RTI ;AND RETURN TO CALLER

POWER DOWN ROUTINE.
SINCE INIT IS ISSUED IN A PWR DN/UP SEQUENCE.
PROGRAM MUST BE RESTARTED AGAIN.

SPWRDN: MOV #SPWRUP, 2#24
HALT
BR -2

POWER UP ROUTINE.
MESSAGE "POWER HAS FAILED..." WILL BE PRINTED OUT.
PROGRAM WILL BE RESTARTED.

SPWRUP: MOV #SPWRDN, 2#24
MOV #STACK, SP
TYPE ,MSG20 ;(15)<(12) POWER FAILED..
STPS, PRTYD
JMP 2#200

.KBDIN: BIT #BIT0, FLAG ;TEST IG FLAG
BEQ OUT ;NO EXIT
BIC #100, 2TKS ;CLEAR TTY IE

1481	006736	104400	010111		.SETSWI:TYPE	SNEQ		:TYPE SWR=
1482	006742	017746	002076		MOV	2SWR,-(SP)		:SET UP OCTAL TYPEOUT
1483	006746	004037	006336		JSR	RD,\$B2016		:DO IT
1484	006752	104400	010121		TYPE	NEQ		:TYPE NEW=
1485	006756	017737	002062	001544	MOV	2SWR,WORK		:SET UP FOR CR DEFAULT
1486	006764	104406	001544		ACCEPT	WORK		:GET VALUE
1487	006770	013777	001544	002046	MOV	WORK,2SWR		:REPLACE IT
1488	006776	032737	000002	011042	BIT	#BIT1,FLAG		:SEE HOW WE GOT HERE
1489	007004	001005			BNE	1\$		:WRONG WAY?
1490	007006	005077	172504		CLR	2TKB		:CLEAR BUFFER
1491	007012	052777	000100	172474	BIS	#100,2TKS		:RESET TTY IE
1492	007020	042737	000003	011042	BIC	#BIT0+BIT1,FLAG		:CLEAR FLAG BITS
1493	007026	000002			OUT:	RTI		:EXIT
1494								
1495	007030	042766	000340	000002	.STPS:	BIC	#PRTY7,2(SP)	:CLEAR OUT PRIORITY BITS
1496	007036	057666	000000	000002	BIS	2(SP),2(SP)		:SET NEW PRIORITY
1497	007044	062716	000002		ADD	#2,(SP)		:SETUP EXIT
1498	007050	000002			RTI			:EXIT
1499								
1500	007052	013746	000006		.SUSWR:	MOV	6,-(SP)	:SAVE 6 ON STACK
1501	007056	013746	000004		MOV	4,-(SP)		:SAVE 4 ON STACK
1502	007062	012737	007102	000004	MOV	#1\$,4		:SETUP TIMEOUT
1503	007070	022777	177777	001746	CMP	#-1,2SWR		:TEST FOR 177570
1504	007076	001402			BEQ	2\$		:NOT ALL 1'S
1505	007100	000407			BR	3\$		:IT'S THERE - EXIT
1506	007102	022626			1\$: CMP	(SP)+,(SP)+		:ADJUST STACK AFTER TRAP
1507	007104	012737	000176	011044	2\$: MOV	#SWREG,SWR		:REPLACE HARDWARE REGISTERS
1508	007112	012737	000174	011046	MOV	#DISPREG,DISPLAY		:WITH SOFTWARE REGISTERS
1509	007120	012637	000004		3\$: MOV	(SP)+,4		:RESTORE 4
1510	007124	012637	000006		MOV	(SP)+,6		:RESTORE 6
1511	007130	000002			RTI			:EXIT

1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520

 :
 : AREA RESERVED FOR MOST ASCIZ MESSAGES.
 :

