

DZV11

DZV11 DIAG PRT 1
CVDZADO

AH-A878D-MC
FICHE 1 OF 1

JUL 1983
COPYRIGHT © 77-83
MADE IN USA



A grid of approximately 15 columns and 15 rows of small, dense data tables. Each cell in the grid contains a small table with multiple columns and rows of text, likely representing diagnostic data or test results. The text is too small to read clearly but appears to be organized in a structured format.

5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709

.REM 8

IDENTIFICATION

PRODUCT CODE: AC-A877D-MC
PRODUCT NAME: CVDZADO DZV-11 DIAG PRT1
DATE RELEASED: MARCH 1983
MAINTAINER: DIAGNOSTIC ENGINEERING

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) 1977,1983 DIGITAL EQUIPMENT CORPORATION

5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766

1. ABSTRACT

THE FUNCTION OF THE DZV11 DIAGNOSTICS IS TO VERIFY THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS ALSO VERIFY THAT THE DZV11 OPERATES IN ITS ENVIRONMENT SUCH AS THE SYSTEM IN WHICH IT IS INSTALLED.

PARAMETERS MAY BE SUPPLIED TO THE PROGRAM BY EITHER 'AUTO SIZING' OR INPUT FROM THE USER ON THE CONSOLE BY HAVING SW00=1 AT START TIME. AUTO SIZING WILL BE DONE ONLY THE FIRST TIME THE PROGRAM IS STARTED AND SW07=0 AND SW00=0 AND SW03=0. THE AUTOSIZER IS DESIGNED TO DETECT DZV11 DEVICE ADDRESSES AND VECTORS ONLY. ALL REMAINING PARAMETERS WILL DEFAULT TO CERTAIN VALUES (SEE SEC.8.5). CONSOLE INPUT MAY BE CONTROLLED AT ANY START TIME THROUGH THE USE OF SW00, SW03, SW04, AND SW06 (SEE SEC. 4.1.1 FOR A DETAILED DESCRIPTION OF THESE SWITCHES).

CURRENTLY THERE ARE THREE STANDALONE DIAGNOSTICS (CVDZA, CVDZB, AND CVDZC) ONE SYSTEM MODULE FOR DEC X/11 (CXDZBA), AND AN OVERLAY FOR ITEP (CVDZD).

CVDZA TOGETHER WITH CVDZB WILL TEST ALL LOGICAL FUNCTIONS OF THE DZV11 INTERFACE MODULE.

CVDZC IS DESIGNED AS A NON-CHAINABLE STANDALONE DIAGNOSTIC PROVIDING THE OPERATOR WITH DIRECT CONTROL OVER THE TESTING OF ALL DZV11 EIA CABLES.

```

*****
*
* NOTE: THIS DIAGNOSTIC HAS BEEN MODIFIED TO RUN IN KXT11 (SBC 11/21)
* BASED SYSTEMS. THE PROGRAM WILL AUTOMATICALLY ADJUST ITSELF TO RUN
* IN THE APPROPRIATE ENVIRONMENT AS FOLLOWS:
*
*
*          LSI-11, 11/2, AND 11/23          SBC 11/21
*          -----
* CSR RANGE:          160010 TO 167770          174000 TO 177770
* VECTOR RANGE:          300 TO 770          300 TO 370
* AUTO-SIZING FOR...
* ...CSR AND VECTOR:   ENABLED          DISABLED
*
*
*****
::GPA

```

THE RELEASE CVDZAD WAS DUE TO THE FACT THAT THE DZV11 DIAGNOSTIC WAS UPDATED TO SUPPORT THE USER FRIENDLY DIAGNOSTIC INTERFACE.

2. REQUIREMENTS

2.1 EQUIPMENT

AN LSI11 CPU WITH MINIMUM 4K OF MEMORY.
ASR 33 (OR EQUIVALENT FOR CONSOLE)
DZV11 INTERFACE MODULE
H329 STAGGERED TURNAROUND CONNECTOR.

5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822

H325 CABLE TURNAROUND CONNECTOR.

NOTE: A STAGGERED TURNAROUND CONNECTOR IS NEEDED IN ORDER TO TEST THE PARITY LOGIC.

2.2 STORAGE

PROGRAM WILL USE ALL 4K OF MEMORY EXCEPT WHERE ABL AND BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1740 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER PARAMETERS HAVE BEEN INPUT FROM CONSOLE (SW00=1); OR AFTER THE 'AUTO SIZING' HAS BEEN DONE. THESE LOCATIONS MAY BE CHANGED IF THE USER UNDERSTANDS THEIR MEANING AND DIFFERENT PARAMETERS ARE REQUIRED.

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA SUCH AS DISK ,MAGTAPE,DECTAPE, OR CASSETTE; FOLLOW INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY * SIZE

4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

3.1.1 STARTING THE PROCESSOR AT THE ABSOLUTE LOADER STARTING ADDRESS WILL LOAD THE DIAGNOSTIC INTO MEMORY.

4. STARTING PROCEEDURE

A. SET SWR TO ZERO FOR 'AUTO SIZING' OR SET SW00=1 FOR USER PARAMETER INPUT FROM CONSOLE TERMINAL. NOTE: LOC. 000176 IS USED AS A SOFTWARE SWITCH REGISTER IN ALL OF THE DZV11 DIAGNOSTICS. (SEE SEC. 4.1) ON THE FIRST STARTUP OF THE DIAGNOSTIC IF SW07=1 AND SW00=0 THE PROGRAM WILL ASSUME THAT THE STATUS TABLE HAS BEEN ALREADY BUILT FROM A PREVIOUS DZV11 DIAGNOSTIC RUN. NOTE: ANY DZV11 DIAGNOSTIC WILL OVERLAY THE STATUS TABLE WHEN LOADED TO PRESERVE ITS CONTENTS

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-2
 CVDZAD.P11 11-MAR-83 10:06

5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878

AND THUS WILL NOT ALTER A PREVIOUSLY BUILT TABLE.
 B. START THE DIAGNOSTIC AT LOC. 200(8). THE PROGRAM WILL TYPE MAINDEC AND PROGRAM NAMES (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING: (ON THE FIRST PROGRAM RUN OR IF PARAMETERS WERE CHANGED)

```
'MAP OF DZV11 STATUS'
1500 160100
1502 000300
1504 000017
1506 017470
1510 000000
```

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.

THE PROGRAM WILL TYPE 'RUNNING' AND PROCEED TO RUN THE DIAGNOSTIC.

4.1 CONTROL SWITCH SETTINGS

NOTE: THIS PROGRAM UTILIZES A SOFTWARE SWITCH REGISTER WHICH MAY BE MODIFIED BY CHANGING LOC. 176 OR BY TYPING CONTROL 'G' (^G) ON THE CONSOLE TERMINAL WHILE THE PROGRAM IS RUNNING.

```
SW 15 SET: HALT ON ERROR
SW 14 SET: LOOP ON CURRENT TEST
SW 13 SET: INHIBIT ERROR PRINT OUT
SW 12 SET: INHIBIT **ALL** TYPE OUT/BELL ON ERROR.
SW 11 SET: INHIBIT ITERATIONS. (QUICK PASS)
SW 10 SET: ESCAPE TO NEXT TEST
SW 09 SET: LOOP WITH CURRENT DATA
SW 08 SET: CATCH ERROR AND LOOP ON IT
SW 07 SET: NO AUTO SIZE. IF 1ST START OF PROGRAM AFTER LOADING AND
IF SW00=0 THEN THE PROGRAM WILL ASSUME THAT THE STATUS MAP
HAS BEEN BUILT FROM A PREVIOUS DZV11 DIAGNOSTIC RUN.

SW 06 SET: RESELECT DZV11'S DESIRED ACTIVE
SW 05 SET: RESERVED
SW 04 SET: SELECT DELAY PARAMETER (SEE SEC. 4.1.1)
SW 03 SET: EXTRA PARAMETER INPUT (SEE SEC. 4.1.1)
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: GET USERS PARAMETERS FROM CONSOLE
```

4.1.1 SWITCH REGISTER CONTROL OF PARAMETER INPUT FROM CONSOLE

```
SW 00 GET USERS PARAMETERS FROM CONSOLE. SETTING THIS SWITCH AT START
UP TIME ALLOWS THE USER TO INPUT AT THE CONSOLE TERMINAL THE
FOLLOWING PARAMETERS: BASE DEVICE ADDRESS, BASE VECTOR ADDRESS,
MODE OF OPERATION (EXTERNAL, INTERNAL, OR STAGGERED), AND THE
NUMBER OF DZV11'S THAT ARE RUNNING. USING THIS SWITCH ALONE WILL
DEFAULT THE FOLLOWING PARAMETERS: ALL 4 LINES ARE SET TO BE
```

5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934

TESTED ON EACH DZV11, THE DEFAULT BAUD RATE IS SET AT 19.2 KBAUD AND THE CHARACTER LENGTH FOR THE MAJORITY OF TESTING IS SET AT EIGHT BITS PER CHARACTER WITH TWO STOP BITS.

SW 03 EXTRA PARAMETER INPUT. SETTING THIS SWITCH AT START UP TIME PROVIDES THE USER WITH THE ABILITY TO SET THE LINES ACTIVE FOR TESTING AND TO SET THE DEFAULT BAUD RATE USED FOR THE MAJORITY OF THE DIAGNOSTIC TESTS. THE DELAY PARAMETER IS AUTOMATICALLY ADJUSTED TO THE BAUD RATE GIVEN BY THE USER.

SW 04 SELECT DELAY PARAMETER. THE DELAY PARAMETER THIS SWITCH CONTROLS DETERMINES THE LENGTH OF TIME THE PROGRAM STALLS WAITING FOR A CHARACTER TO BE COMPLETELY TRANSMITTED OR RECEIVED. THIS DELAY COUNT IS AUTOMATICALLY SET TO PROVIDE ENOUGH DELAY TIME FOR THE DEFAULT BAUD RATE SPECIFIED WHEN RUNNING THE PROGRAM ON AN LSI11 WITH MOS MEMORY. WHEN RUNNING THIS PROGRAM ON A PROCESSOR WITH A FASTER MEMORY SPEED THIS DELAY COUNT SHOULD BE ADJUSTED PROPORTIONATELY HIGHER THAN THE FOLLOWING DEFAULTED VALUES:

2450	:TIME FOR	50 BAUD
1560	:TIME FOR	75 BAUD
1120	:TIME FOR	110 BAUD
0750	:TIME FOR	134 BAUD
0660	:TIME FOR	150 BAUD
0330	:TIME FOR	300 BAUD
0150	:TIME FOR	600 BAUD
0060	:TIME FOR	1200 BAUD
0040	:TIME FOR	1800 BAUD
0030	:TIME FOR	2000 BAUD
0020	:TIME FOR	2400 BAUD
0010	:TIME FOR	3600 BAUD
0001	:TIME FOR	4800 BAUD
0001	:TIME FOR	7200 BAUD
0001	:TIME FOR	9600 BAUD
0001	:TIME FOR	19.2 KBAUD

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 06 RESELECT DZV11'S DESIRED ACTIVE. A MESSAGE IS TYPED OUT ON THE CONSOLE TERMINAL ASKING THE OPERATOR TO TYPE A BIT MAP OF THE DZV'S DESIRED ACTIVE. USING THIS SWITCH ALLOWS LOCATION DZVACTV TO BE ALTERED (SEE SEC. 8.3 FOR A DESCRIPTION OF THIS LOCATION).
 EXAMPLE:
 IF THE DEVICES CORRESPONDING TO THE DZV11'S NUMBERED ZERO, TWO, AND FOUR IN THE DZV11 STATUS MAP (LOC. 1500 THROUGH 1740) ARE TO BE TESTED, TYPE IN: 25
 THIS WILL SET BITS ZERO, TWO, AND FOUR IN LOCATION DZVACTV. ALL REMAINING DEVICES IN THE STATUS MAP WILL THEN NOT BE TESTED.

SW 01 RESTART PROGRAM AT SELECTED TEST IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE THE REASON BEING IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS.

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-4
 CVDZAD.P11 11-MAR-83 10:06

5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990

NOTE: IF RUNNING MULTIPLE DZV11'S; THE DZV11 YOU DESIRE TO BE UNDER TEST MUST BE SELECTED BY THE USE OF SW06 BEFORE LOCKING ON THE TEST. IN OTHER WORDS; EACH TIME THE PROGRAM IS STARTED; THE FIRST DZV11 WILL BE SELECTED TO BE UNDER TEST UNLESS SW06 IS USED TO SELECT ONLY ONE.

- SW 09 LOOP ON CURRENT DATA: THIS SWITCH WILL ONLY WORK IF CALL 'SCOPI' IS IN THAT TEST. THE REASON BEING THAT MOST TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE THUS IN BLOCK DATA, ONE PATTERN CAN'T BE SINGLED OUT. THIS SWITCH IS DESIGNED TO PROVIDE AN AID FOR A TRAINED TROUBLE-SHOOTER TO SAMPLE VARIOUS SIGNALS ON THE MODULE AND IS NOT MEANT TO BE USED AS A GENERAL USER CONTROL SWITCH.
- SW 04 SELECT DELAY PARAMETER: THIS SWITCH SHOULD BE USED WITH CARE AS TOO SHORT A DELAY WILL CAUSE VALID TESTS TO FAIL. (SEE SEC. 4.1.1)

4.1.3 SWITCH REGISTER PRIORITIES

ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 08 GO TO BEGINNING OF THE TEST(ON ERROR).
5. SW 10 GOTO NEXT TEST(ON ERROR).

SCOPE SWITCHES

1. SW 09 (IF ENABLED BY 'SCOPI'). IF AN '*' IS PRINTED IN FRONT OF THE TEST NO. ON AN ERROR REPORT (EX. *TEST NO. 10) SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS *USUALLY* THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0) IF THE PROGRAM USER IS TECHNICALLY TRAINED TO ELECTRONICALLY ISOLATE SIGNAL PROBLEMS ON THE DZV11 MODULE. IF SW09 IS NOT ENABELED; AND THERE IS A *HARD* ERROR (CONSTANT); SW08 IS BEST.
2. FOR INTERMITTENT ERRORS EITHER START THE PROGRAM WITH SW01 AND SW02 SET WHICH WILL ALLOW THE USER TO LOCK ON A SELECTED TEST, OR ELSE SET SW14 AS AN ERROR IS BEING TYPED OUT ON THE TERMINAL. SW14 WILL CONTINUE TO LOOP ON THAT TEST REGARDLESS OF WHETHER AN ERROR OCCURS.
3. SW 14 LOOP ON CURRENT TEST.

4.2 STARTING ADDRESS

SA 200 - THE STARTING ADDRESS FOR ANY DZV11 DIAGNOSTIC IS LOC. 200

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY. AFTER *ALL* AVAILABLE DZV11S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-5
 CVDZAD.P11 11-MAR-83 10:06

5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046

5. OPERATING PROCEEDURE

WHEN THE PROGRAM IS INITIALLY STARTED, MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED AND THE DIAGNOSTIC WILL BEGIN RUNNING.

5.1 NORMAL START OF DIAGNOSTIC

ON THE FIRST START OF THE DIAGNOSTIC AT ADDRESS 200, IF SW00=1 THEN THE FOLLOWING QUESTIONS ARE ASKED AND MUST BE ANSWERED:

"1ST CSR ADDRESS (160000:167770): "
 YOU MUST TYPE IN THE FIRST DZV11 CSR IN THE SYSTEM YOU WISH TESTING TO BEGIN AT. RANGE: 160000:167770

"1ST VECTOR ADDRESS (300:770): "
 YOU MUST TYPE IN THE VECTOR OF THE FIRST DZV11 IN THE SYSTEM UNDER TEST. RANGE 300:770

"MAINTENANCE MODE
 [EXTERNAL <H325> (E)]
 [INTERNAL <DZCSR03=1>(I)]
 [STAGGERED <H329> (S)] :
 TYPE "E" OR "I" OR "S" DEPENDING ON WHICH MODE YOU WISH TO RUN IN. IF RUNNING "EXTERNAL"; ALL SELECTED LINES MUST BE TERMINATED BY AN H325 TEST CONNECTOR.

"# OF DZV11'S <IN OCTAL> (1:20): "
 TYPE TOTAL NUMBER OF DZV11'S TO BE TESTED IN THE SYSTEM. RANGE IS 1 THRU 20 IN OCTAL.

***** IF SW03=1 THEN THE FOLLOWING WILL BE PRINTED *****

"LINES ACTIVE BY BIT <IN OCTAL> (001:017):"
 EACH BIT REPRESENTS A LINE AND ANY COMBINATION OF LINES MAY BE SELECTED (HOWEVER IN STAGGERED MODE TWO ADJACENT LINES MUST BE SELECTED (0-1, 2-3).

"DEFAULT BAUD RATE <IN OCTAL> (00:17): "
 THIS GIVES THE USER A CHANCE TO CHANGE THE DEFAULT BAUD RATE USED IN APP. 90% OF THE TEST. BAUD RATE CHOICES ARE:
 "00"(50 BAUD), "01"(75 BAUD), "02"(110 BAUD), "03"(134 BAUD),
 "04"(150 BAUD), "05"(300 BAUD), "06"(600 BAUD), "07"(1200 BAUD),
 "10"(1800 BAUD), "11"(2000 BAUD), "12"(2400 BAUD), "13"(3600 BAUD),
 "14"(4800 BAUD), "15"(7200 BAUD), "16"(9600 BAUD), "17"(19.2 KBAUD)
 LOW DEFAULT BAUD RATES ARE NOT SUGGESTED SINCE THEY LENGTHEN THE TIME TO COMPLETE A PROGRAM PASS DRAMATICALLY.

IT IS IMPORTANT TO NOTE THAT ALL DZV11'S IN THE SYSTEM MUST BE CONTIGIOUS FOR BOTH ADDRESS AND VECTORS. ALSO ALL THE EXTRA PARAMETERS OTHER THAN CSR AND VECTORS ARE GIVEN TO THE EXISTING DZV11'S IN THE SYSTEM.

6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102

IF THE MODE OF OPERATION IS DIFFERENT FOR EACH DZV11 THIS MUST BE PATCHED INTO THE CORRECT STATUS MAP ENTRY WHICH IS PRINTED AT START TIME. AN ALTERNATIVE IS TO PUT SW00=1 AT START TIME; ANSWER QUESTIONS ABOUT DZV11 UNDER TEST AND INDICATE ONE DZV11 IN THE SYSTEM. IF THE STATUS MAP IS TO BE 'PATCHED' IT MUST BE DONE AFTER THE QUESTIONS ARE ANSWERED OR AFTER THE AUTO SIZE.

5.2 PROGRAM AND/OR OPERATOR ACTION

THE VARIETY OF PROGRAM CONTROL SWITCHES PROVIDED IN THIS DIAGNOSTIC PACKAGE IS DESIGNED TO PROVIDE THE USER WITH A WIDE RANGE OF TROUBLE-SHOOTING TECHNIQUES. BEFORE THE USER ATTEMPTS TO RUN THIS DIAGNOSTIC HE SHOULD BECOME FAMILIAR WITH THE USE OF THESE CONTROL SWITCHES AND THEIR RESTRICTIONS. (SEE SEC. 4.1, 4.1.1, 4.1.2, 4.1.3)

WHEN THE PROGRAM DETECTS AN ERROR THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (DEPENDING ON THE PARTICULAR ERROR). IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT THEN LOOK IN THE PROGRAM LISTING FOR THAT TEST NUMBER AND THEN NOTE THE PC OF THE ERROR REPORT. THE REASON FOR THE ERROR REPORT WILL BECOME CLEARER WHEN READING THE COMMENTS IN THE PROGRAM LISTING.

6. ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.1 ERROR RECOVERY

IF FOR SOME REASON THE DZV11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN, LOOK IN LOCATION '\$TSTNM' (ADDRESS 1246) FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DZV11 WAS DOING AT THE TIME OF THE ERROR.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4.1.2
THE STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW THE PROGRAM WAS STARTED. ALSO IT IS IMPORTANT TO USE THIS LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETELY ISOLATE PROBLEMS.

7.2 OPERATING RESTRICTIONS

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-7
 CVDZAD.P11 11-MAR-83 10:06

6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158

PARAMETER MUST BE INPUT FROM USER OR APT IF "AUTO SIZING" IS NOT USED.

8. MISCELLANEOUS

8.1 EXECUTION TIME

ALL DZV11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 2 MIN. THIS IS ASSUMING SW11=1 (INHIBIT ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED; THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO 'VERIFY NO *HARD* ERRORS' AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS -EACH TIME PROGRAM IS STARTED- WILL BE A 'QUICK PASS' UNTIL ALL DZV11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS CVDZA-D CSR: 160100 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

8.3 KEY LOCATIONS

SLPADR (1252) CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT (1362) CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

STSTNM (1246) CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN (1412) THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DZV11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) 1412/0000000001000000 MEANS THAT DZV11 NO.5 IS THE DZV11 NOW RUNNING.

STATUS MAP
(1500)-(1740)

THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 16 (DECIMAL) DZV11S SEQUENTIALY. THEY CONTAIN THE CSR,VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DZV11.

DZVACTV(1406) EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DZV11 WILL BE TESTED IN TURN. EXAMPLE: (DZVACTV) 1406/0000000000011111 MEANS THAT DZV11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DZVACTV) 1406/0000000000010001 MEANS THAT DZ11 NO. 00,04 WILL BE TESTED.

SBASE (1174) CONTAINS THE RECEIVER CSR OF THE CURRENT DZV11 UNDER TEST.

6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214

8.4 MORE ON THAT 'STATUS TABLE' (1500-1740)

'MAP OF DZV11 STATUS'	
1500	160100
1502	000300
1504	000017
1506	017470
1510	000000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 16 DZV11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE). EXPLANATION:

1500	160100	THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DZV11 IN THE SYSTEM.
1502	000300	THIS IS VECTOR 'A' FOR THE FIRST DZV11 IN THE SYSTEM.
1504	000017	THIS IS THE BINARY REPRESENTATION OF WHAT LINES ARE TO BE TESTED.
1506	017470	THIS IS THE PARAMETER LOCATION USED IN MOST OF THE TESTS. IT INDICATES PARAMETERS OF: RX ON, SPEED SELECT 17 (19.2K BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS. THE USER MAY ALTER THE STOP BITS AND THE SPEED, BUT THE REMAINING PARAMETERS SHOULD BE LEFT ALONE. THIS LOCATION IS USED TO LOAD THE DZV11 LINE PARAMETER REGISTER FOR EACH LINE. THE MEANING OF THE BITS SET IN THIS LOCATION IS THE SAME AS THE FUNCTION OF THE RELATED BITS IN THE DEVICE LINE PARAMETER REGISTER.
1510	000000	THIS LOCATION WILL CONTAIN EITHER ALL ZEROS INDICATING THAT INTERNAL LOOP WAS SELECTED AS MODE OF OPERATION OR IT WILL CONTAIN 100000 INDICATING THAT "STAGGERED MODE" WAS SELECTED OR IT WILL CONTAIN 000200 INDICATING THAT "EXTERNAL" WAS THE MODE SELECTED.