007132	000040			MSG00:	.ASCIZ	/ /
007134	005015	047111	042524	MSG01:	.ASCIZ	<15><12>/INTERFACE TYPE /
007157	015	044412	051123	MSG02:	.ASCIZ	<15><12>/ISR NOT LOADED!!/
007202	005015	052502	020123	MSG03:	.ASCIZ	<15><12>/BUS ADDRESS=/
007221	015	053012	041505	MSG04:	.ASCIZ	<15><12>/VECTOR ADDRESS=/
007243	015	050012	044522	MSG05:	.ASCIZ	<15><12>/PRIORITY=/
007257	015	050012	051101	MSG06:	.ASCIZ	<15><12>/PARAMS #1=/
007274	005015	040520	040522	MSG07:	.ASCIZ	<15><12>/PARAMS #2=/
007311	015	040412	041523	MSG08:	.ASCIZ	<15><12>/ASCII PARAM=/
007330	005015	042523	020124	MSG09:	.ASCIZ	<15><12>/SET SWITCHES.../
007352	005015	047105	042524	MSG10:	.ASCIZ	<15><12>/ENTER DATA/<15><12>
007371	015	042412	042116	MSG11:	.ASCIZ	<15><12>/END OF PASS /
007410	040440	020124	047514	MSG12:	.ASCIZ	/ AT LOC /
007421	040	041522	020126	MSG13:	.ASCIZ	/ RCV CSR=/
007433	040	041530	051123	MSG14:	.ASCIZ	/ XCSR=/
007442	005015	053440	044501	MSG15:	.ASCIZ	<15><12>/ WAITING AT LOC /
007465	015	053412	044501	MSG16:	.ASCIZ	<15><12>/WAITING FOR CLEAR TO SEND AT LOC /
007531	040	020041	041440	MSG17:	.ASCIZ	/ ! CSR= /
007544	006456	000012		MSG18:	.ASCIZ	/./<15><12>
007550	005015	047516	041440	MSG19:	.ASCIZ	<15><12>/NO CLOCKS AVAILABLE/
007576	005015	047520	042527	MSG20:	.ASCIZ	<15><12>/POWER FAILED../
007617	015	047012	020117	MSG21:	.ASCIZ	<15><12>/NO MODE SELECTED./
007643	015	044412	020106	MSG22:	.ASCIZ	<15><12>/IF CALLING,DIAL NUMBER/
007673	015	044412	020106		.ASCIZ	<15><12>/IF ANSWERING,PLACE MODEM IN AUTO-ANSWER/<15><12>
007747	073	041040	042101	MSG23:	.ASCIZ	/; BAD DATA=/
007763	040	020040	047507	MSG24:	.ASCIZ	/ GOOD DATA=/
010001	015	020012	046504	MSG25:	.ASCIZ	<15><12>/ DMBB LINE STATUS REGISTER= /
010040	005015	051124	047101	MSG26:	.ASCIZ	<15><12>/TRANSMITTED DATA=/<15><12>
010066	005015	042522	042503	MSG27:	.ASCIZ	<15><12>/RECEIVED DATA=/<15><12>
010111	015	051412	051127	SWEQ:	.ASCIZ	<15><12>/SWR= /
010121	040	042516	036527	NEQ:	.ASCIZ	/ NEW= /
010130	052536	005015	000	CTLU:	.ASCIZ	/TU/<15><12>
010135	177	000177		MFILL:	.ASCIZ	<177><177>
				.EVEN		

K03

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 36
DZITAC.P11 04-AUG-76 11:27 DN-11 INTERFACE SERVICE ROUTINE

(1)				*****	
1521				DN-11 INTERFACE SERVICE PARAMS	
1522				*****	
1523	010140	047104	000040	DN: .ASCIZ "DN "	
1524	010144	175200		DNBA: 175200	;BUS ADDRESS
1525	010146	000350		DNIV: 350	;INTERRUPT VECTOR
1526	010150	000000		DNPRI: 200	;PRIORITY
1527	010152	177777		DNPAR1: 177777	;NOT USED
1528	010154	177777		DNPAR2: 177777	;NOT USED
1529	010156	010336		DNPAR3: DIALNO	;ADDRESS OF DIAL #
1530					
1531					
1532					
1533		100000		PWI=100000	;POWER INDICATOR
1534		040000		ACR=40000	;ABANDON CALL AND RETRY
1535		010000		DLO=10000	;DATA LINE OCCUPIED
1536		000040		DSS=40	;DATA SET STATUS
1537		000020		PNO=20	;PRESENT NEXT DIGIT
1538		000002		DP=2	;DIGIT PRESENT
1539		000001		CRQ=1	;CALL REQUEST
1540					

```

1541 ;*****
1542 ; START OF DN-11 CODE
1543 ;*****
1544 010160 013704 010144 DNGO: MOV DNBA, R4 ;SETUP BUS ADDR
1545 010164 005014 CLR (R4) ;RESET DN-11
1546 010166 005037 011032 CLR TIME ;RESET TIMER
1547 010172 032737 000002 011032 BIT #2, TIME ;AND WAIT 2 SECS
1548 010200 001774 BEQ -6 ;
1549 010202 012703 010336 MOV #DIALNO, R3 ;SETUP DIAL # ADDRESS
1550 010206 012714 000001 MOV #CRQ, @CSR ;SET CALL REQUEST
1551 010212 032714 100000 BIT #PWI, @CSR ;IS DN AVAILABLE
1552 010216 001425 BEQ DNLI1 ;BR IF YES
1553
1554 010220 011402 MOV @CSR, R2 ;SETUP CSR FOR PRINTOUT
1555 010222 005003 CLR R3
1556 010224 104003 HLT+3 ;PRINTOUT "DN NOT AVAILABLE"
1557 010226 000137 010160 JMP DNGO ;RESTART
1558
1559 010232 032714 000200 DNLI1: BIT #DONE, @CSR ;IS DONE FLAG SET?
1560 010236 001775 BEQ DNLI1 ;WAIT IF NO
1561 010240 042714 000200 BIC #DONE, @CSR ;RESET DONE
1562 010244 032714 140000 BIT #ACR+PWI, @CSR ;ANY ERRORS?
1563 010250 001003 BNE DNLI1E ;BR IF YES
1564
1565
1566 010252 032714 000020 BIT #PND, @CSR ;IS PRESENT NEXT DIGIT SET
1567 010256 001005 BNE DNLI1X ;BR IF YES
1568
1569 010260 011402 DNLI1E: MOV @CSR, R2 ;SETUP CSR FOR PRINTOUT
1570 010262 005003 CLR R3
1571 010264 104004 HLT+4 ;PRINTOUT "DN ERROR"
1572 010266 000137 010160 JMP DNGO ;RESTART
1573
1574 010272 112364 000001 DNLI1X: MOVB (R3)+, 1(R4) ;LOAD NEXT DIGIT
1575 010276 052714 000002 BIS #DP, @CSR ;SET DIGIT PRESENT
1576 010302 105713 TSTB (R3) ;WAS THAT LAST CHAR?
1577 010304 001352 BNE DNLI1 ;BR IF NO
1578
1579 010306 032714 000200 DNLI2: BIT #DONE, @CSR ;WAIT FOR DONE FLAG
1580 010312 001775 BEQ DNLI2
1581
1582 010314 032714 040000 BIT #ACR, @CSR ;WAS CALL ABANDONED?
1583 010320 001405 BEQ DNEX ;BR IF NO
1584
1585 010322 011402 MOV @CSR, R2 ;SETUP CSR FOR PRINTOUT
1586 010324 005003 CLR R3
1587 010326 104005 HLT+5 ;PRINTOUT "CALL ABANDONED"
1588 010330 000137 010160 JMP DNGO ;RESTART
1589
1590 010334 000207 DNEX: RTS PC ;RETURN TO ITP
1591
1592 010336 034470 032467 031460 DIALNO: .ASCIZ "8975030" ;NUMBER TO DIAL
1593 010344 000060
1594 010346 000000 000000 000000 0,0,0,0,0,0
1595 010354 000000 000000 000000

```

1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617010362 046504 000102
010366 170500
010370 000310
010372 000200
010374 000000
010376 177777
010400 177777DMB: .ASCIZ 'DMB'
MBA: 170500
IV: 310
PRIO: 200
PARA1: 0
PARA2: 177777
PARA3: 177777;ISR NAME
;BUS ADDRESS
;VECTOR ADDRESS
;PRIORITY
;PARAM #1
;PARAM #2
;PARAM #3;*****
; DM11-B INTERFACE SERVICE PARAMS
;*****