THE ABOVE IS REPEATED FOR EACH DZV11 IN THE SYSTEM. THE TABLE IS FILLED BY AUTO SIZING OR BY THE MANUAL PARAMETER INPUT PROGRAM AS DESCRIBED PREVIOUSLY. ALSO IF DESIRED BY USER; THE LOCATIONS MAY BE ALTERED BY HAND TO SUIT THE SPECIFIC CONFIGURATION.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

THE PROGRAM WILL START AT ADDRESS 160000 AND START 'REFERENCING' THE ADDRESS IN THE POINTER. IF A NON-EX MEMORY TRAP OCCURS, THE POINTER (HOLDING 160000) IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTIL ADDRESS 167770 IS REACHED. IF A 'BUS REPLY' RESPONSE WAS ISSUED BY THE DZV11 (OR ANY OTHER DEVICE) (NO NXM TRAP), 'MASTER SCAN ENABLE' IS ATTEMPTED TO BE SET AND THE TCR BITS FOR ALL FOUR LINES ARE SET. 'TRDY' IS THEN TESTED TO BE SET AND 'MASTER SCAN ENABLE' IS TESTED TO BE STILL

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-9
 CVDZAD.P11 11-MAR-83 10:06

6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270

SET. THE DIAGNOSTIC WILL THEN CHECK THAT AT LEAST ONE TCR BIT IS STILL SET. IF ALL OF THE ABOVE WORKED, THIS DEVICE IS ASSUMED TO BE A DZV11. IF ANY OF THE ABOVE FAILED, UPDATING OF THE POINTER IS DONE AND THE SEQUENCE IS REPEATED.

NOTE: IF THE PROGRAM DOES NOT FIND YOUR DZV11, SOMETHING IS WRONG AND AUTO SIZING SHOULD NOT BE DONE.

8.5.2 FINDING THE VECTOR

THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND '+2' (NEXT ADDRESS). BIT14 AND BIT5 (TX INTERRUPT ENABLE AND MSTSCAN ENABLE) ARE SET INTO THE DZVCSR. ALL TCR BITS ARE SET, A DELAY OCCURS, AND IF NO INTERRUPT OCCURS (BECAUSE OF A BAD DZV11) THE PROGRAM ASSUMES VECTOR ADDRESS 300 AND THE PROBLEM SHOULD BE FIXED IN THE DIAGNOSTIC. ONCE THE PROBLEM IS FIXED, THE PROGRAM SHOULD BE SETUP AGAIN TO SET THE CORRECT VECTOR. IF AN INTERRUPT OCCURRED, THE ADDRESS TO WHICH THE DZV11 INTERRUPTED TO IS PICKED UP AND REPORTED AS THE VECTOR. NOTE: IF THE VECTOR REPORTED IS NOT THE VECTOR SET UP BY YOU, THERE IS A PROBLEM AND AUTO SIZING SHOULD NOT BE DONE.

8.5.3 PARAMETER ASSUMPTIONS.

SINCE TOO MUCH HARDWARE WOULD NEED TO BE TURNED ON TO SIZE THE REST OF THE PARAMETERS; THE PROGRAM MUST ASSUME THE REMAINING VARIATIONS. THE RESULT IF NOT TO YOUR SPECIFIC CONFIGURATION MAY BE ALTERED BY HAND. IN THIS WAY 95% OF THE PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.

THEREFORE:

- 1) ALL FOUR LINES ARE ASSUMED TO BE TESTED.
- 2) DEFAULT BAUD RATE IS SET TO 17 (19.2 KBAUD).
- 3) MODE OF OPERATION IS "INTERNAL MODE".

FOR ALL PARAMETER ADJUSTMENTS PLEASE REFER TO SECTION 8.4 FOR GREATER DETAIL.

9.0 RUNNING THE DZV11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

THE DZV DIAGNOSTICS HAVE BEEN DESIGNED TO BE COMPATIBLE WITH THE APT (AUTOMATED PRODUCT TEST) SYSTEM. THE DZV LOGIC TEST DIAGNOSTICS (CVDZA, AND CVDZB) CAN BE RUN AS STANDALONE DIAGNOSTICS OR IN EITHER OF THE APT MODES. CVDZC, HOWEVER IS DESIGNED AS A STANDALONE DIAGNOSTIC ONLY AND REQUIRES DIRECT OPERATOR PARTICIPATION.

9.1.2 SETTING UP THE DIAGNOSTIC USING APT

THE DIAGNOSTIC USES SEVERAL VARIABLES IN THE REGION SUBTITLED "APT MAILBOX-ETABLE". THESE VARIABLES ARE:

SSWREG -(1142) USED AS THE SOFTWARE SWITCH REGISTER WHILE RUNNING UNDER APT.

SVECT1 -(1170) USED TO SPECIFY THE FIRST VECTOR ADDRESS

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-10
 CVDZAD.P11 11-MAR-83 10:06

6271		
6272		
6273	SBASE	-(1174) USED TO INDICATE BOTTOM ADDRESS OF DZV11 UNDER TEST
6274	SDEVM	-(1176) A BIT MAP REPRESENTING WHICH DZV11'S WILL BE TESTED
6275		
6276	SCDW1	-(1200) USED TO INDICATE WHICH LINES TO RUN ON ALL DZV11'S
6277		
6278	SCDW2	-(1202) USED TO INDICATE THE DEFAULT TEST MODE. SET TO 0 FOR
6279		INTERNAL TESTING, 200 FOR EXTERNAL LOOP BACK (H325
6280		INSTALLED), OR SET TO 100000 FOR STAGGERED LOOP BACK
6281		TESTING (H329 INSTALLED).
6282	SDDW0	-(1204) EACH OF THE SDDW WORDS DESCRIBES THE PARAMETERS
6283		(LPR) FOR A PARTICULAR DZV11, GOING UP TO 16 DZV11'S
6284		

9.1.3 RUNNING UNDER APT

ALL OF THE VARIABLES MENTIONED IN SECTION 9.1.2 SHOULD BE SET UP PRIOR TO RUNNING THE DIAGNOSTIC UNDER APT.

NOTE

BE SURE SBASE POINTS TO THE FIRST DZV11 BEFORE RUNNING

BASED ON THESE VALUES, THE DIAGNOSTIC WILL SET UP THE STATUS TABLE. THE USER IS THEN FREE TO MONITOR UNDER APT AS NORMAL.

10.0 PROGRAM DESCRIPTION

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC PACKAGE (MAINDEC-11-DZQAC-C3).

INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***

MISCELLANEOUS DEFINITIONS

GENERAL PURPOSE REGISTER DEFINITIONS

PRIORITY LEVEL DEFINITIONS

"SWITCH REGISTER" SWITCH DEFINITIONS

DATA BIT DEFINITIONS (BIT00 TO BIT15)

BASIC "CPU" TRAP VECTOR ADDRESSES

BITS 15-11=CPU TYPE
 11/04=01,11/05=02,11/20=03,11/40
 11/70=06,PDQ=07,Q=10

6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-11
 CVDZAD.P11 11-MAR-83 10:06

BIT 10=REAL TIME CLOCK
 BIT 9=FLOATING POINT PROCESSOR
 BIT 8=MEMORY MANAGEMENT

MEM.TYPE BYTE -- (HIGH BYTE)
 900 NSEC CORE=001
 300 NSEC BIPOLAR=002
 500 NSEC MOS=003

MEM.LAST ADDR.=3 BYTES, THIS WORD AND L

THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
 USED IN THE PROGRAM.

THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR
 THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE I
 NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
 NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS

EM	::POINTS TO THE ERROR MESSAGE
DH	::POINTS TO THE DATA HEADER
DT	::POINTS TO THE DATA
DF	::POINTS TO THE DATA FORMAT

INCREMENT THE PASS NUMBER (\$PASS)
 IF THERES A MONITOR GO TO IT
 IF THERE ISN'T JUMP TO CYCLE

THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 AND LOAD THE TEST NUMBER(\$STNM) INTO THE DISPLAY REG.(DISPLAY<7
 AND LOAD THE ERROR FLAG (\$ERFLG) INTO DISPLAY<15:08>
 THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:

SW14=1 LOOP ON TEST
 SW11=1 INHIBIT ITERATIONS

CALL SCOPE ::SCOPE=IOT

ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 B
 THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
 NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
 NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
 NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.

CALL:
 1) USING A TRAP INSTRUCTION
 TYPE ,MESADR ::MESADR IS FIRST ADDRESS OF AN
 OR
 TYPE
 MESADR

ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.

6327
 6328
 6329
 6330
 6331
 6332
 6333
 6334
 6335
 6336
 6337
 6338
 6339
 6340
 6341
 6342
 6343
 6344
 6345
 6346
 6347
 6348
 6349
 6350
 6351
 6352
 6353
 6354
 6355
 6356
 6357
 6358
 6359
 6360
 6361
 6362
 6363
 6364
 6365
 6366
 6367
 6368
 6369
 6370
 6371
 6372
 6373
 6374
 6375
 6376
 6377
 6378
 6379
 6380
 6381
 6382

6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438

IF BIT7 IN THE ENVIRONMENT MODE (\$ENVM) BYTE IS SET,
THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.

ROUTINE USED TO "AUTO SIZE" THE DZV11
CSR AND VECTOR.

NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
ADDRESS RANGE (160000:167770)
AND THE VECTOR MAY BE ANY WHERE IN THE
FLOATING VECTOR RANGE (300:770)

***** TEST 1 *****
THIS TEST PROVES THE BUS REPLY RESPONSE
DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
DZVCSR, DZVRBUF, DZVTOR, DZVMSR

***** TEST 2 *****
THIS TEST PROVES THAT BIT "DCLR"
CAN BE SET AND THAT IT WILL CLEAR
BY ITSELF

***** TEST 3 *****
TEST TO VERIFY THAT THE R/W BITS OF THE
DZVCSR REGISTER CAN BE SET. THEN VERIFY THAT
THESE BITS CAN BE CLEARED. AND FINALLY, VERIFY
THAT AFTER BEING SET AGAIN THEY CAN BE
CLEARED BY A "DEVICE CLEAR".
THE BITS TESTED ARE: MAINT, MSENAB, SILOEN,
RIE, AND TIE.

***** TEST 4 *****
THIS TESTS THAT ALL OF THE TCR BITS
CAN BE: SET, CLEARED, AND CLEARED BY A DEVICE CLEAR.
THIS TEST ALSO DETERMINES IF THE DTR BITS CAN
BE SET, CLEARED, AND CLEARED BY A RESET.

***** TEST 5 *****
THIS TEST VERIFIES THAT
BITS "RDONE, TRDY, BIT9, BIT8,
AND SILOAL" ARE READ ONLY AND THAT TRDY IS
ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.

***** TEST 6 *****
THIS TEST VERIFIES THAT:
TIE, SILOEN, RIE, MSENAB, AND MAINT ARE THE
ONLY R/W BITS IN THE DZVCSR AND THAT
SETTING "DCLR" IN THE CSR WILL CLEAR THESE BITS.

***** TEST 7 *****
THIS TEST PERFORMS RESET TESTING AND
TESTING OF READ ONLY REGISTER DZVRBUF
AND TESTING OF WRITE ONLY REGISTER DZVLPR

6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494

***** TEST 10 *****
THIS TEST PERFORMS RESET TESTING AND
TESTING OF READ ONLY REGISTER DZVMSR
AND TESTING OF WRITE ONLY REGISTER DZVTDR

***** TEST 11 *****
VERIFY THAT SETTING 'DTR' FOR A LINE WILL
BRING UP 'CO' AND 'RING' FOR:
THE SAME LINE IF IN EXTERNAL MODE
THE STAGGERED LINE IF IN STAGGERED MODE.
LINES ARE STAGGERED AS FOLLOWS:
LINE0 WITH LINE1; LINE2 WITH LINE3.
THIS TEST IS ONLY RUN IF AN H325, OR H329
IS CONNECTED ON THE DZV UNDER TEST.

***** TEST 12 *****
THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
IS READY TO BE LOADED, AND THAT THE LINE SPECI-
FIED IN BITS 8-9 OF DZVCSR CORRESPOND
TO THE LINE SELECTED IN DZVTCR

***** TEST 13 *****
TEST TO TRANSMIT ONE CHAR AND
RECEIVE ONE CHAR ON ONE LINE
AT A TIME. THE CHAR IS '252' AND
ALL SELECTED LINES WILL BE TURNED ON .

THIS IS THE FIRST TIME ANY
DATA IS CHECKED IN THE RECEIVER.
USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.

***** TEST 14 *****
THIS TEST VERIFIES THAT EACH RECEIVING LINE CAN BE
DISABLED BY SETTING RCVON (BIT12 IN THE LPR REGISTER)
TO ZERO FOR EACH LINE.
THIS TEST ALSO VERIFIES THAT THE SILO CAN BE
EMPTIED BY ISSUING A DEVICE MASTER CLEAR.

***** TEST 15 *****
THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
CHARACTERS (FLAG MODE) AND THE RECEIVER RECEIVES (FLAG MODE)
(ONE LINE AT A TIME BASED UPON VALID LINES)
THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED

***** TEST 16 *****
THIS TEST WILL PROVE THAT:
1) THE TRANSMITTER 'BREAK BIT' WORKS
2) THE RECEIVER CAN FLAG 'FRAMING ERRORS'
3) THE RECEIVER CAN FLAG 'PARITY ERRORS'
ONLY ONE LINE AT A TIME WILL BE EXERCISED.

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 3-14
CVDZAD.P11 11-MAR-83 10:06

6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511

***** TEST 17 *****
THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
WHILE THE PROCESSOR STATUS DOES NOT ALLOW INTERRUPTS
BUT WILL INTERRUPT IF THE PROCESSOR STATUS
ALLOWS INTERRUPTS.

***** TEST 20 *****
THIS TEST VERIFIES THAT THE RECEIVER WILL
INTERRUPT BEFORE THE TRANSMITTER EVEN
THOUGH THE TRANSMITTER WAS ENABLED
FIRST. SET PS TO HIGH (MASK INTERRUPTS);
GET RDONE AND TRDY TO SET;
SET TX IE AND RX IE;
CLEAR PS AND EXPECT RX TO INTERRUPT FIRST

8

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10
 CVDZAD.P11 11-MAR-83 10:06

```

8730      .TITLE CVDZA-D
(2)      ;*COPYRIGHT (C) 1977,1983
(2)      ;*DIGITAL EQUIPMENT CORP.
(2)      ;*MAYNARD, MASS. 01754
(2)      ;*
(2)      ;*
(2)      ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
(2)      ;*PACKAGE (MAINDEC-11-DZQAC-C8), OCT, 1982.
(2)      ;*
(2)      000001 $TN=1
(2)      ;STARTING PROCEDURE
(2)      ;LOAD PROGRAM
(2)      ;LOAD ADDRESS 000200
(2)      ;PRESS START
(2)      ;PROGRAM WILL TYPE
(2)      ;"CVDZAD/<200>/FOUR LINE ASYNC MUX TESTS, PART 1 OF 2"
(2)      ;PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
(2)      ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
(2)      ;AND THEN RESUME TESTING

(1)      .REM !
(1)      ;SWITCH REGISTER OPTIONS
(1)      ;-----
(1)
(1)      SW15=100000      :=1,HALT ON ERROR
(1)      SW14=40000      :=1,LOOP ON CURRENT TEST
(1)      SW13=20000      :=1,INHIBIT ERROR TIMEOUT
(1)      SW12=10000      :=1,DELETE TIMEOUT/BELL ON ERROR.
(1)      SW11=4000       :=1,INHIBIT ITERATIONS
(1)      SW10=2000       :=1,ESCAPE TO NEXT TEST ON ERROR
(1)      SW09=1000       :=1,LOOP WITH CURRENT DATA
(1)      SW08=400        :=1,LOOP ON ERROR
(1)      SW07=200        :=1, DO "AUTO SIZING" ON INITIAL START UP.
(1)      SW06=100        :=1, DESELECT SPECIFIC DEVICES
(1)      ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
(1)
(1)      SW05=40
(1)      SW04=20          :=1, SELECT DELAY PARAMETER
(1)      SW03=10          :=1, SELECT SPECIFIC PARAMETERS
(1)      SW02=4           :=1, LOCK ON TEST SELECT
(1)      SW01=2           :=1, RESTART PROGRAM AT SELECTED TEST
(1)      SW00=1          :=1, SELECT DEVICE ADDRESS, VECTOR, ETC.
(1)      !
(2)      .SBTTL BASIC DEFINITIONS
(2)
(2)      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***
(2)      001120 STACK= 1120
(2)      104000 ERROR= EMT      ;;BASIC DEFINITION OF ERROR CALL
(2)      000004 SCOPE= IOT      ;;BASIC DEFINITION OF SCOPE CALL
(2)
(2)      ;*MISCELLANEOUS DEFINITIONS
(2)      000011 HT= 11          ;;CODE FOR HORIZONTAL TAB
(2)      000012 LF= 12          ;;CODE FOR LINE FEED
(2)      000015 CR= 15          ;;CODE FOR CARRIAGE RETURN
(2)      000200 CRLF= 200       ;;CODE FOR CARRIAGE RETURN-LINE FEED
(2)      177776 PS= 177776     ;;PROCESSOR STATUS WORD

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-1
 CVDZAD.P11 11-MAR-83 10:06 BASIC DEFINITIONS

(2)	177776	PSW= PS	
(2)	177774	STKLMT= 177774	::STACK LIMIT REGISTER
(2)	177772	PIRQ= 177772	::PROGRAM INTERRUPT REQUEST REGISTER
(2)	177570	DSWR= 177570	::HARDWARE SWITCH REGISTER
(2)	177570	DDISP= 177570	::HARDWARE DISPLAY REGISTER
(2)		:*GENERAL PURPOSE REGISTER DEFINITIONS	
(2)	000000	R0= X0	::GENERAL REGISTER
(2)	000001	R1= X1	::GENERAL REGISTER
(2)	000002	R2= X2	::GENERAL REGISTER
(2)	000003	R3= X3	::GENERAL REGISTER
(2)	000004	R4= X4	::GENERAL REGISTER
(2)	000005	R5= X5	::GENERAL REGISTER
(2)	000006	R6= X6	::GENERAL REGISTER
(2)	000007	R7= X7	::GENERAL REGISTER
(2)	000006	SP= X6	::STACK POINTER
(2)	000007	PC= X7	::PROGRAM COUNTER
(2)		:*PRIORITY LEVEL DEFINITIONS	
(2)	000000	PR0= 0	::PRIORITY LEVEL 0
(2)	000040	PR1= 40	::PRIORITY LEVEL 1
(2)	000100	PR2= 100	::PRIORITY LEVEL 2
(2)	000140	PR3= 140	::PRIORITY LEVEL 3
(2)	000200	PR4= 200	::PRIORITY LEVEL 4
(2)	000240	PR5= 240	::PRIORITY LEVEL 5
(2)	000300	PR6= 300	::PRIORITY LEVEL 6
(2)	000340	PR7= 340	::PRIORITY LEVEL 7
(2)		:*"SWITCH REGISTER" SWITCH DEFINITIONS	
(2)	100000	SW15= 100000	
(2)	040000	SW14= 40000	
(2)	020000	SW13= 20000	
(2)	010000	SW12= 10000	
(2)	004000	SW11= 4000	
(2)	002000	SW10= 2000	
(2)	001000	SW09= 1000	
(2)	000400	SW08= 400	
(2)	000200	SW07= 200	
(2)	000100	SW06= 100	
(2)	000040	SW05= 40	
(2)	000020	SW04= 20	
(2)	000010	SW03= 10	
(2)	000004	SW02= 4	
(2)	000002	SW01= 2	
(2)	000001	SW00= 1	
(2)	001000	SW9= SW09	
(2)	000400	SW8= SW08	
(2)	000200	SW7= SW07	
(2)	000100	SW6= SW06	
(2)	000040	SW5= SW05	
(2)	000020	SW4= SW04	
(2)	000010	SW3= SW03	
(2)	000004	SW2= SW02	
(2)	000002	SW1= SW01	
(2)	000001	SW0= SW00	

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-2
 CVDZAD.P11 11-MAR-83 10:06 BASIC DEFINITIONS

```

(2)          100000      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(2)          040000      BIT15= 100000
(2)          020000      BIT14= 40000
(2)          010000      BIT13= 20000
(2)          004000      BIT12= 10000
(2)          002000      BIT11= 4000
(2)          001000      BIT10= 2000
(2)          000400      BIT09= 1000
(2)          000200      BIT08= 400
(2)          000100      BIT07= 200
(2)          000040      BIT06= 100
(2)          000020      BIT05= 40
(2)          000010      BIT04= 20
(2)          000004      BIT03= 10
(2)          000002      BIT02= 4
(2)          000001      BIT01= 2
(2)          001000      BIT00= 1
(2)          000400      BIT9=  BIT09
(2)          000200      BIT8=  BIT08
(2)          000100      BIT7=  BIT07
(2)          000040      BIT6=  BIT06
(2)          000020      BIT5=  BIT05
(2)          000010      BIT4=  BIT04
(2)          000004      BIT3=  BIT03
(2)          000002      BIT2=  BIT02
(2)          000001      BIT1=  BIT01
(2)          000001      BIT0=  BIT00

(2)          000004      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(2)          000010      ERRVEC= 4      ;;TIME OUT AND OTHER ERRORS
(2)          000014      RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
(2)          000014      TBITVEC=14    ;; "T" BIT
(2)          000014      TRTVEC= 14     ;;TRACE TRAP
(2)          000014      BPTVEC= 14    ;;BREAKPOINT TRAP (BPT)
(2)          000020      IOTVEC= 20    ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(2)          000024      PWRVEC= 24    ;;POWER FAIL
(2)          000030      EMTVEC= 30    ;;EMULATOR TRAP (EMT) **ERROR**
(2)          000034      TRAPVEC=34   ;; "TRAP" TRAP
(2)          000060      TKVEC= 60     ;;TTY KEYBOARD VECTOR
(2)          000064      TPVEC= 64    ;;TTY PRINTER VECTOR
(2)          000240      PIRQVEC=240   ;;PROGRAM INTERRUPT REQUEST VECTOR

(1)          (1)          ;INSTRUCTION DEFINITIONS
(1)          (1)          ;-----
(1)          (1)          PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
(1)          (1)          POP1SP=5726  ;INCREMENT PROCESSOR STACK 1 WORD
(1)          (1)          PUSHRO=10046 ;SAVE R0 ON STACK
(1)          (1)          POPRO=12600  ;RESTORE R0 FROM STACK
(1)          (1)          PUSH2SP=24646 ;DECREMENT STACK TWICE
(1)          (1)          POP2SP=22626  ;INCREMENT STACK TWICE
(1)          (1)          MASK=BIT7    ;SET INTERRUPT MASK (INHIBIT FURTHER INTERRUPTS)
(1)          (1)          CLEAR=0     ;ALLOW INTERRUPTS (CLEAR PROCESSOR STATUS)
(1)          (1)

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13
 CVDZAD.P11 11-MAR-83 10:06