N03

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 39
 DZITAC.P11 04-AUG-76 11:27 DM11-B INTERFACE SERVICE ROUTINE

```

1618 ;*****
1619 ; DM-11B INTERFACE SERVICE ROUTINE
1620 ;*****
1621 010472 000240 DMBGO: NOP
1622 010404 013704 010356 MOV MBA, R4 ;SETUP BUS ADDR INDEX
1623 010410 005037 011032 CLR TIME ;RESET TIMER
1624
1625
1626 010414 052714 006000 BIS #CS+CM, @CSR ;CLEAR DM-11 B
1627 010420 032714 000020 BIT #BUSY, @CSR ;WAIT TIL FREE
1628 010424 001375 BNE .-4
1629 010426 013714 010374 MOV PARA1, @CSR ;SELECT LINE #
1630 010432 052764 000002 BIS #DTR+LE, 2(R4) ;SET DATA TERM RDY & LINE ENABLE
1631 010440 032737 011000 001562 BIT #20000, FLAGS ;HAS DN11 MADE CONNECTION YET?
1632 010446 001002 BNE IS ;BR IF YES
1633 010450 104400 010343 TYPE ,MSG22 ;TYPE "MAKE CONNECTION"
1634
1635 010454 032777 000005 000362 1S: BIT #OWO+XLB, @SWR ;IS MODE = OWO OR XLB
1636 010462 001444 BEQ REX1 ;BR IF NO
1637
1638
1639 010464 052764 000004 000002 STARTX: BIS #RQTS, 2(R4) ;SET REQUEST TO SEND
1640
1641 010472 032764 000040 000002 CTSW: BIT #CTS, LSTAT(R4) ;IS CLEAR TO SEND SET?
1642 010500 001016 BNE CTSOK ;BR IF YES
1643 010502 023727 011032 000036 CMP TIME, #36 ;30 SECS ELAPSED?
1644 010510 103770 BLO CTSW ;BR IF NO
1645 010512 016446 000002 MOV LSTAT(R4), -(SP) ;TYPE CONTENTS OF RCSR
1646 010516 032777 010000 000320 BIT #SW12, @SWR ;INHIBIT TYPEOUTS?
1647 010524 001001 BNE IS ;BR IF YES
1648 010526 104412 XWAIT ;PRINTOUT 'WAITING FOR CTS'
1649 010530 005037 011032 1S: CLR TIME ;RESET TIMER
1650 010534 000756 BR CTSW ;WAIT SOME MORE
1651
1652 CTSOK:
1653
1654 010536 012737 177777 011024 REX: MOV #-1, SETTLE ;SET UP DELAY FLAG
1655 010544 005037 010576 CLR TEMP1
1656 010550 012737 000014 010600 MOV #14, TEMP2
1657 010556 062737 000001 010576 ADD #1, TEMP1
1658 010564 001374 BNE .-6
1659 010566 005337 010600 DEC TEMP2
1660 010572 001371 BNE .-14
1661 010574 000207 REX1: RTS PC ;RETURN TO CONTROL PROGRAM
1662
1663 010576 000000 TEMP1: 0
1664 010600 000000 TEMP2: 0
1665 000001 OWO=1
1666 000002 OWI=2
1667 000004 TLB=4
1668 000004 XLB=4
1669 000010 ILB=10
1670
1671 ; RCSR EQUATES
1672 .EQUIV R4, RCSR
1673 .EQUIV R4, CSR