PAGE 10-3
 GENERAL DEFINITIONS AND EQUIVALENCES

```

(1)                                     :DZV11 CONTROL AND STATUS REGISTER DEFINITIONS
(1)                                     :(DZVCSR)      BIT DEFINITIONS
(1)                                     :-----
(1) 000010 MAINT = BIT3      :MAINTENANCE MODE ENABLE
(1) 000020 DCLR=BIT4       :DEVICE CLEAR
(1) 000040 MSENAB=BIT5     :MASTER SCAN ENABLE
(1) 000100 RIE=BIT6        :RECEIVER INTERRUPT ENABLE
(1) 000200 RDONE=BIT7      :RECEIVER DONE
(1) 010000 SILOEN= BIT12   :SILO ALARM ENABLE
(1) 020000 SILOAL = BIT13  :SILO ALARM
(1) 040000 TIE=BIT14       :TRANSMITTER INTERRUPT ENABLE
(1) 100000 TRDY=BIT15      :TRANSMITTER READY

(1)                                     :DZVCSR WORD DEFINITIONS
(1)                                     :-----
(1) 000000 TL0=0           :TRANSMIT LINE 0
(1) 000400 TL1=BIT8        :TRANSMIT LINE 1
(1) 001000 TL2=BIT9        :TRANSMIT LINE 2
(1) 001400 TL3=BIT9!BIT8  :TRANSMIT LINE 3

(1)                                     :DZVRBUF BIT DEFINITIONS
(1)                                     :-----
(1) 010000 PARER=BIT12     :PARITY ERROR
(1) 020000 FRMERR=BIT13    :FRAME ERROR
(1) 040000 OVRRUN=BIT14    :OVERRUN ERROR
(1) 100000 DVALID=BIT15    :DATA VALID

(1)                                     :DZVRBUF WORD DEFINITIONS
(1)                                     :-----
(1) 000000 RL0=0           :RECEIVER LINE 0
(1) 000400 RL1=BIT8        :RECEIVER LINE 1
(1) 001000 RL2=BIT9        :RECEIVER LINE 2
(1) 001400 RL3=BIT9!BIT8  :RECEIVER LINE 3

(1)                                     :DZVLPR WORD DEFINITIONS
(1)                                     :-----
(1) 000000 LP0=0           :LINE PARAMETER 0
(1) 000001 LP1=BIT0        :LINE PARAMETER 1
(1) 000002 LP2=BIT1        :LINE PARAMETER 2
(1) 000003 LP3=BIT1!BIT0   :LINE PARAMETER 3

(1) 000000 FIVE=0         :FIVE BITS/CHAR,1 STOP BIT
(1) 000010 SIX=BIT3        :SIX BITS/CHAR,1 STOP BIT
(1) 000020 SEVEN=BIT4      :SEVEN BITS/CHAR,1 STOP BIT
(1) 000030 EIGHT=BIT4!BIT3 :EIGHT BITS/CHAR,1 STOP BIT
(1) 000040 FIVES=BIT5      :FIVE BITS/CHAR,2 STOP BITS
(1) 000050 SIXS=BIT5!BIT3  :SIX BITS/CHAR,2 STOP BITS
(1) 000060 SEVENS=BIT5!BIT4 :SEVEN BITS/CHAR, 2 STOP BITS
(1) 000070 EIGHTS=BIT5!BIT4!BIT3 :EIGHT BITS/CHAR, 2 STOP BITS

(1) 000100 PARITY=BIT6     :PARITY ENABLED
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-4
 CVDZAD.P11 11-MAR-83 10:06 GENERAL DEFINITIONS AND EQUIVALENCES

(1)	000200	ODDPAR=BIT7	:ODD PARITY ENABLED
(1)	000000	ONESTOP=0	:ONE STOP BIT ENABLED
(1)	000040	TWOSTOP=BIT5	:TWO STOP BITS ENABLED
(1)	000000	EVEPAR=0	:EVEN PARITY ENABLED
(1)	010000	RCVON=BIT12	:ENABLE RECEIVER (RECEIVER ON)
(1)	000000	S50=0	:SPEED 50 BAUD
(1)	000400	S75=BIT8	:SPEED 75 BAUD
(1)	001000	S110=BIT9	:SPEED 110 BAUD
(1)	001400	S134=BIT9!BIT8	:SPEED 134.5 BAUD
(1)	002000	S150=BIT10	:SPEED 150 BAUD
(1)	002400	S300=BIT10!BIT8	:SPEED 300 BAUD
(1)	003000	S600=BIT10!BIT9	:SPEED 600 BAUD
(1)	003400	S1200=BIT10!BIT9!BIT8	:SPEED 1200 BAUD
(1)	004000	S1800=BIT11	:SPEED 1800 BAUD
(1)	004400	S2000=BIT11!BIT8	:SPEED 2000 BAUD
(1)	005000	S2400=BIT11!BIT9	:SPEED 2400 BAUD
(1)	005400	S3600=BIT11!BIT9!BIT8	:SPEED 3600 BAUD
(1)	006000	S4800=BIT11!BIT10	:SPEED 4800 BAUD
(1)	006400	S7200=BIT11!BIT10!BIT8	:SPEED 7200 BAUD
(1)	007000	S9600=BIT11!BIT10!BIT9	:SPEED 9600 BAUD
(1)	007400	S19200=BIT11!BIT10!BIT9!BIT8	:SPEED 19200 BAUD
(1)			
(1)		:DZVTCR BIT DEFINITIONS	
(1)		-----	
(1)	000001	TCR0=BIT0	:ENABLE TRANSMISSION ON LINE 0
(1)	000002	TCR1=BIT1	:ENABLE TRANSMISSION ON LINE 1
(1)	000004	TCR2=BIT2	:ENABLE TRANSMISSION ON LINE 2
(1)	000010	TCR3=BIT3	:ENABLE TRANSMISSION ON LINE 3
(1)	000400	DTR0=BIT8	:DATA TERMINAL READY FOR LINE 0
(1)	001000	DTR1=BIT9	:DATA TERMINAL READY FOR LINE 1
(1)	002000	DTR2=BIT10	:DATA TERMINAL READY FOR LINE 2
(1)	004000	DTR3=BIT11	:DATA TERMINAL READY FOR LINE 3
(1)			
(1)		:DZVMSR BIT DEFINITIONS	
(1)		-----	
(1)	000001	RING0=BIT0	:RING INDICATED ON LINE 0
(1)	000002	RING1=BIT1	:RING INDICATED ON LINE 1
(1)	000004	RING2=BIT2	:RING INDICATED ON LINE 2
(1)	000010	RING3=BIT3	:RING INDICATED ON LINE 3
(1)	000400	CO0=BIT8	:CARRIER PRESENT ON LINE 0
(1)	001000	CO1=BIT9	:CARRIER PRESENT ON LINE 1
(1)	002000	CO2=BIT10	:CARRIER PRESENT ON LINE 2
(1)	004000	CO3=BIT11	:CARRIER PRESENT ON LINE 3
(1)			
(1)		:DZVTDR BIT DEFINITIONS	
(1)		-----	
(1)	000400	BRK0=BIT8	:BREAK FOR LINE 0
(1)	001000	BRK1=BIT9	:BREAK FOR LINE 1
(1)	002000	BRK2=BIT10	:BREAK FOR LINE 2
(1)	004000	BRK3=BIT11	:BREAK FOR LINE 3

- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)
- (1)

:TABLE OF LOOP AROUND FUNCTIONS (H325)

I	^
V	^
REC	TRANS
DATA	DATA

I	^
V	^
CO	RTS

I	^
V	^
RING	DTR

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-6
CVDZAD.P11 11-MAR-83 10:06

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

(1) ;:*****
(1) ;-----
(1) ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
(1) ;THE STANDARD 'TRAP CATCHER' IS PLACED
(1) ;BETWEEN ADDRESS 0 TO ADDRESS 776.
(1) ;IT LOOKS LIKE 'PC+2 KALT'.
(1) ;-----
(1) ;:*****
(1)
(1) 000000 . =0
(1) ; UFD DON'T SETUP EMT AS TRAP CATCHER SINCE IT IS USED FOR LINKAGE BETWEEN UFD MONITOR
(1) 000034 . =34
(1) ;STANDARD INTERRUPT VECTORS
(1) ;-----
(1)
(1) . =20
(1) 000020 000020 .SCOPE ;SCOPE LOOP HANDLER
(1) 000022 004462 MASK ;HANDLE AT PRIORITY 7
(1) 000024 007564 $PWDRN ;POWER FAIL HANDLER
(1) 000026 000340 340 ;SERVICE AT PRIORITY LEVEL 7
(1) . =34
(1) 000034 006402 .TRPSRV ;GENERAL HANDLER DISPATCH SERVICE
(1) 000036 000340 340 ;SERVICE AT PRIORITY LEVEL 7
(2) .SBTTL ACT11 HOOKS
(2) ;:*****
(2) ;HOOKS REQUIRED BY ACT11
(2) 000040 $SVPC=. ;SAVE PC
(2) 000046 . =46
(2) 000046 004416 SENDAD ;:1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
(2) 000052 000052 . =52
(2) 000052 000000 .WORD 0 ;:2)SET LOC.52 TO ZERO
(2) 000040 000040 .=$SVPC ;: RESTORE PC
(1)
(1) . =174
(1) 000174 000000 DISPREG:0 ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
(1) 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER FOR SWITCHLESS 11S
(1) 000200 000200 . =200
(1) 000200 000137 002116 JMP .START ;GO TO START OF PROGRAM
(1)
(2) . =1000
(2) 001000 005200 053103 055104 MTITLE: .ASCIZ <200><12>/CVDZAD/<200>/FOUR LINE ASYNC MUX TESTS, PART 1 OF 2/<200>
(2)

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-7
CVDZAD.P11 11-MAR-83 10:06 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

(3)          001120          . =1120
(4)          ::*****
(4)          .SBTTL  APT MAILBOX-ETABLE
(4)          ::*****
(4)          .EVEN
(4) 001120  SMAIL:          ::APT MAILBOX
(4) 001120 000000  SMSGTY: .WORD  AMSGTY  ::MESSAGE TYPE CODE
(4) 001122 000000  SFATAL: .WORD  AFATAL  ::FATAL ERROR NUMBER
(4) 001124 000000  STESTN: .WORD  ATESTN  ::TEST NUMBER
(4) 001126 000000  SPASS: .WORD  APASS  ::PASS COUNT
(4) 001130 000000  SDEVCT: .WORD  ADEVCT  ::DEVICE COUNT
(4) 001132 000000  SUNIT: .WORD  AUNIT  ::I/O UNIT NUMBER
(4) 001134 000000  SMSGAD: .WORD  AMSGAD  ::MESSAGE ADDRESS
(4) 001136 000000  SMSGLG: .WORD  AMSGLG  ::MESSAGE LENGTH
(4) 001140  SETABLE:      ::APT ENVIRONMENT TABLE
(4) 001140      000  SENV: .BYTE  AENV  ::ENVIRONMENT BYTE
(4) 001141      000  SENVM: .BYTE  AENVM  ::ENVIRONMENT MODE BITS
(4) 001142 000000  SSWREG: .WORD  ASWREG  ::APT SWITCH REGISTER
(4) 001144 000000  SUSWR: .WORD  AUSWR  ::USER SWITCHES
(4) 001146 000000  SCPUOP: .WORD  ACPUOP  ::CPU TYPE, OPTIONS
(4)          :*          BITS 15-11=CPU TYPE
(4)          :*          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(4)          :*          11/70=06,PDQ=07,Q=10
(4)          :*          BIT 10=REAL TIME CLOCK
(4)          :*          BIT 9=FLOATING POINT PROCESSOR
(4)          :*          BIT 8=MEMORY MANAGEMENT
(4) 001150      000  SMAMS1: .BYTE  AMAMS1  ::HIGH ADDRESS,M.S. BYTE
(4) 001151      000  SMTYP1: .BYTE  AMTYP1  ::MEM. TYPE,BLK#1
(4)          :*          MEM.TYPE BYTE  -- (HIGH BYTE)
(4)          :*          900 NSEC CORE=001
(4)          :*          300 NSEC BIPOLAR=002
(4)          :*          500 NSEC MOS=003
(4) 001152 000000  SMADR1: .WORD  AMADR1  ::HIGH ADDRESS,BLK#1
(4)          :*          MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(4) 001154      000  SMAMS2: .BYTE  AMAMS2  ::HIGH ADDRESS,M.S. BYTE
(4) 001155      000  SMTYP2: .BYTE  AMTYP2  ::MEM.TYPE,BLK#2
(4) 001156 000000  SMADR2: .WORD  AMADR2  ::MEM.LAST ADDRESS,BLK#2
(4) 001160      000  SMAMS3: .BYTE  AMAMS3  ::HIGH ADDRESS,M.S.BYTE
(4) 001161      000  SMTYP3: .BYTE  AMTYP3  ::MEM.TYPE,BLK#3
(4) 001162 000000  SMADR3: .WORD  AMADR3  ::MEM.LAST ADDRESS,BLK#3
(4) 001164      000  SMAMS4: .BYTE  AMAMS4  ::HIGH ADDRESS,M.S.BYTE
(4) 001165      000  SMTYP4: .BYTE  AMTYP4  ::MEM.TYPE,BLK#4
(4) 001166 000000  SMADR4: .WORD  AMADR4  ::MEM.LAST ADDRESS,BLK#4
(4) 001170 000300  SVECT1: .WORD  AVECT1  ::INTERRUPT VECTOR#1,BUS PRIORITY#1
(4) 001172 000000  SVECT2: .WORD  AVECT2  ::INTERRUPT VECTOR#2BUS PRIORITY#2
(4) 001174 160010  SBASE: .WORD  ABASE  ::BASE ADDRESS OF EQUIPMENT UNDER TEST
(4) 001176 000001  SDEVM: .WORD  ADEVM  ::DEVICE MAP
(4) 001200 000017  SCDW1: .WORD  ACDW1  ::CONTROLLER DESCRIPTION WORD#1
(4) 001202 000000  SCDW2: .WORD  ACDW2  ::CONTROLLER DESCRIPTION WORD#2
(4) 001204 017470  SDDW0: .WORD  ADDW0  ::DEVICE DESCRIPTOR WORD#0
(4) 001206 017470  SDDW1: .WORD  ADDW1  ::DEVICE DESCRIPTOR WORD#1
(4) 001210 017470  SDDW2: .WORD  ADDW2  ::DEVICE DESCRIPTOR WORD#2
(4) 001212 017470  SDDW3: .WORD  ADDW3  ::DEVICE DESCRIPTOR WORD#3
(4) 001214 017470  SDDW4: .WORD  ADDW4  ::DEVICE DESCRIPTOR WORD#4
(4) 001216 017470  SDDW5: .WORD  ADDW5  ::DEVICE DESCRIPTOR WORD#5

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-8
CVDZAD.P11 11-MAR-83 10:06 APT MAILBOX-ETABLE

(4)	001220	017470	SDDW6:	.WORD	ADDW6	::DEVICE	DESCRIPTOR	WORD#6
(4)	001222	017470	SDDW7:	.WORD	ADDW7	::DEVICE	DESCRIPTOR	WORD#7
(4)	001224	017470	SDDW8:	.WORD	ADDW8	::DEVICE	DESCRIPTOR	WORD#8
(4)	001226	017470	SDDW9:	.WORD	ADDW9	::DEVICE	DESCRIPTOR	WORD#9
(4)	001230	017470	SDDW10:	.WORD	ADDW10	::DEVICE	DESCRIPTOR	WORD#10
(4)	001232	017470	SDDW11:	.WORD	ADDW11	::DEVICE	DESCRIPTOR	WORD#11
(4)	001234	017470	SDDW12:	.WORD	ADDW12	::DEVICE	DESCRIPTOR	WORD#12
(4)	001236	017470	SDDW13:	.WORD	ADDW13	::DEVICE	DESCRIPTOR	WORD#13
(4)	001240	017470	SDDW14:	.WORD	ADDW14	::DEVICE	DESCRIPTOR	WORD#14
(4)	001242	017470	SDDW15:	.WORD	ADDW15	::DEVICE	DESCRIPTOR	WORD#15
(4)								
(4)								
(4)	001244		SETEND:					
(4)								

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-9
 CVDZAD.P11 11-MAR-83 10:06 COMMON TAGS

```

(3)          .SBTTL COMMON TAGS
(3)
(4)          ::*****
(3)          ::THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(3)          ::USED IN THE PROGRAM.
(3)
(3) 001244    SCMTAG:          ::START OF COMMON TAGS
(3) 001244    000000          .WORD 0
(3) 001246    000          $TSTNM: .BYTE 0          ::CONTAINS THE TEST NUMBER
(3) 001247    000          $ERFLG: .BYTE 0          ::CONTAINS ERROR FLAG
(3) 001250    000000        $ICNT: .WORD 0          ::CONTAINS SUBTEST ITERATION COUNT
(3) 001252    000000        $LPADR: .WORD 0          ::CONTAINS SCOPE LOOP ADDRESS
(3) 001254    000000        $LPERR: .WORD 0          ::CONTAINS SCOPE RETURN FOR ERRORS
(3) 001256    000000        $ERTTL: .WORD 0          ::CONTAINS TOTAL ERRORS DETECTED
(3) 001260    000          $ITEMB: .BYTE 0          ::CONTAINS ITEM CONTROL BYTE
(3) 001261    001          $ERMAX: .BYTE 1          ::CONTAINS MAX. ERRORS PER TEST
(3) 001262    000000        $ERRPC: .WORD 0          ::CONTAINS PC OF LAST ERROR INSTRUCTION
(3) 001264    000000        $GDADR: .WORD 0          ::CONTAINS ADDRESS OF 'GOOD' DATA
(3) 001266    000000        $BDADR: .WORD 0          ::CONTAINS ADDRESS OF 'BAD' DATA
(3) 001270    000000        $GDDAT: .WORD 0          ::CONTAINS 'GOOD' DATA
(3) 001272    000000        $BDDAT: .WORD 0          ::CONTAINS 'BAD' DATA
(3) 001274    000000        .WORD 0          ::RESERVED--NOT TO BE USED
(3) 001276    000000        .WORD 0
(3) 001300    000          $AUTOB: .BYTE 0          ::AUTOMATIC MODE INDICATOR
(3) 001301    000          $INTAG: .BYTE 0          ::INTERRUPT MODE INDICATOR
(3) 001302    000000        .WORD 0
(3) 001304    177570        $SWR: .WORD DSWR          ::ADDRESS OF SWITCH REGISTER
(3) 001306    177570        $DISPLAY: .WORD DDISP          ::ADDRESS OF DISPLAY REGISTER
(3) 001310    177560        $TKS: 177560          ::TTY KBD STATUS
(3) 001312    177562        $TKB: 177562          ::TTY KBD BUFFER
(3) 001314    177564        $TPS: 177564          ::TTY PRINTER STATUS REG. ADDRESS
(3) 001316    177566        $TPB: 177566          ::TTY PRINTER BUFFER REG. ADDRESS
(3) 001320    000          $NULL: .BYTE 0          ::CONTAINS NULL CHARACTER FOR FILLS
(3) 001321    002          $FILLS: .BYTE 2          ::CONTAINS # OF FILLER CHARACTERS REQUIRED
(3) 001322    012          $FILLC: .BYTE 12          ::INSERT FILL CHARS. AFTER A 'LINE FEED'
(3) 001323    000          $TPFLG: .BYTE 0          ::"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
(3) 001324    000000        $REGAD: .WORD 0          ::CONTAINS THE ADDRESS FROM
(3)          ::WHICH ($REGO) WAS OBTAINED
(5) 001326    000000        $REG0: .WORD 0          ::CONTAINS (($REGAD)+0)
(5) 001330    000000        $REG1: .WORD 0          ::CONTAINS (($REGAD)+2)
(5) 001332    000000        $REG2: .WORD 0          ::CONTAINS (($REGAD)+4)
(5) 001334    000000        $REG3: .WORD 0          ::CONTAINS (($REGAD)+6)
(5) 001336    000000        $REG4: .WORD 0          ::CONTAINS (($REGAD)+10)
(5) 001340    000000        $REG5: .WORD 0          ::CONTAINS (($REGAD)+12)
(5) 001342    000000        $TMP0: .WORD 0          ::USER DEFINED
(5) 001344    000000        $TMP1: .WORD 0          ::USER DEFINED
(5) 001346    000000        $TMP2: .WORD 0          ::USER DEFINED
(5) 001350    000000        $TMP3: .WORD 0          ::USER DEFINED
(5) 001352    000000        $TMP4: .WORD 0          ::USER DEFINED
(3) 001354    000000        $TIMES: 0          ::MAX. NUMBER OF ITERATIONS
(3) 001356    077          $QUES: .ASCII /?/          ::QUESTION MARK
(3) 001357    015          $CRLF: .ASCII <15>          ::CARRIAGE RETURN
(3) 001360    000012        $LF: .ASCIIZ <12>          ::LINE FEED
    
```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-11
CVDZAD.P11 11-MAR-83 10:06 ERROR POINTER TABLE

```

(2)
(2)                ;PROGRAM CONTROL FLAGS
(2)                ;-----
(2) 001422         000      INIFLG: .BYTE 0          ;PROGRAM INITIALIZATION FLAG
(2) 001423         000      HDRFLG: .BYTE 0          ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
(2) 001424         000      MNTFLG: .BYTE 0          ;MAINTENANCE BIT SET FLAG
(2) 001425         000      DONFLG: .BYTE 0          ;TRANSMISSION COMPLETION FLAG
(2)                .EVEN
(2)                ;DATA VARIABLES
(2) 001426         000000   TD0:  .WORD 0
(2) 001430         000000   TD1:  .WORD 0
(2) 001432         000000   TD2:  .WORD 0
(2) 001434         000000   TD3:  .WORD 0
(2) 001436         000000   TR0:  .WORD 0
(2) 001440         000000   TR1:  .WORD 0
(2) 001442         000000   TR2:  .WORD 0
(2) 001444         000000   TR3:  .WORD 0
(2) 001446
(2)                STOP:
(2)                .SBTTL  APT PARAMETER BLOCK
(3)                ;*****
(2)                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(3)                ;*****
(2)                .SX=.      ;;SAVE CURRENT LOCATION
(2)                .=24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
(2) 000024         000200   200    ;;FOR APT START UP
(2)                .=44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
(2) 000044         001446   $APTHDR ;;POINT TO APT HEADER BLOCK
(2)                .=$X      ;;RESET LOCATION COUNTER
(3)                ;*****
(2)                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(2)                ;INTERFACE SPEC.
(2)
(2) 001446         $APTHD:
(2) 001446         000000   $HIBTS: .WORD 0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(2) 001450         001120   $MADR: .WORD $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
(2) 001452         000120   $TSTM: .WORD 80.        ;;RUN TIM OF LONGEST TEST
(2) 001454         000024   $PASTM: .WORD 20.        ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(2) 001456         000000   $UNITM: .WORD 0.         ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(2) 001460         000052   .WORD SETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
(1)                ;DZV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
(1)                ;-----
(1)                .=1500
(1) 001500         DZV.MAP:
(3)
(3) 001500         000001   DZCRO: .BLKW 1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 0
(3) 001502         000001   DZVCO: .BLKW 1          ;RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 0
(3) 001504         000001   LINE0: .BLKW 1          ;ALL LINES SELECTED
(3) 001506         000001   PAR0:  .BLKW 1          ;PARAMETERS
(3) 001510         000001   MANT0: .BLKW 1          ;MAINTENANCE MODE FOR THIS DEVICE
(3)
(3) 001512         000001   DZCR1: .BLKW 1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 1
(3) 001514         000001   DZVC1: .BLKW 1          ;RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 1
(3) 001516         000001   LINE1: .BLKW 1          ;ALL LINES SELECTED

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-12
 CVDZAD.P11 11-MAR-83 10:06 APT PARAMETER BLOCK

(3)	001520	000001	PAR1:	.BLKW	1	:PARAMETERS
(3)	001522	000001	MANT1:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001524	000001	DZCR2:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 2
(3)	001526	000001	DZVC2:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 2
(3)	001530	000001	LINE2:	.BLKW	1	:ALL LINES SELECTED
(3)	001532	000001	PAR2:	.BLKW	1	:PARAMETERS
(3)	001534	000001	MANT2:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001536	000001	DZCR3:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 3
(3)	001540	000001	DZVC3:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 3
(3)	001542	000001	LINE3:	.BLKW	1	:ALL LINES SELECTED
(3)	001544	000001	PAR3:	.BLKW	1	:PARAMETERS
(3)	001546	000001	MANT3:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001550	000001	DZCR4:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 4
(3)	001552	000001	DZVC4:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 4
(3)	001554	000001	LINE4:	.BLKW	1	:ALL LINES SELECTED
(3)	001556	000001	PAR4:	.BLKW	1	:PARAMETERS
(3)	001560	000001	MANT4:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001562	000001	DZCR5:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 5
(3)	001564	000001	DZVC5:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 5
(3)	001566	000001	LINE5:	.BLKW	1	:ALL LINES SELECTED
(3)	001570	000001	PAR5:	.BLKW	1	:PARAMETERS
(3)	001572	000001	MANT5:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001574	000001	DZCR6:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 6
(3)	001576	000001	DZVC6:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 6
(3)	001600	000001	LINE6:	.BLKW	1	:ALL LINES SELECTED
(3)	001602	000001	PAR6:	.BLKW	1	:PARAMETERS
(3)	001604	000001	MANT6:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001606	000001	DZCR7:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 7
(3)	001610	000001	DZVC7:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 7
(3)	001612	000001	LINE7:	.BLKW	1	:ALL LINES SELECTED
(3)	001614	000001	PAR7:	.BLKW	1	:PARAMETERS
(3)	001616	000001	MANT7:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001620	000001	DZCR10:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 10
(3)	001622	000001	DZVC10:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 10
(3)	001624	000001	LINE10:	.BLKW	1	:ALL LINES SELECTED
(3)	001626	000001	PAR10:	.BLKW	1	:PARAMETERS
(3)	001630	000001	MANT10:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001632	000001	DZCR11:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 11
(3)	001634	000001	DZVC11:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 11
(3)	001636	000001	LINE11:	.BLKW	1	:ALL LINES SELECTED
(3)	001640	000001	PAR11:	.BLKW	1	:PARAMETERS
(3)	001642	000001	MANT11:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
(3)	001644	000001	DZCR12:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZV11 NUMBER 12
(3)	001646	000001	DZVC12:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZV11 NUMBER 12
(3)	001650	000001	LINE12:	.BLKW	1	:ALL LINES SELECTED
(3)	001652	000001	PAR12:	.BLKW	1	:PARAMETERS
(3)	001654	000001	MANT12:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-13
 CVDZAD.P11 11-MAR-83 10:06 APT PARAMETER BLOCK

(3)					
(3)	001656	000001	DZCR13:	.BLKW	1
(3)	001660	000001	DZVC13:	.BLKW	1
(3)	001662	000001	LINE13:	.BLKW	1
(3)	001664	000001	PAR13:	.BLKW	1
(3)	001666	000001	MANT13:	.BLKW	1
(3)					
(3)	001670	000001	DZCR14:	.BLKW	1
(3)	001672	000001	DZVC14:	.BLKW	1
(3)	001674	000001	LINE14:	.BLKW	1
(3)	001676	000001	PAR14:	.BLKW	1
(3)	001700	000001	MANT14:	.BLKW	1
(3)					
(3)	001702	000001	DZCR15:	.BLKW	1
(3)	001704	000001	DZVC15:	.BLKW	1
(3)	001706	000001	LINE15:	.BLKW	1
(3)	001710	000001	PAR15:	.BLKW	1
(3)	001712	000001	MANT15:	.BLKW	1
(3)					
(3)	001714	000001	DZCR16:	.BLKW	1
(3)	001716	000001	DZVC16:	.BLKW	1
(3)	001720	000001	LINE16:	.BLKW	1
(3)	001722	000001	PAR16:	.BLKW	1
(3)	001724	000001	MANT16:	.BLKW	1
(3)					
(3)	001726	000001	DZCR17:	.BLKW	1
(3)	001730	000001	DZVC17:	.BLKW	1
(3)	001732	000001	LINE17:	.BLKW	1
(3)	001734	000001	PAR17:	.BLKW	1
(3)	001736	000001	MANT17:	.BLKW	1
(1)					
(1)	001740	177777	DZV.END:		177777

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-14
CVDZAD.P11 11-MAR-83 10:06 APT PARAMETER BLOCK

```

(1)                                     ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
(1)                                     ;POINTERS TO SUBROUTINES CAN BE FOUND
(1)                                     ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS
(1)                                     ;*****
(1)                                     ;-----
(1) 001742                               .TRPTAB:
(3) 001742 104400 ADVANCE=TRAP+0           ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
(2) 001742 006476 .ADVANCE
(3) 001742 104401 SCOP1=TRAP+1           ;CALL TO LOOP ON CURRENT DATA HANDLER
(2) 001744 004726 .SCOP1
(3) 001744 104402 TYPE=TRAP+2           ;CALL TO TELETYPE OUTPUT ROUTINE
(2) 001746 004752 .TYPE
(3) 001746 104403 INSTR=TRAP+3          ;CALL TO ASCII STRING INPUT ROUTINE
(2) 001750 005602 .INSTR
(3) 001750 104404 INSTER=TRAP+4         ;CALL TO INPUT ERROR HANDLER
(2) 001752 005706 .INSTER
(3) 001752 104405 PARAM=TRAP+5         ;CALL TO NUMERICAL DATA INPUT ROUTINE
(2) 001754 005726 .PARAM
(3) 001754 104406 SETFLG=TRAP+6        ;CALL TO SET FLAG ROUTINE
(2) 001756 010422 .SETFLG
(3) 001756 104407 SAV05=TRAP+7        ;CALL TO REGISTER SAVE ROUTINE
(2) 001760 006126 .SAV05
(3) 001760 104410 RES05=TRAP+10       ;CALL TO REGISTER RESTORE ROUTINE
(2) 001762 006166 .RES05
(3) 001762 104411 CONVRT=TRAP+11      ;CALL TO DATA OUTPUT ROUTINE
(2) 001764 006220 .CONVRT
(3) 001764 104412 CNVRT=TRAP+12      ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
(2) 001766 006224 .CNVRT
(3) 001766 104413 DEVICE.CLR=TRAP+13   ;CALL TO ISSUE A DEVICE CLEAR
(2) 001770 006424 .DEVICE.CLR
(3) 001770 104414 DELAY=TRAP+14       ;CALL TO DELAY FOR FAST CPU'S
(2) 001772 006456 .DELAY
(3) 001772 104415 PARMD=TRAP+15      ;CONVERT DECIMAL STRING TO OCTAL
(2) 001774 011432 .PARMD
(3) 001774 104416 PAWCH=TRAP+16      ;SET FLAG ECHO OR CABLE
(2) 001776 010542 .PAWCH
(3) 001776 104417 DCLASM=TRAP+17     ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
(2) 002000 006444 .DCLASM
(3) 002000 104420 SHIFT=TRAP+20      ;CALL TO ROTATE LINE POINTER
(2) 002002 006510 .SHIFT
(3) 002002 104421 LPRSET=TRAP+21     ;CALL TO SET UP LPR DEVICE REGISTER
(2) 002004 006526 .LPRSET
(3) 002004 104422 BUFSET=TRAP+22    ;CALL TO ZERO BUFFER AREA
(2) 002006 006566 .BUFSET
(1)                                     ;-----
(1)                                     ;*****

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-15
 CVDZAD.P11 11-MAR-83 10:06 APT PARAMETER BLOCK

```

(1)                                     :DZV11 VECTOR AND REGISTER INDIRECT POINTERS
(1)                                     :WORKING AREA
(1)
(1) 002010 160040 DZVCSR: 160040 :R/W
(1) 002012 160041 HDZVCSR:160041 :R/W
(1) 002014 160042 DZVRBUF:160042 :READ ONLY
(1) 002016 160043 HDZVRBUF:160043 :READ ONLY
(1) 002020 160042 DZVLPR: 160042 :WRITE ONLY
(1) 002022 160043 HDZVLPR:160043 :WRITE ONLY
(1) 002024 160044 DZVTCR: 160044 :R/W
(1) 002026 160045 HDZVTCR:160045 :R/W
(1) 002030 160046 DZVMSR: 160046 :READ ONLY
(1) 002032 160047 HDZVMSR:160047 :READ ONLY
(1) 002034 160046 DZVTDR: 160046 :WRITE ONLY
(1) 002036 160047 HDZVTDR:160047 :WRITE ONLY
(1)
(1)                                     :DEFAULT DZV VECTORS
(1)
(1) 002040 000300 DZVRIV: 300 :REC INTR VECTOR
(1) 002042 000302 DZVRIS: 302 :REC INTR STATUS
(1) 002044 000304 DZVTIV: 304 :XMIT INTR VECTOR
(1) 002046 000306 DZVTIS: 306 :XMIT INTR STATUS
(1)
(1)

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-16
CVDZAD.P11 11-MAR-83 10:06 APT PARAMETER BLOCK

```
(1)
(1)
(1)
(1)
(1) 002050
(1) 002050 000000
(1) 002052 000000
(1) 002054 000000
(1) 002056 000000
(1) 002060 000000
(1) 002062 000000
(1) 002064 000000
(1) 002066 000000
(1) 002070 000000
(1) 002072 000000
(1) 002074 000000
(1) 002076 000000
(1) 002100 000000
(1) 002102 000000
(1) 002104 000000
(1) 002106 000000
(1) 002110 000000
(1) 002112 000000
(1) 002114 000000
```

:TIME TABLE FOR RELATIVE TIMING TESTS
:-----

```
TMTBL:
T50: 0
T75: 0
T110: 0
T134: 0
T150: 0
T300: 0
T600: 0
T1200: 0
T1800: 0
T2000: 0
T2400: 0
T3600: 0
T4800: 0
T7200: 0
T9600: 0
TEIGHT:0
TSEVEN: 0
TSIX: 0
TFIVE: 0
```



```
(1) 002354 100004          BPL      158           ;IF NOT, GO CHECK FOR AUTO-SIZING
(1) 002356 004737 011434   JSR      PC,SETAPT    ;OTHERWISE, GO DO APT SIZING FROM ETABLE
(1) 002362 000137 003706   JMP      1058         ;GO PRINT DZV STATUS TABLE
(1) 002366 032737 000040 000052 158:  BIT      @R15,@#52    ; ARE WE UNDER UFD ?
(1) 002374 001011          BNE      178           ;YES, THEN GO AUTOSIZE THE SETUP
(1) 002376 005737 000042   TST      @#42         ; CHAINED UNDER XXDP ??           ;;GPA
(1) 002402 001404          BEQ      168           ; BR IF NOT                         ;;GPA
(1) 002404 004737 011434   CALL     SETAPT       ; YES, SET-UP FROM ETABLE         ;;GPA
(1) 002410 000137 003706   JMP      1058         ; AND PROCEED                       ;;GPA
(1) 002414 004737 007346   CALL     GETSWR       ; GET INITIAL SWITCH SETTING.    ;;GPA
(1) 002420 032777 000001 176656 178:  BIT      #SW00,@SWR   ;RESELECT ?
(1) 002426 001002          BNE      208           ;IF YES, GO SET UP THE INFORMATION
(1) 002430 000137 002732   JMP      558          ;IF NO, SKIP THE INTERROGATION
(1) 002434 012700 001500 208:  MOV      #DZV.MAP,RO  ;POINT TO THE BEGINNING OF THE MAP TABLE
(1) 002440 105037 001423   CLRB     HDRFLG       ;MAKE SURE A MAP GETS PRINTED
(1) 002444 005020 258:  CLR      (RO)+        ;CLEAR A TABLE LOCATION
(1) 002446 020027 001740   CMP      RO,#DZV.END  ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
(1) 002452 001374          BNE      258          ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
(1) 002454 105337 001422   DECB     INIFLG       ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!
```

```
(1) ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
(1) ;TABLE AND SET UP THE DIAGNOSTIC.
```

```
(1) ;GET THE BASE ADDRESS OF THE DZV11'S
(1) GETCSR= .           ; POINTER FOR FALCON TWEAKER           ;;GPA
(2) 002460 104403   INSTR          ;CALL THE STRING INPUT ROUTINE
(2) 002462 003152   918           ;POINTER TO MESSAGE TO BE PRINTED
(2) 002464 104405   PARAM         ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2) 002466 160000   160000        ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 002470 167770   167770        ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 002472 001500   DZCRO         ;POINTER TO MAP LOCATION TO BE FILLED
(2) 002474 007         .BYTE 7        ;MASK OF INVALID BITS FOR THIS PARAMETER
(2) 002475 001         .BYTE 1        ;NUMBER OF PARAMETERS TO STORE
(1) 002476 013737 001500 001174  MOV      DZCRO,$BASE ;COPY BASE ADDRESS TO ETABLE
```

```
(1) ;GET THE BASE VECTOR ADDRESS
(1) GETVEC= .           ; POINTER FOR FALCON TWEAKER           ;;GPA
(2) 002504 104403   INSTR          ;CALL THE STRING INPUT ROUTINE
(2) 002506 003216   928           ;POINTER TO MESSAGE TO BE PRINTED
(2) 002510 104405   PARAM         ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2) 002512 000300   300           ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 002514 000776   776           ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 002516 001502   DZVCO         ;POINTER TO MAP LOCATION TO BE FILLED
(2) 002520 003         .BYTE 3        ;MASK OF INVALID BITS FOR THIS PARAMETER
(2) 002521 001         .BYTE 1        ;NUMBER OF PARAMETERS TO STORE
(1) 002522 013737 001502 001170  MOV      DZVCO,$VECT1 ;COPY VECTOR TO ETABLE
```

```
(1) ;GET THE MODE OF OPERATION (E,I,S)
(2) 002530 104403   INSTR          ;CALL THE STRING INPUT ROUTINE
(2) 002532 003445   968           ;POINTER TO THE MESSAGE TO BE PRINTED
(2) 002534 104406   SETFLG        ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
(2) 002536 001510   MANTO         ;THIS IS THE FLAG BEING SETUP
```

```
(1) ;GET THE NUMBER OF DZV11'S RUNNING
(2) 002540 104403   INSTR          ;CALL THE STRING INPUT ROUTINE
(2) 002542 003402   958           ;POINTER TO MESSAGE TO BE PRINTED
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-19
 CVDZAD.P11 11-MAR-83 10:06 PROGRAM INITIALIZATION AND START UP.

(2)	002544	104405				PARAM				:CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)	002546	000001				1				:LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)	002550	000020				16.				:HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)	002552	001344				STMP1				:POINTER TO MAP LOCATION TO BE FILLED
(2)	002554	000				.BYTE	0			:MASK OF INVALID BITS FOR THIS PARAMETER
(2)	002555	001				.BYTE	1			:NUMBER OF PARAMETERS TO STORE
(1)										
(1)	002556	012737	000017	001504		MOV	#17,LINE0			:SET UP DEFAULT LINES
(1)	002564	012737	017470	001506		MOV	#17470,PAR0			:SET UP DEFAULT LPR PARAMETER
(1)										:RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
(1)	002572	032777	000010	176504		BIT	#SW03,@SWR			:DO YOU WANT PARAMETERS?
(1)	002600	001402				BEQ	30\$:IF NO, SKIP THE PARAMETER CALL
(1)	002602	004737	002762			JSR	PC,65\$:GET PARAMETERS
(1)	002606	012737	000001	001410	30\$:	MOV	#1,SAVACTV			:INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
(1)	002614	113737	001344	001414		MOVB	STMP1,DZVNUM			:COPY THE NUMBER OF DEVICES
(1)	002622	005337	001344		35\$:	DEC	STMP1			:STMP1 CONTAINS THE COUNT OF UNINITIALIZED
(1)	002626	001404				BEQ	40\$:SELECTED DEVICES
(1)	002630	000261				SEC				:SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
(1)	002632	006137	001410			ROL	SAVACTV			:POINT TO THE NEXT DEVICE
(1)	002636	000771				BR	35\$:GO DO THIS PROCEDURE AGAIN
(1)	002640	013737	001410	001346	40\$:	MOV	SAVACTV,STMP2			:# OF TIMES
(1)	002646	012700	001500			MOV	#DZCRO,R0			:SET A POINTER TO THE SPECIFIED INFORMATION
(1)	002652	012701	001512			MOV	#DZCR1,R1			:POINT R1 TO THE REST OF THE MAP TABLE
(1)	002656	012702	001204			MOV	#SDDW0,R2			:POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
(1)	002662	000241				CLC				:INITIALIZE THE "C" BIT FOR A ROTATION
(1)	002664	006037	001346			ROR	STMP2			:SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
(1)	002670	006237	001346		45\$:	ASR	STMP2			:ISOLATE A SELECTION FLAG IN THE "C" BIT
(1)	002674	103404				BCS	50\$:IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
(1)	002676	012711	177777			MOV	#-1,(R1)			:TERMINATE THE LIST
(1)	002702	000137	003650			JMP	100\$:GO TO THE NEXT BLOCK
(1)	002706	012011			50\$:	MOV	(R0)+,(R1)			:ADDRESS
(1)	002710	062721	000010			ADD	#10,(R1)+			:POINT TO THE NEXT DZV11 ADDRESS VALUE
(1)	002714	012011				MOV	(R0)+,(R1)			:VECTOR
(1)	002716	062721	000010			ADD	#10,(R1)+			:POINT TO THE NEXT VECTOR VALUE
(1)	002722	012021				MOV	(R0)+,(R1)+			:LINES
(1)	002724	012021				MOV	(R0)+,(R1)+			:PARAMETERS
(1)	002726	012021				MOV	(R0)+,(R1)+			:MAINTENANCE MODE
(1)	002730	000757				BR	45\$			
(1)	002732	032777	000010	176344	55\$:	BIT	#SW03,@SWR			:ASK PARAMETERS ?
(1)	002740	001002				BNE	60\$:IF NO, GO DO AUTO SIZING
(1)	002742	000137	003650			JMP	100\$:GO SET UP FOR AUTO SIZING
(1)	002746	004737	002762		60\$:	JSR	PC,65\$:GO ASK PARAMETERS
(1)	002752	105337	001422			DECB	INIFLG			:INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
(1)	002756	000137	003706			JMP	105\$:GO TO THE NEXT BLOCK
(1)										
(1)										:GET THE ACTIVE LINES PARAMETER
(1)										
(1)					65\$:					
(1)	002762									
(2)	002762	104403				INSTR				:CALL THE STRING INPUT ROUTINE
(2)	002764	003257				93\$:POINTER TO MESSAGE TO BE PRINTED
(2)	002766	104405				PARAM				:CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)	002770	000001				1				:LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)	002772	000017				17				:HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)	002774	001504				LINE0				:POINTER TO MAP LOCATION TO BE FILLED
(2)	002776	360				.BYTE	360			:MASK OF INVALID BITS FOR THIS PARAMETER
(2)	002777	001				.BYTE	1			:NUMBER OF PARAMETERS TO STORE

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-20
 CVDZAD.P11 11-MAR-83 10:06 PROGRAM INITIALIZATION AND START UP.

```

(1) 003000 105037 001423          CLRB   HDRFLG          ;MAKE SURE THE CHANGES ARE PRINTED
(1)
(1)
(1)
(1)
(1)
(1) 003004 005737 001510          TST    MANTO          ;IS STAGGERED THE MODE OF OPERATION?
(1) 003010 100021                    BPL    85$            ;IF NOT, SKIP THIS SEGMENT
(1) 003012 013703 001504          MOV    LINE0,R3      ;GET A SCRATCH COPY OF THE ACTIVE LINES
(1) 003016 006003          70$:  ROR    R3        ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
(1) 003020 103410          BCS    80$            ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
(1) 003022 001414          BEQ    85$            ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
(1) 003024 006203          ASR    R3            ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
(1) 003026 103373          BCC    70$           ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
(1) 003030 104402 001356          75$:  TYPE   ,SQUES    ;THIS IS AN INCORRECT PARAMETER
(1) 003034 104402 010346          TYPE   ,MBADLN      ;LET THE USER KNOW ABOUT IT
(1) 003040 000750          BR     65$           ;GO GET THE CORRECT PARAMETER
(1) 003042 001772          80$:  BEQ    75$            ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
(1) 003044 006203          ASR    R3            ;GET THE NEXT FLAG
(1) 003046 103370          BCC    75$           ;IF IT ISN'T SET, THERE'S AN ERROR
(1) 003050 000241          CLC                    ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
(1) 003052 000761          BR     70$           ;GO TEST THE NEXT PAIR OF FLAGS
(1)
(1)
(1)
(1)
(1) 003054          85$:
(2) 003054 104403          INSTR
(2) 003056 003332          94$
(2) 003060 104405          PARAM
(2) 003062 000000          0
(2) 003064 000017          17
(2) 003066 001506          PARO
(2) 003070 000          .BYTE 0
(2) 003071 001          .BYTE 1
(1) 003072 012702 001504          MOV    #LINE0,R2    ;POINT TO THE LINE SELECTION PARAMETER
(1) 003076 012703 001506          MOV    #PARO,R3     ;POINT TO THE CHOSEN PARAMETERS
(1) 003102 011304          MOV    (R3),R4      ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
(1) 003104 006304          ASL    R4           ;ALIGN INDEX ON WORD BOUNDARY
(1) 003106 016437 017462 006474          MOV    DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
(1) 003114 000313          SWAB   (R3)         ;PLACE IN HIGH BYTE
(1) 003116 052713 010070          BIS    #10070,(R3)  ;PLACE EXTRA PARAMETERS INTO LOC
(1) 003122 011262 000012          90$:  MOV    (R2),12(R2)  ;LOAD THE LINES
(1) 003126 011363 000012          MOV    (R3),12(R3)  ;LOAD THE PARAMETERS
(1) 003132 062702 000012          ADD    #12,R2       ;POINT TO THE NEXT SET
(1) 003136 062703 000012          ADD    #12,R3       ;.. OF BOTH PARAMETERS
(1) 003142 020327 001734          CMP    R3,#PAR17    ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
(1) 003146 001365          BNE    90$          ;IF NOT, GO LOAD SOME MORE PARAMETERS
(1) 003150 000207          RTS    PC           ;RETURN TO CALLING BLOCK
(1) 003152 030600 052123 041440 91$: .ASCIZ <200>/1ST CSR ADDRESS (160000:167770): /
(1) 003216 030600 052123 053040 92$: .ASCIZ <200>/1ST VECTOR ADDRESS (300:770): /
(1) 003257 200 044514 042516 93$: .ASCIZ <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:17): /
(1) 003332 042200 043105 052501 94$: .ASCIZ <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17): /
(1) 003402 021600 047440 020106 95$: .ASCIZ <200>/# OF DZV11'S <IN OCTAL> (1:20): /
(1) 003445 200 040515 047111 96$: .ASCII <200>/MAINTENANCE MODE/
(1) 003466 020200 042533 052130 .ASCII <200>/ [EXTERNAL <H325> (E)]/
(1) 003522 020200 044533 052116 .ASCII <200>/ [INTERNAL <DZVCSR03=1>(I)]/
(1) 003557 200 055440 052123 .ASCIZ <200>/ [STAGGERED <H329> (S)]: /