```

1674			
1675	100000	RI=100000	;RING INDICATOR
1676	000100	CF=100	;CARRIER FLAG
1677	000040	CTS=40	;CLEAR TO SEND
1678	010000	SRD=10000	;SEC. RECEIVE DATA
1679	004000	CS=4000	;CLEAR SCAN
1680	002000	CM=2000	;CLEAR MUX
1681	001000	MM=1000	;MAINT MODE
1682	000400	STEP=400	;STEP
1683	000200	DONE=200	;DONE
1684	000100	IE=100	;INTERRUPT ENABLE
1685	000040	SE=40	;SCAN ENABLE
1686	000020	BUSY=20	
1687	000017	LINE=17	;BUST ;LINE NUMBER
1688			
1689		;LINE STATUS REGGRSTER EQUATES	
1690	000002	LSTAT=2	
1691			
1692	000004	RTS=4	;REQUEST TO SEND
1693	000002	DTR=2	;DATA TERMINAL READY
1694	000001	LE=1	;LINE ENABLE
1695			

```

1696      011000
1697
1698
1699
1700
1701
1702
1703
1704
1705      011000
1706      011000      000002
1707      011004      000001
1708      011006      000001
1709      011010      000001
1710      011012      000001
1711      011014      000001
1712      011016      000001
1713
1714      011020      000001
1715      011022      000001
1716      011024      000001
1717      011026      000001
1718      011030      000001
1719      011032      000001
1720      011034      000001
1721      011036      000001
1722      011040
1723      011040      000001
1724      011041
1725      011041      000001
1726
1727      011042      000001
1728      011044      177570
1729      011046      177570
1730      011050      000001
1731      011052      000001
1732      011054      000001
1733      011056      000001
1734      011060      000001
1735      011062      000001
1736      011064      000001
1737      011066      000001
1738      011070      000001
1739      011072      000001
1740      011074      000001
1741
1742      011076      000001
1743      011100      000001
1744      011102      000001
1745

```

```

      .=11000
*****
      THE INTERFACE SERVICE ROUTINE IS LOADED HERE
*****
      THE FOLLOWING 18 WORDS ARE USED AS
      THE LINKAGE BETWEEN THE ISR AND THE
      CONTROL PROGRAM.

VISR:
ISR:      .BLKW 2  :.ASCIZ "DXX"
BA:      .BLKW 1  :175610      :BUS ADDRESS
VA:      .BLKW 1  :300         :VECTOR ADDRESS
PRIOR:   .BLKW 1  :340         :PRIORITY
PARAM1:  .BLKW 1  :-1         :PARAM #1
PARAM2:  .BLKW 1  :-1         :PARAM #2
PARAM3:  .BLKW 1  :-1         :PARAM #3

IRDA:    .BLKW 1  :.WORD 0     :INITIAL READ DATA ADDRESS
IXDA:    .BLKW 1  :.WORD 0     :INITIAL XMIT DATA ADDRESS
SETTLE:  .BLKW 1  :.WORD 0     :LINE SETTLE DELAY FLAG
IRCC:    .BLKW 1  :.WORD 0     :INITIAL RCV CHAR COUNT
B2016:   .BLKW 1  :.WORD 0     :ADDR OF BIN TO OCT TYPE ROUTINE
TIME:    .BLKW 1  :.WORD 0     :TIMER
MODEA:   .BLKW 1  :.WORD 0     :ADDR OF ITP PARAMS
          .BLKW 1  :.WORD START:ISR ENTRY ADDRESS

TX. TERM: .BLKB 1 :.BYTE 000    :TRANSMITTER TERMINATING CHAR.
RX. TERM: .BLKB 1 :.BYTE 012    :RECEIVER TERMINATING CHAR.

FLAG:    .BLKW 1
SWR:     177570
DISPLAY: 177570
START:   .BLKW 1 :NOP
          .BLKW 1 :NOP
          .BLKW 1 :NOP
          .BLKW 1 :NOP
CONT.:   .BLKW 1 :NOP
          .BLKW 1 :NOP
          .BLKW 1 :NOP
          .BLKW 1 :NOP
FINI:    .BLKW 1 :MOV  #340,PS    :LOCK OUT INTERRUPTS.
          .BLKW 1 :MOV  #ENTER,2(SP) :SET FOR RETURN IF SW14=0
          .BLKW 1 :JSR  PC,SAVE05 :GO SAVE YOUR REGISTERS.
          .BLKW 1 :RTS  PC        :EXIT

ENTER:   .BLKW 1 :JSR  PC,REST05  :GO AND RESTORE REGISTERS
          .BLKW 1 :MOV  #-1,DELAY :INDICATE DELAY FOR TX.
          .BLKW 1 :JMP  CONT.     :CONTINUE IN PROGRAM
          .END

```

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 43
DZITAC.P11 04-AUG-76 11:27 CROSS REFERENCE TABLE -- USER SYMBOLS

ACCEPT=	104406	707	1123	1240	1415#	1486				
ACR =	040000	1534#	1562	1582						
ANYEX	005012	1125#								
ANYNOR	005002	1123#								
ARIA	003734	910#	914#	923						
ASTRK	001532	684	760#							
BA	011004	1707#								
BEGIN	003254	639	749#	810	898	956				
BIT0 =	000001	615#	943	1478	1492					
BIT00 =	000001	605#	615							
BIT01 =	000002	604#	614							
BIT02 =	000004	603#	613							
BIT03 =	000010	602#	612							
BIT04 =	000020	601#	611							
BIT05 =	000040	600#	610							
BIT06 =	000100	599#	609							
BIT07 =	000200	598#	608							
BIT08 =	000400	597#	607							
BIT09 =	001000	596#	606							
BIT1 =	000002	614#	964	1488	1492					
BIT10 =	002000	595#								
BIT11 =	004000	594#								
BIT12 =	010000	593#								
BIT13 =	020000	592#	1140							
BIT14 =	040000	591#	1100							
BIT15 =	100000	590#	1188							
BIT2 =	000004	613#								
BIT3 =	000010	612#	690	722						
BIT4 =	000020	611#	694							
BIT5 =	000040	610#	699	710						
BIT6 =	000100	609#								
BIT7 =	000200	608#								
BIT8 =	000400	607#								
BIT9 =	001000	606#								
BPTVEC=	000014	622#								
BUSY =	000020	1627	1686#							
BZ016	011030	1057#	1718#							
CF =	000100	1676#								
CLRIB	004374	1027#	1029							
CM =	002000	1626	1680#							
CONT.	011060	1734#								
CRLF	001536	703	718	761#	974	1087	1143	1171	1186	1296
CRQ =	000001	1539#	1550							
CS =	004000	1626	1679#							
CTLU	010130	1310	1520#							
CTS =	000040	1641	1677#							
CTSOK	010536	1642	1652#							
CTSM	010472	1641#	1644	1650						
DAT	001564	775#	1006							
DIALNO	010336	1529	1549	1592#						
DISPLA	011046	1508#	1729#							
DISPRE	000174	634#	1508							
DLO =	010000	1535#								
DMB	010362	830	878	879	1601#					
DMEGO	010402	1069	1621#							
DNETST	003574	874	877#							

[illegible]

F04

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 45
 DZITAC.P11 04-AUG-76 11:27 CROSS REFERENCE TABLE -- USER SYMBOLS

GETPRM	003672	902#					
GETP3	003716	901	906	910#			
GETSTR=	104404	868	917	1015	1413#		
GETVA	003646	894#					
GOTIC	006026	1238	1245#				
HUTX	005250	1189	1192#				
IBUF	001604	785#	918	919	949	1017	1026 1028
IE =	000100	1684#					
ILB =	000010	1669#					
IOTVEC=	000020	623#					
IRCC	011026	1717#					
IRDA	011020	950	1017#	1097	1714#		
ISR	011000	1075	1706#				
IV	010370	1603#					
IXDA	011022	951	1006#	1021#	1091	1715#	
KSDIN =	104416	1081	1192	1423#			
LE =	000001	1630	1694#				
LF	001537	762#	1250	1298			
LINE =	000017	1687#					
LSTAT =	000002	1641	1645	1690#			
L1	001374	711	713	716#	733	741	952
L2	001426	725#	732				
L3	001452	723	735#				
MANBA	003434	828	833	836	840#		
MANIN	003346	822#					
MANINX	003510	850	854	857#			
MBA	010366	1602#	1622				
MFILL	010135	958	1520#				
MM =	001000	1681#					
MODEA	011034	1720#					
MSECS	006322	1034#	1316#	1318#	1322#		
MSG00	007132	727	737	1520#			
MSG01	007134	864	1520#				
MSG02	007157	885	1520#				
MSG03	007202	891	1520#				
MSG04	007221	894	1520#				
MSG05	007243	897	1520#				
MSG06	007257	902	1520#				
MSG07	007274	907	1520#				
MSG08	007311	912	1520#				
MSG09	007330	965	968	1520#			
MSG10	007352	1014	1520#				
MSG11	007371	1084	1520#				
MSG12	007410	1148	1520#				
MSG13	007421	1177	1520#				
MSG14	007433	1183	1520#				
MSG15	007442	1433	1520#				
MSG16	007465	1436	1520#				
MSG17	007531	1520#					
MSG18	007544	1449	1520#				
MSG19	007550	1050	1520#				
MSG20	007576	1474	1520#				
MSG21	007617	1003	1520#				
MSG22	007643	1520#	1633				
MSG23	007747	1159	1520#				
MSG24	007763	1166	1520#				