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-21
 CVDZAD.P11 11-MAR-83 10:06 PROGRAM INITIALIZATION AND START UP.

```

(1) 003616 042600 052116 051105 97$: .ASCIZ <200>/ENTER DELAY PARAMETER: /
(1) 003650 003650 .EVEN
(1) 003650 122737 000377 001422 100$: CMPB #377,INIFLG ;ONLY DO AUTO SIZE ON 1ST START
(1) 003656 001013 BNE 105$ ;
(1) 003660 032777 000200 175416 BIT #BIT7,@SWR ;BIT7=1??
(1) 003666 001007 BNE 105$ ;BR IF NO AUTO SIZE
(1) 003670 005737 017536 TST KXTFLAG ; FALCON ?? ;:GPA
(1) 003674 001402 BEQ 1002$ ; SKIP NEXT IF NOT. ;:GPA
(1) 003676 000137 002434 JMP 20$ ; YES, DON'T AUTO-SIZE. ;:GPA
(1) 003702 1002$: JSR PC,AUTO.SIZE ;GO DO THE AUTO SIZE ;:GPA
(1) 003706 004737 011562 105$: TSTB HDRFLG ;HAS THE TABLE BEEN TYPED YET?
(1) 003712 105737 001423 BNE 120$ ;IF SO, DON'T TYPE IT AGAIN
(1) 003714 001021 DECIB HDRFLG ;INDICATE THAT THE TABLE WILL BE TYPED
(1) 003714 105337 001423 TYPE ,XHEAD ;TYPE MAP HEADER
(1) 003720 104402 010320 MOV #DZV.MAP,RO ;SET POINTER
(1) 003724 012700 001500 MOV RO,$TMP1 ;POINT TO THE MAP LOCATION
(1) 003730 010037 001344 110$: MOV (RO)+,$TMP2 ;SET DATA
(1) 003734 012037 001346 CMP #-1,$TMP2 ;END OF LIST?
(1) 003740 022737 177777 001346 BEQ 120$ ;BR IF YES
(1) 003746 001403 115$: CONVRT ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
(1) 003750 104411 XSTATQ ;CONVERT THE DATA AT THIS ADDRESS
(1) 003752 010410 BR 110$ ;GO PRINT THE NEXT PARAMETER
(1) 003754 000765 MOV SAVACTV,DZVACTV ;COPY BIT MAP OF SYSTEM DEVICES ACTIVE
(1) 003756 013737 001410 001406 120$: MOV DZVNUM,SAVNO ;COPY NO. OF SYSTEM DEVICES ACTIVE
(1) 003764 113737 001414 001416 BIT #SW06,@SWR ;DESELECT SPECIFIC DEVICES??
(1) 003772 032777 000100 175304 BEQ 135$ ;BR IF NO.
(1) 004000 001431 121$: INSTR ;CALL THE STRING INPUT ROUTINE
(2) 004002 104403 MNEW ;POINTER TO MESSAGE TO BE PRINTED
(2) 004004 010236 PARAM ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2) 004006 104405 1 ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 004010 000001 177777 ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 004012 177777 DZVACTV ;POINTER TO MAP LOCATION TO BE FILLED
(2) 004014 001406 .BYTE 0 ;MASK OF INVALID BITS FOR THIS PARAMETER
(2) 004016 000 .BYTE 1 ;NUMBER OF PARAMETERS TO STORE
(2) 004017 001 CMP DZVACTV,SAVACTV ;IS THE VALUE VALID?
(1) 004020 023737 001406 001410 BLOS 122$ ;BRANCH IF YES
(1) 004026 101403 TYPE ,MERR3 ;IF NOT THEN TYPE ERROR
(1) 004030 104402 010110 BR 121$ ;GO REASK QUESTION
(1) 004034 000762 CLRIB SAVNO ;CLEAR NO. OF DEVICES BEING TESTED
(1) 004036 105037 001416 001344 122$: MOV DZVACTV,$TMP1 ;COPY BIT MAP OF ACTIVE DEVICES BEING TESTED
(1) 004042 013737 001406 001344 126$: ASR $TMP1 ;SHIFT OUT AN ACTIVE BIT
(1) 004050 006237 001344 BCC 127$ ;IF NOT ACTIVE SKIP INCREMENT
(1) 004054 103002 INCB SAVNO ;IF ACTIVE RECORD IT
(1) 004056 105237 001416 BNE 127$ ;IF ALL ACTIVE BITS RECORDED DON'T BRANCH
(1) 004062 001372 127$: BIT #SW04,@SWR ;CHECK TO SEE IF DELAY COUNT CHANGES
(1) 004064 032777 000020 175212 135$: BEQ 140$ ;IF NOT, GO CLEAR VECTOR AREA
(2) 004072 001407 INSTR ;CALL THE STRING INPUT ROUTINE
(2) 004074 104403 97$ ;POINTER TO MESSAGE TO BE PRINTED
(2) 004076 003616 PARAM ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2) 004100 104405 1 ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 004102 000001 177777 ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2) 004104 177777 DLYCNT ;POINTER TO MAP LOCATION TO BE FILLED
(2) 004106 006474 .BYTE 0 ;MASK OF INVALID BITS FOR THIS PARAMETER
(2) 004110 000

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-22
 CVDZAD.P11 11-MAR-83 10:06 PROGRAM INITIALIZATION AND START UP.

```

(2) 004111      001
(1) 004112 012700 000300
(1) 004116 012701 000302
(1) 004122 010120
(1) 004124 005021
(1) 004126 022021
(1) 004130 005737 017536
(1) 004134 001403
(1) 004136 020027 000400
(1) 004142 000402
(1) 004144
(1) 004144 022700 001000
(1) 004150 001364
(1)
(1)
(1)
(1)
(1) 004152 012706 001120
(1) 004156 106427 000200
(1) 004162 005737 000042
(1) 004166 001015
(1) 004170 032777 000004 175106
(1) 004176 001406
(1) 004200 104402 010134
(1) 004204 012737 000240 004474
(1) 004212 000403
(1) 004214 013737 004722 004474
(1) 004222 012737 010722 001252
(1) 004230 113737 001416 001415
(1) 004236 104402 010025
(1) 004242 000177 175004

      .BYTE      1
140$: MOV      #300,R0      ;NUMBER OF PARAMETERS TO STORE
      MOV      #302,R1      ;PREPARE TO CLEAR THE FLOATING
145$: MOV      R1,(R0)+     ;VECTOR AREA. 300-776
      CLR      (R1)+        ;START PUTTING 'PC+2 - HALT'
      CMP      (R0)+,(R1)+  ;IN VECTOR AREA.
      TST      KXTFLAG      ;POP POINTERS
      BEQ      1001$        ; IF FALCON...
      CMP      R0,#400      ;...STOP AT 400.
      402              ; SKIP NEXT
1001$: CMP      #1000,R0    ;ALL DONE??
      BNE      145$        ;BR IF NO.

      ;TEST START AND RESTART
      ;-----

.BEGIN: MOV      #STACK,SP  ;SET UP STACK
      MTPS     #MASK        ;LOCK OUT INTERRUPTS
      TST      @#42         ;IS PROGRAM UNDER MONITOR CONTROL
      BNE      2$          ;BR IF YES
      BIT      #BIT2,@SWR   ;CHECK FOR LOCK ON TEST
      BEQ      1$          ;BR IF NO LOCK DESIRED.
      TYPE     ,MLOCK       ;TYPE LOCK SELECTED.
      MOV      #NOP,TTST    ;ADJUST SCOPE ROUTINE.
      BR      2$           ;CONTINUE ALONG.
1$:  MOV      BRW,TTST      ;PREPARE NURMAL SCOPE ROUTINE
2$:  MOV      #CYCLE,$LPADR ;START AT 'CYCLE' FIND WHICH DEVICE TO TEST
      MOVB     SAVNO,SAVNUM ;COPY ACTIVE DEVICES BEING TESTED
      TYPE     ,MR          ;TYPE 'RUNNING'
      JMP      @SLPADR      ;START TESTING
    
```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-23
CVDZAD.P11 11-MAR-83 10:06 END OF PASS ROUTINE

8731
(2)
(2)
(2)
(2)
(3)
(3)
(4)
(3)
(3)
(3)
(3)
(3)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(5)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(2)
(2)
(2)
(2)
(2)

:END OF PASS
:TYPE NAME OF TEST
:UPDATE PASS COUNT
:CHECK FOR EXIT TO ACT-11
:RESTART TEST
.SBTTL END OF PASS ROUTINE
:*****
:*INCREMENT THE PASS NUMBER (\$PASS)
:*IF THERES A MONITOR GO TO IT
:*IF THERE ISN'T JUMP TO CYCLE

SEOP:

SCOPE
CLR SERRPC ;CLEAR LAST ERROR PC
CLRB SERFLG ;CLEAR ERROR FLAG
TYPE ,MEPASS ;TYPE END PASS
TYPE ,MCSRX ;TYPE CSR
CNVRT ,XCSR ;SHOW IT
TYPE ,MVECX ;TYPE VECTOR
CNVRT ,XVEC ;SHOW IT
INC \$PASS ;RAISE PASS COUNT
TYPE ,MPASSX ;TYPE PASSES
CNVRT ,XPASS ;SHOW IT
DEC \$PASS ;RESTORE PASS COUNT
TYPE ,MERRX ;TYPE ERRORS
CNVRT ,XERR ;SHOW IT
INC \$DEVCT ;INC DEVCNT FOR APT
DECB SAVNUM ;ARE ALL DEVICES TESTED?
BNE \$DOAGN ;BR IF NO.
MOVB SAVNO, SAVNUM ;RESTORE THE COUNT
CLR \$TIMES ;ZERO THE NUMBER OF ITERATIONS
INC \$PASS ;INCREMENT THE PASS NUMBER
BIC #100000, \$PASS ;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;LOOP?
SEOPCT: .WORD 1
BGT \$DOAGN ;:YES
MOV (PC)+, @ (PC)+ ;:RESTORE COUNTER
SENDCT: .WORD 1
SGET42: MOV @#42, R0 ;:GET MONITOR ADDRESS
BEQ \$DOAGN ;:BRANCH IF NO MONITOR
RESET ;:CLEAR THE WORLD
SENDAD: JSR PC, (R0) ;:GO TO MONITOR
NOP ;:SAVE ROOM
NOP ;:FOR
NOP ;:ACT11
SDOAGN: ;
SRTNAD: JMP @ (PC)+ ;:RETURN
.WORD CYCLE
XCSR: 1
.BYTE 6, 2
DZVCSR
XVEC: 1
.BYTE 3, 2

000042

001415

001126

002

002

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-24
CVDZAD.P11 11-MAR-83 10:06 END OF PASS ROUTINE

(2) 004444 002040
(2) 004446 000001
(2) 004450 006 002
(2) 004452 001126
(2) 004454 000001
(2) 004456 006 002
(2) 004460 001256

DZVRIV
XPASS: 1
.BYTE 6,2
\$PASS
XERR: 1
.BYTE 6,2
\$ERTTL

:SCOPE LOOP AND ITERATION HANDLER
:-----

.SBTTL SCOPE HANDLER ROUTINE

: *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
: *AND LOAD THE TEST NUMBER(\$TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
: *AND LOAD THE ERROR FLAG (\$ERFLG) INTO DISPLAY<15:08>
: *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
: *SW14=1 LOOP ON TEST
: *SW11=1 INHIBIT ITERATIONS
: *CALL
: * SCOPE ;:SCOPE=IOT

(3) 004462
(3) 004462 005037 001262
(5) 004466 022716 012274
(5) 004472 001413
(5) 004474 000406
(5) 004476 105777 174606
(5) 004502 100067
(5) 004504 017766 174602 177776
(3) 004512 032777 040000 174564
(3) 004520 001060
(3) 004522 000416
(3) 004524 013746 000004
(3) 004530 012737 004550 000004
(3) 004536 005737 177060
(3) 004542 012637 000004
(3) 004546 000436
(3) 004550 022626
(3) 004552 012637 000004
(3) 004556 000441
(3) 004560
(3) 004560 105737 001247
(3) 004564 001404
(3) 004566 105037 001247
(3) 004572 005037 001354
(3) 004576 032777 004000 174500
(3) 004604 001011
(3) 004606 005737 001126
(3) 004612 001406
(3) 004614 005237 001250
(3) 004620 023737 001354 001250
(3) 004626 002015

\$SCOPE:
.SCOPE: CLR \$ERRPC ;CLEAR LAST ERROR PC.
CMP #TST1+2,(SP) ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
BEQ \$XTSTR ;IF SO, DON'T LOOP ON IT
TTST: BR 1\$;GOTO 1\$ (IF LOCK SW02=1; THIS LOC =240)
TSTB @STKS ;KEYBOARD DONE?
BPL \$OVER ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
MOV @STKB,-2(SP) ;CLEAR DONE BIT
1\$: BIT #BIT14,@SWR ;:LOOP ON PRESENT TEST?
BNE \$OVER ;:YES IF SW14=1
:*****START OF CODE FOR THE XOR TESTER*****
\$XTSTR: BR 6\$;:IF RUNNING ON THE 'XOR' TESTER CHANGE
;:THIS INSTRUCTION TO A 'NOP' (NOP=240)
MOV @#ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
MOV #5\$,@#ERRVEC ;:SET FOR TIMEOUT
TST @#177060 ;:TIME OUT ON XOR?
MOV (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
BR \$SVLAD ;:GO TO THE NEXT TEST
5\$: CMP (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
MOV (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
BR \$OVER ;:LOOP ON THE PRESENT TEST
6\$: *****END OF CODE FOR THE XOR TESTER*****
2\$: TSTB \$ERFLG ;:HAS AN ERROR OCCURRED?
BEQ 3\$;:BR IF NO
4\$: CLRB \$ERFLG ;:ZERO THE ERROR FLAG
CLR \$TIMES ;:CLEAR THE NUMBER OF ITERATIONS TO MAKE
3\$: BIT #BIT11,@SWR ;:INHIBIT ITERATIONS?
BNE 1\$;:BR IF YES
TST \$PASS ;:IF FIRST PASS OF PROGRAM
BEQ 1\$;: INHIBIT ITERATIONS
INC \$ICNT ;:INCREMENT ITERATION COUNT
CMP \$TIMES,\$ICNT ;:CHECK THE NUMBER OF ITERATIONS MADE
BGE \$OVER ;:BR IF MORE ITERATION REQUIRED

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-25
CVDZAD.P11 11-MAR-83 10:06

SCOPE HANDLER ROUTINE

```

(3) 004630 012737 000001 001250 1$: MOV #1,SICNT ;;REINITIALIZE THE ITERATION COUNTER
(3) 004636 013737 004724 001354 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
(3) 004644 105237 001246 SSVLAD: INCB $STNM ;;COUNT TEST NUMBERS
(3) 004650 113737 001246 001124 MOV $STNM,$STESTN ;;SET TEST NUMBER IN APT MAILBOX
(3) 004656 011637 001252 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
(3) 004662 013777 001246 174416 SOVER: MOV $STNM,@DISPLAY ;;DISPLAY TEST NUMBER
(3) 004670 013716 001252 MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
(5) 004674 004737 007320 JSR PC,SERV.G ;;FIND OUT IF ^G WAS TYPED
(5) 004700 105037 001424 CLRB MNTFLG ;;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
(5) 004704 005737 001372 TST MODE ;;HAS THE MODE BEEN CHANGED?
(5) 004710 001003 BNE 4$ ;;IF NOT INTERNAL , GO DO A TEST
(5) 004712 112737 000010 001424 MOV $MAINT,MNTFLG ;;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
(5) 004720 000002 4$: RTI ;;GO DO THE TEST
(5) 004722 000406 BRW: 406
(3) 004724 000005 SMXCNT: 5 ;;MAX. NUMBER OF ITERATIONS

```

;;CHECK FOR FREEZE ON CURRENT DATA
:-----

```

(1) 004726 032777 001000 174350 .SCOP1: BIT #SW09,@SWR ;;IS SW09=1(SET)?
(1) 004734 001405 BEQ 1$ ;;BR IF NOT SET.
(1) 004736 005737 001364 TST LOCK ;;IS THERE A TIGHT LOOP SPECIFIED?
(1) 004742 001402 BEQ 1$ ;;IF NO, RETURN
(1) 004744 013716 001364 MOV LOCK,(SP) ;;IF YES, GOTO THE ADDRESS IN LOCK.
(1) 004750 000002 1$: RTI ;;GO BACK.
(1) 004752 032737 000040 000052 .TYPE: BIT #BITS,@#52 ;;ARE WE UNDER UFD ?
(1) 004760 001004 BNE 1$ ;;YES,THEN NO TYPEOUT
(1) 004762 032777 010000 174314 BIT #SW12,@SWR ;;INHIBIT ALL PRINTOUT??
(1) 004770 001403 BEQ $TYPE ;;IF NOT, GO TYPE
(1) 004772 062716 000002 1$: ADD #2,(SP) ;;SKIP OVER MESSAGE POINTER
(1) 004776 000002 RTI ;;RETURN TO WHERE PROCEDURE WAS INVOKED

```

.SBTTL TYPE ROUTINE

```

(2) ;;*****
(2) ;;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(2) ;;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(2) ;;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(2) ;;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(2) ;;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(2) ;;*
(2) ;;*CALL:
(2) ;;*1) USING A TRAP INSTRUCTION
(2) ;;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(2) ;;*OR
(2) ;;* TYPE
(2) ;;* MESADR
(2) ;;*

```

```

(2) 005000 105737 001323 $TYPE: TSTB $TPFLG ;;IS THERE A TERMINAL?
(2) 005004 100002 BPL 1$ ;;BR IF YES
(2) 005006 000000 HALT ;;HALT HERE IF NO TERMINAL
(2) 005010 000430 BR 3$ ;;LEAVE
(2) 005012 010046 1$: MOV RO,-(SP) ;;SAVE RO
(2) 005014 017600 MOV @2(SP),RO ;;GET ADDRESS OF ASCIZ STRING
(2) 005020 122737 000001 001140 CMPB #APTENV,$ENV ;;RUNNING IN APT MODE

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-27
 CVDZAD.P11 11-MAR-83 10:06 TYPE ROUTINE

```

(2) 005262          102$:
(2) 005262 005726      TST      (SP)+      ;;FIX STACK      ;MJD001
(2) 005264          10$:      TSTB     @STPS      ;;WAIT UNTIL PRINTER IS READY ;MJD001
(2) 005264 105777 174024      BPL      10$
(2) 005270 100375      MOVB     2(SP),@STPB  ;;LOAD CHAR TO BE TYPED INTO DATA REG. ;MJD001
(2) 005272 116677 000002 174016      CMPB     #CR,2(SP)  ;;IS CHARACTER A CARRIAGE RETURN?
(2) 005300 122766 000015 000002      BNE      1$        ;;BRANCH IF NO
(2) 005306 001003      CLRB     $CHARCNT  ;;YES--CLEAR CHARACTER COUNT
(2) 005310 105037 005330      BR       $TYPEX    ;;EXIT
(2) 005314 000406      CMPB     #LF,2(SP)  ;;IS CHARACTER A LINE FEED?
(2) 005316 122766 000012 000002 1$:      BEQ      $TYPEX    ;;BRANCH IF YES
(2) 005324 001402      INCB     (PC)+     ;;COUNT THE CHARACTER
(2) 005326 105227      SCHARNT: .WORD    0  ;;CHARACTER COUNT STORAGE
(2) 005330 000000      STYPEX: RTS      PC
(2) 005332 000207
    
```

.SBTTL APT COMMUNICATIONS ROUTINE

```

(2) 005334 112737 000001 005600 *****
(2) 005342 112737 000001 005576 SATY1:  MOVB   #1,$FFLG  ;;TO REPORT FATAL ERROR
(2) 005350 000403      SATY3:  MOVB   #1,$MFLG  ;;TO TYPE A MESSAGE
(2) 005352 112737 000001 005600      BR       $ATYC
(2) 005360      SATY4:  MOVB   #1,$FFLG  ;;TO ONLY REPORT FATAL ERROR
(2) 005360 010046      SATYC:
(4) 005362 010146      MOV     RO,-(SP)    ;;PUSH RO ON STACK
(2) 005364 105737 005576      MOV     R1,-(SP)    ;;PUSH R1 ON STACK
(2) 005370 001450      TSTB   $MFLG      ;;SHOULD TYPE A MESSAGE?
(2) 005372 122737 000001 001140      BEQ     5$        ;;IF NOT: BR
(2) 005400 001031      CMPB   #APTENV,$ENV  ;;OPERATING UNDER APT?
(2) 005402 132737 000100 001141      BNE     3$        ;;IF NOT: BR
(2) 005410 001425      BITB   #APTPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
(2) 005412 017600 000004      BEQ     3$        ;;IF NOT: BR
(2) 005416 062766 000002 000004      MOV     @4(SP),RO   ;;GET MESSAGE ADDR.
(2) 005424 005737 001120      ADD     #2,4(SP)    ;;BUMP RETURN ADDR.
(2) 005430 001375      1$:     TST     $MSGTYPE  ;;SEE IF DONE W/ LAST XMISSION?
(2) 005432 010037 001134      BNE     1$        ;;IF NOT: WAIT
(2) 005436 105720      2$:     MOV     RO,$MSGAD  ;;PUT ADDR IN MAILBOX
(2) 005440 001376      TSTB   (RO)+      ;;FIND END OF MESSAGE
(2) 005442 163700 001134      BNE     2$
(2) 005446 006200      SUB     $MSGAD,RO  ;;SUB START OF MESSAGE
(2) 005450 010037 001136      ASR     RO        ;;GET MESSAGE LNGTH IN WORDS
(2) 005454 012737 000004 001120      MOV     RO,$MSGGLT  ;;PUT LENGTH IN MAILBOX
(2) 005462 000413      MOV     #4,$MSGTYPE  ;;TELL APT TO TAKE MSG.
(2) 005464 017637 000004 005510 3$:     BR       5$
(2) 005472 062766 000002 000004      MOV     @4(SP),4$  ;;PUT MSG ADDR IN JSR LINKAGE
(4) 005500 013746 177776      ADD     #2,4(SP)    ;;BUMP RETURN ADDRESS
(2) 005504 004737 005000      MOV     177776,-(SP)  ;;PUSH 177776 ON STACK
(2) 005510 000000      JSR    PC,$TYPE  ;;CALL TYPE MACRO
(2) 005512      4$:     .WORD    0
(2) 005512      5$:
(2) 005512 105737 005600      10$:   TSTB   $FFLG      ;;SHOULD REPORT FATAL ERROR?
(2) 005516 001416      BEQ     12$      ;;IF NOT: BR
(2) 005520 005737 001140      TST     $ENV      ;;RUNNING UNDER APT?
(2) 005524 001413      BEQ     12$      ;;IF NOT: BR
(2) 005526 005737 001120      11$:   TST     $MSGTYPE  ;;FINISHED LAST MESSAGE?
(2) 005532 001375      BNE     11$      ;;IF NOT: WAIT
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-28
 CVDZAD.P11 11-MAR-83 10:06 APT COMMUNICATIONS ROUTINE

```

(2) 005534 017637 000004 001122      MOV    @4(SP),SFATAL    ;;GET ERROR #
(2) 005542 062766 000002 000004      ADD    #2,4(SP)        ;;BUMP RETURN ADDR.
(2) 005550 005237 001120                INC    SMSGTYPE        ;;TELL APT TO TAKE ERROR
(2) 005554 105037 005600      12$:  CLR  SF  FLG        ;;CLEAR FATAL FLAG
(2) 005560 105037 005577                CLR  SL  FLG        ;;CLEAR LOG FLAG
(2) 005564 105037 005576                CLR  SM  FLG        ;;CLEAR MESSAGE FLAG
(4) 005570 012601                MOV    (SP)+,R1        ;;POP STACK INTO R1
(4) 005572 012600                MOV    (SP)+,R0        ;;POP STACK INTO R0
(2) 005574 000207                RTS    PC              ;;RETURN
(2) 005576      000      SMFLG: .BYTE 0        ;;MESSG. FLAG
(2) 005577      000      SLFLG: .BYTE 0        ;;LOG FLAG
(2) 005600      000      SFFLG: .BYTE 0        ;;FATAL FLAG
(2)                                .EVEN
(2)                                APTSIZE=200
(2)                                APTENV=001
(2)                                APTSPOOL=100
(2)                                APTCSUP=040
(1)
(1)                                ;STRING INPUT ROUTINE
(1)                                ;-----
(1) 005602 010346      .INSTR: MOV    R3,-(SP)    ;SAVE R3 ON STACK
(1) 005604 010446                MOV    R4,-(SP)    ;SAVE R4 ON STACK
(1) 005606 017637 000004 005624      MOV    @4(SP),.MSG   ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
(1) 005614 062766 000002 000004      ADD    #2,4(SP)    ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
(1) 005622 104402      .INST1: TYPE        ;PRINT THE MESSAGE
(1) 005624 000000      .MSG:  0           ;MESSAGE IS POINTED TO FROM HERE
(1) 005626 012704 010616                MOV    #INBUF,R4    ;POINT R4 TO THE INPUT BUFFER
(1) 005632 012703 000007                MOV    #7,R3        ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
(1) 005636 105777 173446      1$:  TSTB @STKS      ;HAS A CHARACTER BEEN RECEIVED?
(1) 005642 100375                BPL    1$           ;IF NO, KEEP WAITING FOR IT
(1) 005644 117714 173442                MOV    @STKB,(R4)   ;IF YES, SAVE IT IN THE INPUT BUFFER
(1) 005650 142714 000200                BICB  #200,(R4)    ;KEEP ONLY THE 7-BIT ASCII INFORMATION
(1) 005654 122427 000015                CMPB  (R4)+,#15    ;IS THIS CHARACTER A LINE FEED?
(1) 005660 001417                BEQ    INSTR2      ;IF SO, TERMINATE THE INPUT SEQUENCE
(1) 005662 105777 173426      2$:  TSTB @STPS      ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
(1) 005666 100375                BPL    2$           ;IF WE CAN'T, WAIT UNTIL WE CAN
(1) 005670 017777 173416 173420      MOV    @STKB,@STPB  ;ECHO THE CHARACTER BACK
(1) 005676 005303                DEC    R3           ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
(1) 005700 001356                BNE    1$           ;IF WE DON'T HAVE 7, GO GET SOME MORE
(1) 005702 012604                MOV    (SP)+,R4    ;IF WE HAVE 7, RESTORE R4
(1) 005704 012603                MOV    (SP)+,R3    ;RESTORE R3
(1) 005706 010346      .INSTE: MOV    R3,-(SP) ;SAVE R3 ON THE STACK
(1) 005710 010446                MOV    R4,-(SP)    ;SAVE R4 ON THE STACK
(1) 005712 104402 001356                TYPE  ,SQUES       ;PRINT A QUESTION MARK... WHAT'S GOING ON?
(1) 005716 000741                BR    .INST1       ;GO PRINT THE MESSAGE AGAIN
(1) 005720 012604      INSTR2: MOV    (SP)+,R4 ;RESTORE R4
(1) 005722 012603                MOV    (SP)+,R3    ;RESTORE R3
(1) 005724 000002                RTI                ;RETURN TO THE MAIN PROCEDURE
(1)
(1)                                ;CONVERT ASCII STRING TO OCTAL
(1)                                ;-----
(1) 005726 010546      .PARAM: MOV    R5,-(SP) ;SAVE R5 ON THE STACK
(1) 005730 010446                MOV    R4,-(SP)    ;SAVE R4 ON THE STACK
(1) 005732 016605 000004                MOV    4(SP),R5    ;GET THE SETUP INFORMATION POINTER
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-29
 CVDZAD.P11 11-MAR-83 10:06 APT COMMUNICATIONS ROUTINE

```

(1) 005736 012537 006116      MOV      (R5)+,LOLIM      ;SET THE LOW LIMIT FOR THE INPUT
(1) 005742 012537 006120      MOV      (R5)+,HILIM      ;SET THE HIGH LIMIT FOR THE INPUT
(1) 005746 012537 006122      MOV      (R5)+,DEVADR      ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
(1) 005752 112537 006124      MOV      (R5)+,LOBITS      ;GET THE MASK OF THE INCORRECT BITS
(1) 005756 112537 006125      MOV      (R5)+,ADRCNT      ;GET THE COUNT OF ITEMS TO BE STORED
(1) 005762 010566 000004      MOV      R5,4(SP)        ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
(1) 005766 005005                PARAM1: CLR      R5        ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
(1) 005770 012704 010616      MOV      #INBUF,R4        ;POINT TO THE INPUT BUFFER
(1) 005774 122714 000015      CMPB     #15,(R4)         ;IS THIS CHARACTER A CARRIAGE RETURN?
(1) 006000 001420                1$: BEQ      PARERR        ;IF SO, PRINT THE MESSAGE AGAIN
(1) 006002 121427 000060      CMPB     (R4),#60         ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
(1) 006006 002415                BLT      PARERR        ;IF SO, GO PRINT THE MESSAGE AGAIN
(1) 006010 121427 000067      CMPB     (R4),#67         ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
(1) 006014 003012                BGT      PARERR        ;IF SO, GO PRINT THE MESSAGE AGAIN
(1) 006016 142714 000060      BICB     #60,(R4)         ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
(1) 006022 152405                BISB     (R4)+,R5        ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
(1) 006024 122714 000015      CMPB     #15,(R4)         ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
(1) 006030 001406                BEQ      LIMITS         ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
(1) 006032 006305                ASL      R5              ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
(1) 006034 006305                ASL      R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
(1) 006036 006305                ASL      R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
(1)                                ;NEXT THREE BITS
(1) 006040 000760                BR       1$              ;GO GET THE NEXT CHARACTER
(1) 006042 104404                PARERR: INSTER          ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
(1) 006044 000750                BR       PARAM1         ;TRY GETTING THE PARAMETERS AGAIN
(1)                                ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
(1)                                ;-----
(1) 006046 020537 006120      LIMITS: CMP      R5,HILIM      ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
(1) 006052 101373                BHI      PARERR        ;IF YES, GO PRINT THE MESSAGE AGAIN
(1) 006054 020537 006116      CMP      R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
(1) 006060 103770                BLO      PARERR        ;IF YES, GO PRINT THE MESSAGE AGAIN
(1) 006062 133705 006124      BITB     LOBITS,R5        ;ARE ANY INCORRECT BITS SET IN THE RESULT?
(1) 006066 001365                BNE      PARERR        ;IF SO, GO PRINT THE MESSAGE AGAIN
(1)                                ;STORE NUMBER AT SPECIFIED ADDRESS
(1) 006070 013704 006122      1$: MOV      DEVADR,R4        ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
(1) 006074 010524                MOV      R5,(R4)+        ;STORE THE RESULT
(1) 006076 062705 000002      ADD      #2,R5            ;CALCULATE THE NEXT DATUM
(1) 006102 105337 006125      DECB     ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
(1) 006106 001372                BNE      1$              ;IF NOT, GO STORE THE NEXT DATUM
(1) 006110 012604                MOV      (SP)+,R4        ;RESTORE R4
(1) 006112 012605                MOV      (SP)+,R5        ;RESTORE R5
(1) 006114 000002                RTI                     ;RETURN TO THE MAIN PROGRAM
(1) 006116 000000                LOLIM: 0                  ;LOWEST ACCEPTABLE VALUE
(1) 006120 000000                HILIM: 0                  ;HIGHEST ACCEPTABLE
(1) 006122 000000                DEVADR: 0                 ;LOCATION WHERE RESULT WILL BE STORED
(1) 006124 000                LOBITS: .BYTE 0           ;INCORRECT BITS MASK
(1) 006125 000                ADRCNT: .BYTE 0          ;COUNT OF ITEMS TO BE STORED
(1)                                ;SAVE PC OF TEST THAT FAILED AND R0-R5
(1)                                ;-----
(1)
(1)

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-30
 CVDZAD.P11 11-MAR-83 10:06 APT COMMUNICATIONS ROUTINE

```

(1) 006126 016637 000004 001404 .SAV05: MOV 4(SP),SAVPC ;SAVE R7 (PC)
(1)
(1) ;SAVE R0-R5
(1)
(1) 006134 010537 001340 SV05: MOV R5,$REG5 ;SAVE R5
(1) 006140 010437 001336 MOV R4,$REG4 ;SAVE R4
(1) 006144 010337 001334 MOV R3,$REG3 ;SAVE R3
(1) 006150 010237 001332 MOV R2,$REG2 ;SAVE R2
(1) 006154 010137 001330 MOV R1,$REG1 ;SAVE R1
(1) 006160 010037 001326 MOV R0,$REG0 ;SAVE R0
(1) 006164 000002 RTI ;LEAVE.
(1)
(1) ;RESTORE R0-R5
(1)
(1) 006166 013700 001326 .RES05: MOV $REG0,R0 ;RESTORE R0
(1) 006172 013701 001330 MOV $REG1,R1 ;RESTORE R1
(1) 006176 013702 001332 MOV $REG2,R2 ;RESTORE R2
(1) 006202 013703 001334 MOV $REG3,R3 ;RESTORE R3
(1) 006206 013704 001336 MOV $REG4,R4 ;RESTORE R4
(1) 006212 013705 001340 MOV $REG5,R5 ;RESTORE R5
(1) 006216 000002 RTI ;LEAVE
(1)
(1) ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
(1)
(1)
(1)
(1) 006220 104402 001357 .CONVR: TYPE ;$CRLF ;PRINT A CARRIAGE RETURN
(1) 006224 010046 .CNVRT: MOV R0,-(SP) ;SAVE R0
(1) 006226 010146 MOV R1,-(SP) ;SAVE R1
(1) 006230 010346 MOV R3,-(SP) ;SAVE R3
(1) 006232 010446 MOV R4,-(SP) ;SAVE R4
(1) 006234 010546 MOV R5,-(SP) ;SAVE R5
(1) 006236 017601 000012 MOV @12(SP),R1 ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
(1) 006242 062766 000002 000012 ADD #2,12(SP) ;POINT TO WHERE MAIN PROGRAM WILL RESUME
(1) 006250 012137 006374 MOV (R1)+,WRDCNT ;GET NUMBER OF WORDS TO BE PRINTED
(1) 006254 112105 1$: MOV (R1)+,R5 ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
(1) 006256 112100 MOV (R1)+,R0 ;GET THE NUMBER OF SPACES TO PRINT
(1) 006260 013104 MOV @ (R1)+,R4 ;COPY THE WORD TO BE CONVERTED
(1) 006262 110537 006376 MOV R5,CHRCNT ;COPY THE CHARACTER COUNT
(1) 006266 010403 3$: MOV R4,R3 ;COPY THE ARGUMENT WORD AGAIN
(1) 006270 042703 177770 BIC #^C<7>,R3 ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
(1) 006274 062703 000060 ADD #060,R3 ;MAKE AN ASCII CHARACTER OUT OF THEM
(1) 006300 110346 MOV R3,-(SP) ;SAVE THAT CHARACTER
(1) 006302 006004 ROR R4 ;MOVE THE NEXT THREE BITS INTO PLACE
(1) 006304 006204 ASR R4 ;MOVE THEM AGAIN
(1) 006306 006204 ASR R4 ;AND FINALLY A THIRD TIME
(1) 006310 005305 DEC R5 ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
(1) ;BUILT?
(1) 006312 001365 BNE 3$ ;IF NO, GO BUILD THE NEXT ONE.
(1) 006314 012703 010660 MOV #MDATA,R3 ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
(1) 006320 112623 4$: MOV (SP)+,(R3)+ ;STORE THE CHARACTER, STARTING WITH THE MOST
(1) 006322 105337 006376 DECB CHRCNT ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
(1) 006326 001374 BNE 4$ ;IF NO, GO TRANSFER ANOTHER
(1) 006330 105700 TSTB R0 ;ARE ANY SPACES TO BE PRINTED?
(1) 006332 001404 BEQ 6$ ;IF NO, DON'T SET UP ANY
(1) 006334 112723 000040 5$: MOV #040,(R3)+ ;ADD A SPACE TO THE OUTPUT BUFFER
(1) 006340 105300 DECB R0 ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
    
```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-31
 CVDZAD.P11 11-MAR-83 10:06 APT COMMUNICATIONS ROUTINE

```
(1) 006342 001374      BNE      5$          ;IF YES, GO ADD ANOTHER SPACE
(1) 006344 105013      CLR      (R3)       ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
(1) 006346 104402 010660 TYPE      ,MDATA   ;PRINT THE STRING WE JUST BUILT
(1) 006352 005337 006374 DEC      W@DCNT    ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
(1) 006356 001336      BNE      1$          ;IF YES, GO CONVERT THEM
(1) 006360 012605      MOV      (SP)+,R5   ;RESTORE R5
(1) 006362 012604      MOV      (SP)+,R4   ;RESTORE R4
(1) 006364 012603      MOV      (SP)+,R3   ;RESTORE R3
(1) 006366 012601      MOV      (SP)+,R1   ;RESTORE R1
(1) 006370 012600      MOV      (SP)+,R0   ;RESTORE R0
(1) 006372 000002      RTI                       ;RETURN TO THE MAIN PROGRAM
(1) 006374 000000      WRDCNT: 0
(1) 006376      000      CHRCNT: .BYTE
(1) 006377      000      SPACNT: .BYTE 0
(1)                                         ;NUMBER OF CHARACTERS TO PRINT
(1)                                         ;NUMBER OF SPACES TO PRINT
(1) 006400 000000      BINWRD: 0
```

```
(1)                                         ;TRAP DISPATCH SERVICE
(1)                                         ;ARGUMENT OF TRAP IS EXTRACTED
(1)                                         ;AND USED AS OFFSET TO OBTAIN POINTER
(1)                                         ;TO SELECTED SUBROUTINE
```

```
(1) 006402 010046      .TRPSR: MOV      RO,-(SP)      ;SAVE RO. USE RO TO FIND TRAP ROUTINE
(1) 006404 016600 000002 MOV      2(SP),RO      ;GET TRAP ADDRESS
(1) 006410 005740      TST      -(RO)       ;GET TRAP
(1) 006412 111000      MOVB     (RO),RO     ;GET RIGHT BYTE OF TRAP (TRAP OFFSET)
(1) 006414 006300      ASL      RO         ;POSITION OFFSET FOR TABLE INDEXING
(1) 006416 016000 001742 MOV      .TRPTAB(RO),RO ;PLACE INDEXED ADDRESS OF TABLE IN RO
(1) 006422 000200      RTS      RO         ;TRANSFER TO THAT ADDRESS AND RESTORE OLD RO
```

```
(1)                                         ;DEVICE CLEAR ROUTINE
(1)                                         ;ISSUE A DEVICE CLEAR
```

```
(1) 006424      .DEVICE.CLR:
(1) 006424 052777 000020 173356 BIS      #DCLR,@DZVCSR ;SET DCLR
(1) 006432 032777 000020 173350 1$: BIT      #DCLR,@DZVCSR ;DID IT CLEAR?
(1) 006440 001374      BNE      1$          ;BR IF NO
(1) 006442 000002      RTI                       ;EXIT ROUTINE
```

```
(1)                                         ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
```

```
(1) 006444 104413      .DCLASM: DEVICE.CLR ;ISSUE A DEVICE CLEAR
(1) 006446 153777 001424 173334 BISS     MNTFLG,@DZVCSR ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
(1) 006454 000002      RTI                       ;RETURN TO CALLING ROUTINE
```

```
(1) 006456      .DELAY:
(1) 006456 010046      MOV      RO,-(SP)   ;SAVE RO
(1) 006460 013700 006474 MOV      DLYCNT,RO  ;SET COUNT
(1) 006464 005300      DEC      RO         ;DELAY
(1) 006466 001376      BNE      1$          ;
(1) 006470 012600      MOV      (SP)+,RO  ;RESTORE RO
(1) 006472 000002      RTI                       ;LEAVE ROUTINE
(1) 006474 000001      DLYCNT: .WORD      1 ;PATCHABLE LOC FOR MORE TIME
```

;ADVANCE TO NEXT TEST HANDLER

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-32
 CVDZAD.P11 11-MAR-83 10:06 APT COMMUNICATIONS ROUTINE

```

(1)
(1)
(1) 006476 013716 001362
(1) 006502 005037 001364
(1) 006506 000002
(1)
(1)
(1)
(1)
(1) 006510 106302
(1) 006512 032702 000020
(1) 006516 001402
(1) 006520 022626
(1) 006522 104400
(1) 006524 000002
(1)

```

```

:-----
.ADVANCE:MOV NEXT,(SP) :CRUNCH STACK WITH ADDRESS OF SCOPE CALL
          CLR LOCK :RESET TIGHT LOOP ADDRESS
          RTI :CHECK TO SEE IF OLD TEST GETS REPEATED

:ROUTINE TO SHIFT LINE POINTER
:AND SWITCH TESTS IF NECESSARY
:-----
.SHIFT: ASLB R2 :POINT TO THE NEXT LINE
        BIT #BIT4,R2 :HAVE WE PASSED ALL LINE POINTERS?
        BEQ 1$ :IF NOT, RETURN TO THE TEST
        POP2SP :REMOVE THE TRAP CALL FROM THE STACK
        ADVANCE :GO TO THE NEXT TEST
1$: RTI :RETURN TO THE PRESENT TEST

```

APT COMMUNICATIONS ROUTINE

```

(1)                                     ;LINE PARAMETER REGISTER SETUP ROUTINE
(1)
(1) 006526 010146      .LPRSET:MOV    R1,-(SP)      ;SAVE CONTENTS OF R1
(1) 006530 010246      MOV      R2,-(SP)      ;SAVE CONTENTS OF R2
(1) 006532 013701 001370  MOV      PAR,R1      ;MOVE DEFAULT PARAM. INTO R1
(1) 006536 012702 000001  MOV      #1,R2      ;INIT. FOR LINE 1
(1) 006542 010177 173252 1$:  MOV      R1,@DZVLPR  ;LOAD PARAM. REGISTER
(1) 006546 005201      INC      R1      ;SET R1 FOR NEXT LINE
(1) 006550 106302      ASLB   R2      ;SET R2 FOR NEXT LINE
(1) 006552 032702 000020  BIT      #BIT4,R2  ;ALL LINES DONE?
(1) 006556 001771      BEQ     1$      ;IF NO LOAD NEXT LINE
(1) 006560 012602      MOV      (SP)+,R2  ;RELOAD R2
(1) 006562 012601      MOV      (SP)+,R1  ;RELOAD R1
(1) 006564 000002      RTI      ;RETURN
(1)
(1)                                     ;ROUTINE TO ZERO DATA BUFFER
(1)
(1) 006566 010046      .BUFSET:MOV   R0,-(SP)  ;SAVE CONTENTS OF R0
(1) 006570 012700 001426  MOV      #TDO,R0     ;SET R0 TO TOP OF BUFFER
(1) 006574 005020      1$:  CLR      (R0)+     ;CLEAR BUFFER LOCATION
(1) 006576 022700 001446  CMP      #STOP,R0    ;IS BUFFER ALL CLEARED
(1) 006602 001374      BNE     1$         ;IF NOT CLEAR NEXT LOCATION
(1) 006604 012600      MOV      (SP)+,R0   ;RELOAD R0
(1) 006606 000002      RTI      ;RETURN
(1)
(2)
(3)
(2)                                     ;:*****
(2) .SBTTL  ABORT ROUTINE FOR LCP/ORION UFD MODE
(3) ;:*****
(2)
(2) 006610 000040      UFD=BITS
(2) 006616 032737 000040 000052  ABORT:  BIT      #UFD,52  ;TEST FOR USER FRIENDLY MODE
(2) 006616 001420      BEQ     ABORT2      ;IF NOT UFD THEN CONTINUE NORMAL OPERATION
(2)
(2) 006620 013737 002152 000030  MOV      SAV30,30    ;RESTORE EMT LOCATION (30)
(2) 006626 013737 002154 000032  MOV      SAV32,32    ;RESTORE EMT PRIORITY LOCATION (32)
(2) 006634 104042      EMT      +42        ;GET DCA LOCATION INTO R0 FROM MONITOR
(2) 006636 012760 177777 000042  MOV      #-1,42(R0)  ;SET A -1 INTO LOCATION DRSEFF IN MONITOR
(2) 006644 013700 000042  ABORT1: MOV      42,R0 ;AND PUT THE MONITOR RETURN ADDRESS IN R0
(2) 006650 005037 000042      CLR      42        ;CLEAR MONITOR RETURN FLAG
(2) 006654 000137 004416  ABORT2: JMP      SENDAD ;RETURN TO MONITOR-DO NOT PUSH STACK HERE
(2) 006660 000207      RTS      PC        ;IF NOTUFD RETURN TO MAINLINE
(2)
(3)
(1)                                     ;:*****
(1)
(1)                                     ;ERROR HANDLER
(1) ;-----
(1) 006662 004737 006610      $ERROR: JSR      PC,ABORT ;CHECK IF WE ARE UNDER UFD
(1) 006666 004737 007320      JSR      PC,SERV.G  ;FIND OUT IF <^G> WAS HIT
(1) 006672 032777 010000 172404  BIT      #SW12,@SWR ;BELL ON ERROR?
(1) 006700 001406      BEQ     XBX        ;BR IF NO BELL
(1) 006702 105777 172406      TSTB   @STPS      ;TTY READY.
(1) 006706 100003      BPL     XBX        ;DON'T WAIT IF TTY NOT READY.