G04

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 46
 DZITAC.P11 04-AUG-76 11:27 CROSS REFERENCE TABLE -- USER SYMBOLS

MSG25	010001	1441	1520#				
MSG26	010040	1090	1520#				
MSG27	010066	1096	1520#				
NEQ	010121	1484	1520#				
NOISR	003620	880	884#				
NOLC	004444	1037	1043#				
NOQ	004120	947	954#				
NORTC	004512	1039	1049	1052#			
NULL	001524	667	753#				
OCT	006004	1240#	1243				
OUT	007026	1479	1493#				
OWI =	000002	1088	1666#				
OWO =	000001	1094	1635	1665#			
PARAM1	011012	1710#					
PARAM2	011014	1711#					
PARAM3	011016	1712#					
PARA1	010374	1605#	1629				
PARA2	010376	1606#					
PARA3	010400	1607#					
PNO =	000020	1537#	1566				
PRI0	010372	1604#					
PRIOR	011010	1709#					
PRTY0 =	000000	580#	1056	1103	1475		
PRTY1 =	000040	581#					
PRTY2 =	000100	582#					
PRTY3 =	000140	583#	1133				
PRTY4 =	000200	584#					
PRTY5 =	000240	585#					
PRTY6 =	000300	586#					
PRTY7 =	000340	587#	1495				
PWI =	100000	1533#	1551	1562			
RVVEC=	000024	624#					
QUES	001526	702	759#	866	915	1121	1295
REST	004232	967	998#				
RESTR	004240	998	999#				
RESVEC=	000010	619#					
REX	010536	1654#					
REX1	010574	1636	1661#				
RI =	100000	1675#					
ROTS =	000004	1639	1692#				
RSTART	004132	955	958#	1004			
RWAIT =	104410	1417#					
RX.TER	011041	1724#					
SE =	000040	1685#					
SETSWI=	104422	966	1427#				
SETTLE	011024	1654#	1716#				
SFO =	010000	1678#					
STACK =	001070	533#	800	1055	1105	1473	
START	011050	1730#					
STARTX	010464	1639#					
STEP =	000400	1682#					
STPS =	104414	1056	1103	1133	1421#	1475	
SUSUR =	104420	808	1425#				
SUTIME	004404	1034#					
SUXCC	004322	1008	1010	1017#			
SWE0	010111	1481	1520#				

[illegible]

J04

MAINDEC-11-DZITA-C INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 29-OCT-76 14:35 PAGE 50
DZITAC.P11 04-AUG-76 11:27 CROSS REFERENCE TABLE -- MACRO NAMES

BOX	6298	678	754	796	813	819	860	976	1011	1060	1076	1116	1129	1210	1224
	1313	1429	1520	1541	1596	1618	1697								
HELLO	18														
HLT	5368	1556	1571	1587											

. ABS. 011104 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

.DZITAC.SEG/SOL/CRF/NL:TOC=DZITAC.P11
RUN-TIME: 9 14 1 SECONDS
RUN-TIME RATIO: 54/26=2.0
CORE USED: 11K (21 PAGES)