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-34
 CVDZAD.P11 11-MAR-83 10:06 ABORT ROUTINE FOR LCP/ORION UFD MODE

(1)	006710	112777	000207	172400		MOVB	#207,@STPB	:PUSH A BELL AT THE TTY.
(1)	006716	032777	020000	172360	XBX:	BIT	#SW13,@SWR	:DELETE ERROR PRINT OUT?
(1)	006724	001113				BNE	HALTS	:BR IF NO PRINT OUT WANTED.
(1)	006726	021637	001262			CMP	(SP),SERRPC	:WAS THIS ERROR FOUND LAST TIME?
(1)	006732	001404				BEQ	1\$:BR IF YES
(1)	006734	011637	001262			MOV	(SP),SERRPC	:RECORD BEING HERE
(1)	006740	105037	001247			CLRB	SERFLG	:PREPARE HEADER
(1)	006744	104407			1\$:	SAV05		:SAVE ALL PROC REGISTERS
(1)	006746	011605				MOV	(SP),R5	:GET THE PC OF ERROR
(1)	006750	162705	000002			SUB	#2,R5	:GET ADDRESS OF TRAP CALL
(1)	006754	011504				MOV	(R5),R4	:GET ERROR INSTRUCTION
(1)	006756	110437	001260			MOVB	R4,\$ITEMB	:COPY TEST NUMBER FOR APT HANDLING
(1)	006762	006304				ASL	R4	:MULT BY TWO
(1)	006764	061504				ADD	(R5),R4	:DOUBLE IT
(1)	006766	006304				ASL	R4	:MULT AGAIN
(1)	006770	042704	177001			BIC	#177001,R4	:CLEAR JUNK
(1)	006774	062704	016302			ADD	#.ERRTAB,R4	:GET POINTER
(1)	007000	012437	007124			MOV	(R4)+,ERRMSG	:GET ERROR MESSAGE
(1)	007004	012437	007136			MOV	(R4)+,DATAHD	:GET DATA HEADRER
(1)	007010	011437	007150			MOV	(R4),DATABP	:GET DATA TABLE
(1)	007014	105737	001247			TSTB	SERFLG	:TYPE HEADER
(1)	007020	001403				BEQ	TYPMSG	:BR IF YES
(1)	007022	005737	007150			TST	DATABP	:DOES DATA TABLE EXIST?
(1)	007026	001044				BNE	TYPDAT	:BR IF YES.
(1)	007030	104402	001357		TYPMSG:	TYPE	,\$CRLF	:TYPE A CARRIAGE RETURN
(1)	007034	104402	001357			TYPE	,\$CRLF	:AND TYPE ANOTHER
(1)	007040	005737	001364			TST	LOCK	
(1)	007044	001402				BEQ	1\$	
(1)	007046	104402	010233			TYPE	,\$MASTEK	
(1)	007052	104402	010221		1\$:	TYPE	,\$MTSTN	
(1)	007056	104412	007312			CNVRT	,\$XTSTN	:SHOW IT
(1)	007062	104402	010313			TYPE	,\$MERRPC	:TYPE PC.
(1)	007066	104412	007304			CNVRT	,\$ERTAB0	:SHOW IT
(1)	007072	104402	010163			TYPE	,\$MCSRX	
(1)	007076	104412	004432			CNVRT	,\$XCSR	
(1)	007102	104402	001357			TYPE	,\$CRLF	:GIVE A CR/LF
(1)	007106	112737	177777	001247		MOVB	#-1,SERFLG	:NO MORE HEADER UNLESS NO DATA TABLE.
(1)	007114	005737	007124			TST	ERRMSG	:IS THERE AN ERROR MESSAGE?
(1)	007120	001402				BEQ	WTBS.FM	:BR IF NO.
(1)	007122	104402				TYPE		:TYPE
(1)	007124	000000			ERRMSG: 0			: ERROR MESSAGE
(1)	007126				WTBS.FM:			
(1)	007126	005737	007136			TST	DATAHD	:DATA HEADER?
(1)	007132	001402				BEQ	TYPDAT	:BR IF NO
(1)	007134	104402				TYPE		:TYPE
(1)	007136	000000			DATAHD: 0			: DATA HEADER
(1)	007140	005737	007150		TYPDAT:	TST	DATABP	:DATA TABLE?
(1)	007144	001402				BEQ	RESREG	:BR IF NO.
(1)	007146	104411				CONVRT		:SHOW
(1)	007150	000000			DATABP: 0			: DATA TABLE
(1)	007152	104410			RESREG:	RES05		:RESTORE PROC REGISTERS
(1)	007154	122737	000001	001140	HALTS:	CMPB	#APTENV,\$ENV	:IS APT RUNNING?
(1)	007162	001007				BNE	15\$:SKIP APT CALL IF NOT
(1)	007164	113737	001260	007176		MOVB	\$ITEMB,5\$:COPY ERROR NUMBER
(1)	007172	004737	005352			JSR	PC,\$ATV4	:CALL APT SERVICE
(1)	007176	000000			5\$:	.WORD	0	:ERROR NUMBER STUCK HERE

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-35
 CVDZAD.P11 11-MAR-83 10:06 ABORT ROUTINE FOR LCP/ORION UFD MODE

```

(1) 007200 000777          10$: BR      10$      ;LOCK UP HERE
(1) 007202 022737 004416 000042 15$: CMP     #SENDAD,#42 ;CHECK TO SEE IF IN ACT-11 MODE
(1) 007210 001403          BEQ     20$      ;IF SO, HANDLE ACCORDINGLY
(1) 007212 005777 172066     TST     @SWR     ;HALT ON ERROR?
(1) 007216 100006          BPL     EXITER   ;BR IF NO HALT ON ERROR
(1) 007220 016677 000002 172060 20$: MOV     2(SP),@DISPLAY ;SHOW ERROR PC IN DATA DISPLAY
(1) 007226 004737 006610     JSR     PC,ABORT ;ARE WE UNDER UFD ?
(1) 007232 000000          HALT                    ;HALT
(1) 007234 005237 001256     EXITER: INC     SERTTL ;UPDATE ERROR COUNT
(1) 007240 004737 007320     JSR     PC,SERV.G ;FIND OUT IF ^G WAS TYPED
(1) 007244 032777 000400 172032  BIT     #SW08,@SWR ;GOTO TOP OF TEST?
(1) 007252 001007          BNE     1$        ;BR IF YES
(1) 007254 032777 002000 172022  BIT     #SW10,@SWR ;GOTO NEXT TEST?
(1) 007262 001407          BEQ     2$        ;BR IF NO
(1) 007264 013737 001362 001252  MOV     NEXT,$LPADR ;SET FOR NEXT TEST
(1) 007272 012706 001120     1$:  MOV     #STACK,SP ;RESET SP
(1) 007276 000177 171750     JMP     @SLPADR   ;GOTO SPECIFIED TEST
(1) 007302 000002     2$:  RTI                    ;RETURN
(1) 007304 000001     ERTAB0: 1
(1) 007306 006 002          .BYTE 6,2
(1) 007310 001404          SAVPC
(1) 007312 000001     XTSTN: 1
(1) 007314 002 002          .BYTE 2,2
(1) 007316 001246          $TSTNM
(1) 007320 017746 171766     SERV.G: MOV     @STKB,-(SP) ;OTHERWISE, GET THE LAST CHARACTER TYPED
(1) 007324 042716 000200     BIC     #BIT7,(SP) ;STRIP PARITY(EIGHTH) BIT
(1) 007330 122726 000007     CMPB   #7,(SP)+ ;IS IT ^G?
(1) 007334 001076          BNE     6$        ;IF NOT, IGNORE INPUT
(1) 007336 032777 004000 171744  BIT     #4000,@STKS ;RX BUSY?
(1) 007344 001365          BNE     SERV.G   ;BR IF YES
(1) 007346 007346          GETSWR= . ;:GPA
(1) 007346 017737 171732 007554  MOV     @SWR,90$ ;SAVE (SWR).
(1) 007354 104402 007534     1$:  TYPE   .89$ ;TYPE HEADER FOR OLD SWITCH REGISTER
(1) 007360 104412 007546     CNVRT  .88$ ;TYPE THE NUMBER ITSELF
(1) 007364 104402 007556     TYPE   .91$ ;AFTER HAVING CONVERTED IT TO ASCII
(1) 007370 105037 007562     CLRB   92$ ;CLEAR SWR CHANGE FLAG
(1) 007374 005077 171704     CLR    @SWR ;CLEAR THE SOFTWARE SWITCH REGISTER
(1) 007400 105777 171704     3$:  TSTB  @STKS ;WAIT FOR DONE.
(1) 007404 100375          BPL     3$        ;CONTINUE WAITING FOR IT
(1) 007406 017746 171700     MOV     @STKB,-(SP) ;PUT THE CHARACTER ON THE STACK
(1) 007412 042716 000200     BIC     #BIT7,(SP) ;STRIP PARITY BIT
(1) 007416 122726 000015     CMPB   #15,(SP)+ ;IS IT THE CARRIAGE RETURN CHAR?
(1) 007422 001433          BEQ     4$        ;IF SO, GO PRINT CRLF
(1) 007424 105777 171664     2$:  TSTB  @STPS ;IS THE OUTPUT BUFFER AVAILABLE
(1) 007430 100375          BPL     2$        ;IF NOT, WAIT FOR IT TO BE READY
(1) 007432 105237 007562     INCB   92$ ;INDICATE THAT THE SWR WAS CHANGED
(1) 007436 014677 171654     MOV     -(SP),@STPB ;PLACE THE CHARACTER THERE(ECHO BACK)
(1) 007442 000241          CLC                    ;GET READY TO ROTATE
(1) 007444 006177 171634     ROL    @SWR ;MOVE THE EXISTING BITS OVER
(1) 007450 006177 171630     ROL    @SWR ;TO MAKE ROOM FOR THE INCOMING
(1) 007454 006177 171624     ROL    @SWR ;THREE BITS FROM THIS CHARACTER
(1) 007460 103735          BCS     1$        ;ERROR
(1) 007462 022627 000060     CMP    (SP)+,#60 ;IS IT LOWER THAN 0?
(1) 007466 002732          BLT    1$        ;IF SO, GO ASK AGAIN
(1) 007470 026627 177776 000067  CMP    -2(SP),#67 ;IS IT HIGHER THAN 7?
(1) 007476 003326          BGT    1$        ;IF SO, GO ASK AGAIN
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-36
CVDZAD.P11 11-MAR-83 10:06

ABORT ROUTINE FOR LCP/ORION UFD MODE

```

(1) 007500 042746 177770      BIC    #^C<7>,-(SP)    ;ISOLATE INFORMATION BITS
(1) 007504 052677 171574      BIS    (SP)+,@SWR      ;ADD THEM TO THE SWITCH REGISTER
(1) 007510 000733              BR     3$              ;GO CHECK FOR THE NEXT CHARACTER
(1) 007512 105737 007562      4$:   TSTB   9? $      ;HAS THE SWR BEEN CHANGED?
(1) 007516 001003              BNE    5$              ;IF YES GO TYPE CRLF
(1) 007520 013777 007554 171556  MOV    90$,@SWR        ;IF NOT RESTORE SWR
(1) 007526 104402 001357      5$:   TYPE   , $CRLF    ;TYPE A CARRIAGE RETURN AND LINE FEED
(1) 007532 000207              6$:   RTS     PC        ;RETURN TO CALLING PROCEDURE
(1)

```

```

(1) 007534 020200 051450 051127 89$:   .ASCIZ  <200>? (SWR)=/?
(1) 007542 036451 000057
(1)

```

```

(1) 007546 000001      .EVEN
(1) 007550      006      000      88$:   1
(1) 007552 007554      .BYTE   6,0
(1) 007554 000000      90$:   .WORD   0
(1) 007556 036457 000057      91$:   .ASCIZ  ?/=/?
(1) 007562      000      92$:   .BYTE   0
(1)

```

.SBTTL POWER DOWN AND UP ROUTINES

```

(2)
(2)
(2) 007564 012737 007730 000024 :POWER DOWN ROUTINE
(2) 007572 012737 000340 000026 $PWRDN: MOV    # $ILLUP,@#PWRVEC ;;SET FOR FAST UP
(4) 007600 010046              MOV    #340,@#PWRVEC+2 ;;PRIO:7
(4) 007602 010146              MOV    R0,-(SP)        ;;PUSH R0 ON STACK
(4) 007604 010246              MOV    R1,-(SP)        ;;PUSH R1 ON STACK
(4) 007606 010346              MOV    R2,-(SP)        ;;PUSH R2 ON STACK
(4) 007610 010446              MOV    R3,-(SP)        ;;PUSH R3 ON STACK
(4) 007612 010546              MOV    R4,-(SP)        ;;PUSH R4 ON STACK
(4) 007614 010746 171464      MOV    R5,-(SP)        ;;PUSH R5 ON STACK
(2) 007620 010637 007734      MOV    @SWR,-(SP)      ;;PUSH @SWR ON STACK
(2) 007624 012737 007636 000024  MOV    SP,$SAVR6      ;;SAVE SP
(2) 007632 000000              MOV    # $PWRUP,@#PWRVEC ;;SET UP VECTOR
(2) 007634 000776              HALT
(2)                               BR     -2              ;;HANG UP
(2)

```

```

(2)
(2) 007636 012737 007730 000024 :POWER UP ROUTINE
(2) 007644 013706 007734      $PWRUP: MOV    # $ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
(2) 007650 005037 007734      MOV    $SAVR6,SP      ;;GET SP
(2) 007654 005237 007734      CLR    $SAVR6         ;;WAIT LOOP FOR THE TTY
(2) 007660 001375              1$:   INC    $SAVR6       ;;WAIT FOR THE INC
(4) 007662 012677 171416      BNE    1$             ;;OF WORD
(4) 007666 012605              MOV    (SP)+,@SWR     ;;POP STACK INTO @SWR
(4) 007670 012604              MOV    (SP)+,R5       ;;POP STACK INTO R5
(4) 007672 012603              MOV    (SP)+,R4       ;;POP STACK INTO R4
(4) 007674 012602              MOV    (SP)+,R3       ;;POP STACK INTO R3
(4) 007676 012601              MOV    (SP)+,R2       ;;POP STACK INTO R2
(4) 007700 012600              MOV    (SP)+,R1       ;;POP STACK INTO R1
(2) 007702 012737 007564 000024  MOV    (SP)+,R0       ;;POP STACK INTO R0
(2) 007710 012737 000340 000026  MOV    # $PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
(2) 007716 104402              MOV    #340,@#PWRVEC+2 ;;PRIO:7
(2) 007720 007736      SPWRMG: TYPE          ;;REPORT THE POWER FAILURE
(2) 007722 012716      MOV    .WORD   MPFAIL  ;;POWER FAIL MESSAGE POINTER
(2)                               MOV    (PC)+,(SP)     ;;RESTART AT RESTART

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-37
CVDZAD.P11 11-MAR-83 10:06

POWER DOWN AND UP ROUTINES

```

(2) 007724 011266      SPWRAD: .WORD  RESTART      ;;RESTART ADDRESS
(2) 007726 000002      RTI
(2) 007730 000000      $ILLUP: HALT      ;;THE POWER UP SEQUENCE WAS STARTED
(2) 007732 000776      BR      .-2      ;; BEFORE THE POWER DOWN WAS COMPLETE
(2) 007734 000000      $SAVR6: 0      ;;PUT THE SP HERE
(2) 007736 050200 051127 043040 MPFAIL: .ASCIZ <200>/PWR FAILED. RESTART AT LAST TEST /
(2) 010001 200 047105 020104 MEPASS: .ASCIZ <200>/END PASS CVDZA-D /
(2) 010025 200 052522 047116 MR: .ASCIZ <200>/RUNNING /
(2) 010041 200 051120 043517 MERR2: .ASCIZ <200>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 010110 044600 051516 043125 MERR3: .ASCIZ <200>/INSUFFICIENT DATA!/
(2) 010134 046200 041517 020113 MLOCK: .ASCIZ <200>/LOCK ON SELECTED TEST/
(2) 010163 103 051123 020072 MCSRX: .ASCIZ /CSR: /
(2) 010171 126 041505 020072 MVECX: .ASCIZ /VEC: /
(2) 010177 120 051501 042523 MPASSX: .ASCIZ /PASSES: /
(2) 010210 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /
(2) 010221 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 010233 052 000040 MASTEK: .ASCIZ /* /
(2) 010236 052200 050131 020105 MNEW: .ASCIZ <200>/TYPE A BIT MAP OF DZV11'S DESIRED ACTIVE: /
(2) 010313 120 035103 000040 MERRPC: .ASCIZ /PC: /
(2) 010320 046600 050101 047440 XHEAD: .ASCIZ <200>/MAP OF DZV11 STATUS/<200>
(2) 010346 044600 046114 043505 MBADLN: .ASCIZ <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
(2) 010410 000002      .EVEN
(2) 010412 006 003      XSTATQ: 2
(2) 010414 001344      .BYTE 6.3
(2) 010416 006 002      $TMP1
(2) 010420 001346      .BYTE 6.2
(1)                      $TMP2
                          .EVEN

```

```

(2)                                     :THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
(2)                                     :-----
(2)                                     :E=EXTERNAL LOOP BACK
(2)                                     :I=INTERNAL LOOP BACK
(2)                                     :S=STAGGERED LOOP BACK
(2) 010422 017605 000000                .SETFLG:MOV @ (SP),R5      :PICK UP ADDRESS OF TAG
(2) 010426 042737 000040 010616        BIC #0,INBUF      :STRIP LOWER CASE
(2) 010434 122737 000105 010616        CMPB #'E,INBUF   :IS IT EXTERNAL LOOP BACK ?
(2) 010442 001005                    BNE 4$           :NO
(2) 010444 013715 010534                MOV 1$, (R5)     :YES STORE INFO
(2) 010450 105037 001424                CLRB MNTFLG     :SET MAINT BIT =0
(2) 010454 000422                    BR 7$           :GET OUT
(2) 010456 122737 000111 010616 4$:    CMPB #'I,INBUF   :IS IT INTERNAL LOOP BACK ?
(2) 010464 001006                    BNE 5$           :NO
(2) 010466 013715 010536                MOV 2$, (R5)     :YES STORE INFO
(2) 010472 112737 000010 001424        MOVB #MAINT,MNTFLG :SET UP THE MAINTENANCE FLAG LOADER
(2) 010500 000410                    BR 7$           :GET OUT
(2) 010502 122737 000123 010616 5$:    CMPB #'S,INBUF   :IS IT STAGGERED LOOP BACK ?
(2) 010510 001007                    BNE 6$           :WHAT ?
(2) 010512 013715 010540                MOV 3$, (R5)     :YES STORE INFO
(2) 010516 105037 001424                CLRB MNTFLG     :ZERO BITS
(2) 010522 062716 000002                ADD #2, (SP)     :POP AROUND
(2) 010526 000002                    RTI
(2) 010530 104404                6$: INSTER      :RETRY
(2) 010532 000733                    BR .SETFLG      :DITTO
(2) 010534 000200                1$: .WORD 200   :EXTERNAL = E
(2) 010536 000000                2$: .WORD 0     :INTERNAL = I
(2) 010540 100000                3$: .WORD 100000 :STAGGERED = S

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-39
 CVDZAD.P11 11-MAR-83 10:06 POWER DOWN AND UP ROUTINES

```

(2)                                     :COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
(2)                                     :BUFFER TO THE CHARACTERS 'E' AND 'C'.
(2)                                     :IF THE CHARACTER IS 'E' CLEAR THE FLAG
(2)                                     :IF THE CHARACTER IS 'C' SET THE FLAG
(2)
(2) 010542 017605 000000                .PAWCH:MOV      @ (SP),R5
(2) 010546 142737 000040 010616         BICB   #40,INBUF      :SET FOR LOWER CASE INPUT
(2) 010554 122737 000105 010616         CMPB   #'E,INBUF     :IS IT 'E' ?
(2) 010562 001002                        BNE    1$
(2) 010564 105015                        CLRB   (R5)          :000
(2) 010566 000406                        BR     2$
(2) 010570 122737 000103 010616 1$:     CMPB   #'C,INBUF     :IS IT 'C' ?
(2) 010576 001005                        BNE    3$
(2) 010600 112715 177777                MOVB   #-1,(R5)      :3177
(2) 010604 062716 000002                2$:     ADD    #2,(SP)
(2) 010610 000002                        RTI
(2) 010612 104404                        3$:     INSTER          :RETRY
(2) 010614 000752                        BR     .PAWCH
(2)
(2)                                     :BUFFERS FOR INPUT-OUTPUT
(2)
(2) 010616 000000                INBUF: 0
(2) 010660 010660                .=.+40
(2)                                     : TEMP: 0           : TEMP AREA UNUSED.           ::GPA
(2)                                     : .=.+40           : DELETED TO CONSERVE SPACE  ::GPA
(2) 010660 000000                MDATA: 0
(2) 010722 010722                .=.+40
(2)

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-41
 CVDZAD.P11 11-MAR-83 10:06 POWER DOWN AND UP ROUTINES

```

(2) 011170 012700 012272      MOV      #TST1,RO
(2) 011174 022710 000004      5$:     CMP      #4,(RO)
(2) 011200 001020              BNE      6$
(2) 011202 022760 012737 000002  CMP      #12737,2(RO)
(2) 011210 001014              BNE      6$
(2) 011212 023760 001246 000004  CMP      $TSTNM,4(RO)      ;IS THIS THE TEST ?
(2) 011220 001010              BNE      6$                ;IF NOT, DON'T PROCESS NUMBER
(2) 011222 010037 001252      MOV      RO,$LPADR          ;SAVE PC
(2) 011226 062737 000002 001252  ADD      #2,$LPADR          ;POP OVER PREVIOUS SCOPE
(2) 011234 104402 001357      TYPE     ,$CRLF
(2) 011240 000412              BR       8$
(2) 011242 005720              6$:     TST      (RO)+
(2) 011244 020027 015750      CMP      RO,#TLAST+10
(2) 011250 001351              BNE      5$
(2) 011252 104402 001356      TYPE     ,SQUES
(2) 011256 000733              BR       4$
(2) 011260 012737 012272 001252  7$:     MOV      #TST1,$LPADR      ;PREPARE TEST ADDRESS
(2) 011266              8$:
(2) 011266 000177 167760      RESTART:JMP    @,$LPADR      ;GO START TESTING.***WARNING!***
(2)                                  ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!
(2)                                  ;THIS UTILITY SETS UP CSR'S,SETS UP VECTORS.
(2) 011272 013700 002040      DZVLEV: MOV    DZVRIV,RO      ;PLACE THE BASE VECTOR ADDRESS IN RO
(2) 011276 062700 000002      ADD      #2,RO              ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
(2) 011302 010037 002042      MOV      RO,DZVRIS          ;STORE IT HERE
(2) 011306 062700 000002      ADD      #2,RO              ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
(2) 011312 010037 002044      MOV      RO,DZVTIV          ;STORE IT HERE
(2) 011316 062700 000002      ADD      #2,RO              ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
(2) 011322 010037 002046      MOV      RO,DZVTIS          ;STORE IT HERE
(2)                                  ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZV11. $BASE IS THE BASE ADDRESS
(2)                                  ;OF THE DEVICE
(2) 011326 013700 001174      MOV      $BASE,RO           ;COPY THE ADDRESS BEING LOADED
(2) 011332 010037 002010      MOV      RO,DZVCSR          ;XXX0
(2) 011336 005200              INC      RO
(2) 011340 010037 002012      MOV      RO,HDZVCSR         ;XXX1
(2) 011344 005200              INC      RO
(2) 011346 010037 002014      MOV      RO,DZVRBUF         ;XXX2
(2) 011352 010037 002020      MOV      RO,DZVLPR          ;XXX2
(2) 011356 005200              INC      RO
(2) 011360 010037 002016      MOV      RO,HDZVRBUF        ;XXX3
(2) 011364 010037 002022      MOV      RO,HDZVLPR         ;XXX3
(2) 011370 005200              INC      RO
(2) 011372 010037 002024      MOV      RO,DZVTCR          ;XXX4
(2) 011376 005200              INC      RO
(2) 011400 010037 002026      MOV      RO,HDZVTCR         ;XXX5
(2) 011404 005200              INC      RO
(2) 011406 010037 002030      MOV      RO,DZVMSR          ;XXX6
(2) 011412 010037 002034      MOV      RO,DZVTDR          ;XXX6
(2) 011416 005200              INC      RO
(2) 011420 010037 002032      MOV      RO,HDZVMSR         ;XXX7
(2) 011424 010037 002036      MOV      RO,HDZVTDR         ;XXX7
(2) 011430 000207              RTS      PC

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-43
CVDZAD.P11 11-MAR-83 10:06 POWER DOWN AND UP ROUTINES

(2)

; END OF .PARMD DELETE RANGE

&

::GPA

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-44
 CVDZAD.P11 11-MAR-83 10:06 POWER DOWN AND UP ROUTINES

```

(2)                                     ;*ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
(2)                                     ;*IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
(2)                                     ;*THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.
(2) 011434 012700 001500                SETAPT: MOV    #DZV.MAP,R0        ;POINT TO THE DEVICE MAP TABLE
(2) 011440 013701 001174                MOV    $BASE,R1          ;BUILD DEVICE ADDRESSES IN R1
(2) 011444 013702 001170                MOV    $VECT1,R2        ;BUILD DEVICE VECTORS IN R2
(2) 011450 042702 177007                BIC    #^C<770>,R2      ;STRIP AWAY OTHER INFORMATION
(2)                                     ;
(2) 011454 012704 001204                MOV    #SDDW0,R4        ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
(2) 011460 013705 001176                MOV    $DEVN,R5        ;GET THE MAP OF ACTIVE DEVICES
(2) 011464 105037 001414                CLRB  DZVNUM           ;INITIALIZE NO. OF DEVICES IN SYSTEM
(2) 011470 005037 001410                CLR   SAVACTV          ;CLEAR THE ACTIVE BIT MAP
(2) 011474 006005                        1$:  ROR    R5           ;GET A DEVICE SELECTION BIT
(2) 011476 103407                        BCS   3$              ;IF IT IS SELECTED, GO SET UP A MAP
(2) 011500 001422                        BEQ   5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
(2) 011502 005724                        TST   (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
(2) 011504 062701 000010                2$:  ADD   #10,R1       ;SET UP THE NEXT ADDRESS
(2) 011510 062702 000010                ADD   #10,R2          ;SET UP THE NEXT VECTOR GROUP
(2) 011514 000767                        BR    1$              ;GO SEE IF MORE DEVICES REMAIN
(2) 011516 006137 001410                3$:  ROL   SAVACTV     ;SET BIT IN ACTIVE DEVICE MAP
(2) 011522 105237 001414                INCB  DZVNUM          ;INCREMENT NO. OF ACTIVE DEVICES IN SYSTEM
(2) 011526 010120                MOV   R1,(R0)+        ;LOAD DEVICE ADDRESS
(2) 011530 010220                MOV   R2,(R0)+        ;LOAD THE VECTOR ADDRESS
(2) 011532 013720 001200                MOV   $CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
(2) 011536 012420                MOV   (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
(2) 011540 013720 001202                MOV   $CDW2,(R0)+     ;LOAD DEFAULT TESTING MODE
(2) 011544 000757                        BR    2$              ;GO BUILD THE NEXT ADDRESS
(2) 011546 012710 177777                5$:  MOV   #-1,(R0)    ;TERMINATE THE DEVICE MAP
(2) 011552 012737 001142 001304        MOV   #$$SWREG,SWR    ;SET TO SOFTWARE APT SWITCH REGISTER
(2) 011560 000207                        RTS   PC              ;RETURN TO PRINT STATUS TABLE
    
```

```

(2)                                     ;*ROUTINE USED TO "AUTO SIZE" THE DZV11
(2)                                     ;*CSR AND VECTOR.
(2)                                     ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
(2)                                     ;* ADDRESS RANGE (160000:167770)
(2)                                     ;* AND THE VECTOR MAY BE ANY WHERE IN THE
(2)                                     ;* FLOATING VECTOR RANGE (300:770)
(2)                                     ;*
(2) 011562                                AUTO.SIZE:
(2) 011562 000005                        RESET
(2) 011564 105337 001422                CSRMAP: DECB   INIFLG        ;INSURE A BUS INIT.
(2) 011570 012702 001500                MOV    #DZV.MAP,R2     ;SHOW THAT I WAS HERE
(2) 011574 012703 001204                MOV    #SDDW0,R3      ;LOAD MAP POINTER.
(2) 011600 005022                        1$:  CLR   (R2)+        ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
(2) 011602 022702 001740                CMP    #DZV.END,R2   ;ZERO ENTIRE MAP
(2) 011606 001374                        BNE   1$              ;ALL DONE?
(2) 011610 105037 001414                CLRB  DZVNUM          ;BR IF NO
(2) 011614 012702 001500                MOV    #DZV.MAP,R2   ;SET OCTAL NUMBER OF DZV11'S TO 0
(2) 011620 012701 160000                MOV    #160000,R1    ;SET FOR FIRST ADDRESS TO BE TESTED
(2) 011624 012737 012074 000004        MOV    #6$,@#4       ;SET FOR NON-EXISTENT DEVICE TIME OUT
(2) 011632 052711 000040                2$:  BIS   #BIT5,(R1)  ;TRY TO SET MASTER SCAN ENABLE
(2) 011636 052761 000017 000004        BIS   #17,4(R1)     ;TRY TO TRANSMIT ON ANY LINE
(2) 011644 005000                        CLR   R0              ;USE R0 AS A COUNTER
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-45
 CVDZAD.P11 11-MAR-83 10:06 POWER DOWN AND UP ROUTINES

```

(2) 011646 005711          7$:   TST      (R1)          ;HAS TRANSMITTER READY COME UP?
(2) 011650 100403          BMI      8$          ;IF SO, GO GET A FINAL CHECK
(2) 011652 005300          DEC      R0          ;REDUCE COUNT. TIME UP?
(2) 011654 001374          BNE     7$          ;IF NOT, KEEP WAITING
(2) 011656 000437          BR      3$          ;ASSUME IT'S NOT A DZV11
(2) 011660 032761 000017 000004 8$:   BIT      #17,4(R1)   ;ARE ANY TCR BITS STILL SET? THEY SHOULD BE
(2) 011666 001433          BEQ     3$          ;IF IT'S NOT, ASSUME IT'S NOT A DZV11
(2) 011670 032711 000040          BIT      #BIT5,(R1) ;IS MASTER SCAN ENABLE STILL SET?
(2) 011674 001430          BEQ     3$          ;IF NOT, ASSUME IT'S NOT A DZV11
(2) 011676 052711 000020          BIS     #20,(R1)   ;SET DEVICE CLEAR
(2) 011702 000240          NOP
(2) 011704 032711 000040          BIT     #40,(R1)   ;DID SCANNER CLEAR
(2) 011710 001022          BNE     3$          ;IF NOT ASSUME IT IS NOT DZV
(2) 011712 005061 000004          CLR     4(R1)      ;GET RID OF TCR BITS
(2)                                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZV11 CSR ADDRESS.
(2) 011716 010122          MOV     R1,(R2)+   ;STORE CSR IN CORE TABLE.
(2) 011720 005722          TST     (R2)+      ;POP OVER VECTOR STORE AREA
(2) 011722 012722 000017          MOV     #17,(R2)+ ;SET THE DEFAULT LINE SELECTION PARAMETER
(2) 011726 012712 017470          MOV     #17470,(R2);SET THE DEFAULT PARAMETERS
(2) 011732 012223          MOV     (R2)+,(R3)+;COPY PARAMETERS INTO ETABLE DESCRIPTOR
(2) 011734 005022          CLR     (R2)+      ;SET THE DEFAULT MODE OF OPERATION
(2) 011736 012712 177777          MOV     #-1,(R2)  ;TERMINATE LIST
(2) 011742 105237 001414          INCB   DZVNUM     ;UPDATE DEVICE COUNTER
(2) 011746 122737 000020 001414          CMPB   #20,DZVNUM ;ARE MAX. NO. OF DEV FOUND?
(2) 011754 001405          BEQ     100$      ;YES DON'T LOOK FOR ANY MORE.
(2) 011756 062701 000010          3$:   ADD     #10,R1    ;UPDATE CSR POINTER ADDRESS
(2) 011762 022701 164000          CMP     #164000,R1
(2) 011766 001321          BNE     2$          ;BR IF MORE ADDRESS TO CHECK.
(2) 011770          100$:
(2) 011770 105737 001414          TSTB   DZVNUM     ;WERE ANY DZV11'S FOUND AT ALL?
(2) 011774 001430          BEQ     5$          ;ERROR AUTO SIZER FOUND NO DZV11'S IN THIS SYS.
(2) 011776 113701 001414          MOVB   DZVNUM,R1
(2) 012002 012737 000001 001410          MOV     #1,SAVACTV ;CREATE A BIT MAP OF THE ACTIVE
(2) 012010 005301          4$:   DEC     R1          ;DEVICES IN THE SYSTEM
(2) 012012 001404          BEQ     98$
(2) 012014 000261          SEC
(2) 012016 006137 001410          ROL     SAVACTV
(2) 012022 000772          BR      4$
(2) 012024 013737 001500 001174 98$:   MOV     DZCRO,$BASE ;POINT TO THE ADDRESS OF FIRST DEVICE
(2) 012032 013737 001510 001202          MOV     MANTO,$CDW2 ;INDICATE TO ETABLE WHAT MODE IS BEING USED
(2) 012040 012737 000006 000004 99$:   MOV     #6,#4      ;RESTORE TRAP VECTOR
(2) 012046 013737 001410 001176          MOV     SAVACTV,$DEV ;SAVE ACTIVE REGISTER
(2) 012054 000412          BR      VECMAP    ;GO FIND THE VECTOR NOW.
(2) 012056 104402 010041          5$:   TYPE   ,MERR2    ;NOTIFY OPR THAT NO DZV11'S FOUND.
(2) 012062 005000          CLR     R0          ;MAKE DATA DISPLAY ZERO
(2) 012064 004737 006610          JSR    PC,ABORT   ;ARE WE UNDER UFD ?
(2) 012070 000000          HALT
(2) 012072 000776          BR      -2
(2) 012074 012716 011756          6$:   MOV     #-2
(2) 012100 000002          RTI    #3$,(SP)   ;ENTERED BY NON-EXISTENT TIME-OUT
(2)                                ;RETURN TO MAINSTREAM
(2) 012102 012737 000200 000022 VECMAP: MOV     #MASK,#22   ;SET IOT TRAP PRIORITY
(2) 012110 012737 012224 000020          MOV     #4$,#20   ;SET IOT TRAP VECTOR
(2) 012116 012702 001500          MOV     #DZV.MAP,R2 ;SET SOFTWARE POINTER
(2) 012122 012700 000300          MOV     #300,R0   ;FLOATING VECTORS START HERE.
(2) 012126 012701 000302          MOV     #302,R1   ;PC OF IOT INSTR.

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-46
 CVDZAD.P11 11-MAR-83 10:06 POWER DOWN AND UP ROUTINES

(2)	012132	010120			1\$:	MOV	R1,(R0)+	;START FILLING VECTOR AREA
(2)	012134	012721	000004			MOV	#4,(R1)+	;WITH .+2; IOT
(2)	012140	022021				CMP	(R0)+,(R1)+	;ADD 2 TO R0 +R1
(2)	012142	020127	001000			CMP	R1.#1000	;HAS THE VECTOR AREA BEEN EXCEEDED?
(2)	012146	101771				BLOS	1\$;BR IF MORE TO FILL
(2)	012150	013704	001410			MOV	SAVACTV,R4	;STORE TEMPORARILY
(2)	012154	006004			2\$:	ROR	R4	;BRING OUT A BIT
(2)	012156	103036				BCC	5\$;BR IF ALL DONE
(2)	012160	106427	000000			MTPS	#0	;ZERO CPU PRIO
(2)	012164	012772	040040	000000		MOV	#BIT14+BITS5,a(R2)	;SET TIE AND MAS SCAN
(2)	012172	011201				MOV	(R2),R1	;GET CSR
(2)	012174	112761	000017	000004		MOVB	#17,4(R1)	;SET THE TCR BITS FOR ALL LINES
(2)								;ATTEMPT TO FORCE AN INTERRUPT
(2)	012202	005200				INC	R0	;STALL
(2)	012204	001376				BNE	.-2	; FOR TIME TO INTERRUPT
(2)	012206	012762	000300	000002		MOV	#300,2(R2)	;NO INTERRUPT ASSUME 300 AND FIX DZV11 LATER
(2)	012214	000005				RESET		;INIT
(2)	012216	062702	000012		3\$:	ADD	#12,R2	;POP SOFTWARE POINTER
(2)	012222	000754				BR	2\$;KEEP GOING
(2)	012224	011662	000002		4\$:	MOV	(SP),2(R2)	;GET VECTOR ADDRESS
(2)	012230	162762	000010	000002		SUB	#10,2(R2)	;POINT BACK TO THE CORRECT VECTOR
(2)	012236	042762	000007	000002		BIC	#7,2(R2)	;CLEAR JUNK
(2)	012244	022626				POP2SP		;POP IOT JUNK OFF STACK
(2)	012246	012716	012216			MOV	#3\$, (SP)	;SET FOR RETURN
(2)	012252	000002				RTI		
(2)	012254	013737	001502	001170	5\$:	MOV	DZVCO,\$VECT1	;COPY VECTOR OF FIRST DEVICE INTO ETABLE
(2)	012262	012737	004462	000020		MOV	#.SCOPE,IOTVEC	;RESTORE THE SCOPE TRAP
(2)	012270	000207				RTS	PC	;ALL DONE WITH "AUTO SIZING"

8735
8736

```

(1)
(1)
(1)
(3)
(6)
(5) 012272 000004
(3) 012274 012737 000001 001246
(3) 012302 012737 012462 001362
(1) 012310 012737 012450 000004
(1) 012316 012737 000200 000006
(1) 012324 012737 012332 001364
(1) 012332 013700 002010
(1) 012336 011001
(1) 012340 000240
(1) 012342 005010
(1) 012344 000240
(1) 012346 012737 012354 001364
(1) 012354 013700 002014
(1) 012360 011001
(1) 012362 000240
(1) 012364 005010
(1) 012366 000240
(1) 012370 012737 012376 001364
(1) 012376 013700 002024
(1) 012402 011001
(1) 012404 000240
(1) 012406 005010
(1) 012410 000240
(1) 012412 012737 012420 001364
(1) 012420 013700 002030
(1) 012424 011001
(1) 012426 000240
(1) 012430 005010
(1) 012432 000240
(1) 012434 012737 000006 000004
(1) 012442 005037 000006
(1) 012446 104400
(1) 012450 011601
(1) 012452 022626
(1) 012454 104001
(1) 012456 104401
(1) 012460 000111

```

```

***** TEST 1 *****
*THIS TEST PROVES THE BUS REPLY RESPONSE
*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
*
DZVLSR, DZVRBUF, DZVTCR, DZVMSR
::* TEST 1
*****
TST1: SCOPE
MOV #1,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST2,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV #5$,4 ;SET TRAP VECTOR
MOV #MASK,6 ;SET PRIORITY TO HIGH(MASK INTERRUPTS)
MOV #1$,LOCK ;SET RETURN IF SW09=11
1$: MOV DZVCSR,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ THE ADDRESS
NOP ;WASTE TIME
CLR (RO) ;WRITE THE ADDRESS
NOP ;WASTE TIME
MOV #2$,LOCK ;SET RETURN ADDRESS FOR SW09
2$: MOV DZVRBUF,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ THE ADDRESS
NOP ;
CLR (RO) ;WRITE THE ADDRESS
NOP ;WASTE TIME
MOV #3$,LOCK ;SET RETURN ADDRESS FOR SW09
3$: MOV DZVTCR,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ THE ADDRESS
NOP ;
CLR (RO) ;WRITE THE ADDRESS
NOP ;
MOV #4$,LOCK ;SET RETURN ADDRESS
4$: MOV DZVMSR,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ FROM ADDRESS
NOP ;
CLR (RO) ;WRITE THE ADDRESS
NOP ;
MOV #6,4 ;SET TRAP CATCHER BACK TO NORMAL
5$: ADVANCE ;SCOPE THIS TEST
MOV (SP),R1 ;SAVE PC OF TRAP
POP2SP ;POP TRAP OFF STACK
ERROR+ 1 ;*NO BUS REPLY RESPONSE.
SCOP1 ;SW09=1?
JMP (R1) ;RTI

```

8737
8738
8739
8740
8742

```

(5)
(4) 012462 000004
(2) 012464 012737 000002 001246
(2) 012472 012737 012526 001362
8743 012500 013700 002010
8744 012504 012710 000020
8745 012510 005005

```

```

***** TEST 2 *****
*THIS TEST PROVES THAT BIT "DCLR"
*CAN BE SET AND THAT IT WILL CLEAR
*BY ITSELF
::* TEST 2
*****
TST2: SCOPE
MOV #2,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST3,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZVCSR,RO ;SET POINTER
MOV #DCLR,(RO) ;SET DCLR
CLR R5 ;SET EXPECTED TO 0

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-48
CVDZAD.P11 11-MAR-83 10:06 DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

8746 012512 005003          CLR      R3          ;DUAL LOOP COUNTER
8747 012514 011004          2$: MOV     (R0),R4    ;IS DCLR CLEAR?
8748 012516 001403          BEQ     3$          ;IF YES , GO TO THE NEXT TEST
8749 012520 105203          INCB   R3          ;IF NO,COUNT 1 OF 256 TICKS
8750 012522 001374          BNE    2$          ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
8751 012524 104002          ERROR+ 2          ;*DCLR FAILED TO CLEAR
8752 012526
8753
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(3)
(6)
(5) 012526 000004          TST3: SCOPE
(3) 012530 012737 000003 001246 MOV     #3,$TSTNM    ;LOAD THE NUMBER OF THIS TEST
(3) 012536 012737 012704 001362 MOV     #TST4,NEXT   ;POINT TO THE START OF THE NEXT TEST
(1) 012544 013700 002010          MOV     DZVCSR,R0   ;GET BASE ADDRESS
(1) 012550 012703 012664          MOV     #5$,R3      ;SET R3 TO TOP OF TABLE
(1) 012554 011305          1$: MOV     (R3),R5    ;SET BIT
(1) 012556 012737 012564 001364 MOV     #11$,LOCK    ;SETUP FOR TIGHT SCOPE LOOP
(1) 012564 010510          11$: MOV    R5,(R0)   ;SET BIT IN DEVICE
(1) 012566 011004          MOV    (R0),R4      ;READ THE BIT FROM DEVICE
(1) 012570 020504          CMP    R5,R4        ;WAS BIT SET?
(1) 012572 001401          BEQ    2$          ;BR IF YES
(1) 012574 104002          ERROR+ 2          ;*BIT R/W FAILURE
(1) 012576 104401          2$: SCOP1          ;IS SWITCH 9 SET?
(1) 012600 012737 012606 001364 MOV     #12$,LOCK    ;SET FOR NEXT TIGHT SCOPE LOOP
(1) 012606 040510          12$: BIC    R5,(R0)   ;CLEAR THE BIT.
(1) 012610 011004          MOV    (R0),R4      ;READ DEVICE
(1) 012612 001403          BEQ    3$          ;BR IF BITS WERE CLEARED.
(1) 012614 005005          CLR    R5           ;CLEAR FOR ERROR PRINTOUT
(1) 012616 104002          ERROR+ 2          ;*BIT FAILED TO CLEAR
(1) 012620 011305          MOV    (R3),R5      ;RESTORE THE BIT.
(1) 012622 104401          3$: SCOP1          ;SW09 SET?
(1) 012624 012737 012632 001364 MOV     #13$,LOCK    ;SET UP FOR NEXT TIGHT SCOPE
(1) 012632 010510          13$: MOV    R5,(R0)   ;SET THE BIT AGAIN
(1) 012634 104413          DEVICE.CLR          ;ISSUE DEVICE CLEAR
(1) 012636 011004          MOV    (R0),R4      ;READ THE BIT.
(1) 012640 001403          BEQ    4$          ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
(1) 012642 005005          CLR    R5           ;SET EXPECTED TO ZERO
(1) 012644 104002          ERROR+ 2          ;*BIT NOT CLEARED BY DEVICE CLEAR
(1) 012646 011305          MOV    (R3),R5      ;RESTORE BIT AGAIN
(1) 012650 104401          4$: SCOP1          ;SW09 SET?
(1) 012652 062703 000002 ADD     #2,R3        ;POP R3
(1) 012656 005713          TST    (R3)         ;IS THIS THE END OF TABLE?
(1) 012660 001407          BEQ    6$          ;IF YES GET OUT
(1) 012662 000734          BR     1$          ;OTHERWISE TEST NEXT BIT
(1) 012664 000010          5$: #MAINT          ;CSR BIT: INTERNAL MAINTENANCE
(1) 012666 000040          #MSENAB          ;CSR BIT: MASTER SCAN ENABLE
(1) 012670 010000          #SILOEN          ;CSR BIT: SILO ENABLE
(1) 012672 000100          #RIE             ;CSR BIT: RECEIVER INTER. ENABLE
(1) 012674 040000          #TIE             ;CSR BIT: TRANS. INTER. ENABLE

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-49
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

(1) 012676 000000          #0          :END OF TABLE
(1) 012700 005037 001364 6$: CLR      LOCK      :ZERO LOCK INDICATOR
8754
(1)                               :***** TEST 4 *****
(1)                               :*THIS TESTS THAT ALL OF THE TCR BITS
(1)                               :*CAN BE: SET, CLEARED, AND CLEARED BY A DEVICE CLEAR.
(1)                               :*THIS TEST ALSO DETERMINES IF THE DTR BITS CAN
(1)                               :*BE SET, CLEARED, AND CLEARED BY A RESET.
(3)                               ::* TEST 4
(6)                               :*****
(5) 012704 000004          TST4: SCOPE
(3) 012706 012737 000004 001246 MOV     #4,STSTNM      :LOAD THE NUMBER OF THIS TEST
(3) 012714 012737 013110 001362 MOV     #TST5,NEXT    :POINT TO THE START OF THE NEXT TEST
(1) 012722 013700 002024          MOV     DZVTCR,R0     :SET DEVICE ADDRESS
(1) 012726 012703 013014          MOV     #5$,R3        :SET R3 POINTER TO TOP OF TABLE
(1) 012732 012737 012742 001364 1$: MOV     #11$,LOCK     :SET LOCK FOR SW09 SCOPE LOOP
(1) 012740 011305          MOV     (R3),R5       :SET EXPECTED RESULTS
(1) 012742 010510          MOV     R5,(R0)       :SET THE BIT
(1) 012744 011004          MOV     (R0),R4       :READ THE BIT FROM THE DEVICE
(1) 012746 020504          CMP     R5,R4         :DID THE BIT SET?
(1) 012750 001401          BEQ     2$            :BR IF YES
(1) 012752 104002          ERROR+ 2             :*BIT FAILED TO SET.
(1) 012754 104401          SCOP1                :SW09 SET?
(1) 012756 012737 012764 001364 2$: MOV     #3$,LOCK     :SET UP FOR NEXT TIGHT SCOPE LOOP
(1) 012764 040510          3$: BIC     R5,(R0)     :CLEAR THE BIT
(1) 012766 011004          MOV     (R0),R4       :READ THE REGISTER
(1) 012770 001403          BEQ     4$            :BR IF YES
(1) 012772 005005          CLR     R5            :SET EXPECTED TO 0
(1) 012774 104002          ERROR+ 2             :*REPORT BIT NOT CLEAR
(1) 012776 011305          MOV     (R3),R5       :RESTORE R5
(1) 013000 104401          4$: SCOP1                :SW09 SET?
(1) 013002 062703 000002          ADD     #2,R3         :POP POINTER TO NEXT TABLE ENTRY
(1) 013006 005713          TST     (R3)          :END OF TABLE?
(1) 013010 001412          BEQ     6$            :IF YES JUMP OVER TABLE
(1) 013012 000747          BR     1$            :START TESTING NEXT BIT
(1) 013014 000001          5$: #TCR0                :TCR BIT FOR LINE 0
(1) 013016 000002          #TCR1                :TCR BIT FOR LINE 1
(1) 013020 000004          #TCR2                :TCR BIT FOR LINE 2
(1) 013022 000010          #TCR3                :TCR BIT FOR LINE 3
(1) 013024 000400          #DTR0                :DTR BIT FOR LINE 0
(1) 013026 001000          #DTR1                :DTR BIT FOR LINE 1
(1) 013030 002000          #DTR2                :DTR BIT FOR LINE 2
(1) 013032 004000          #DTR3                :DTR BIT FOR LINE 3
(1) 013034 000000          #0                    :END OF TABLE
(1) 013036 005037 001364 6$: CLR     LOCK         :CLEAR TIGHT SCOPE LOOP INDIC.
(1) 013042 012710 177777          MOV     #-1,(R0)     :SET ALL BITS IN TCR REGISTER
(1) 013046 012705 007400          MOV     #007400,R5   :SET EXPECTED
(1) 013052 104413          DEVICE.CLR          :SET DCLR BIT IN CSR
(1) 013054 011004          MOV     (R0),R4       :READ REGISTER
(1) 013056 020504          CMP     R5,R4         :TCR BITS CLEARED?
(1) 013060 001401          BEQ     7$            :IF YES BRANCH
(1) 013062 104002          ERROR+ 2             :TCR BITS NOT CLEARED!
(1) 013064 005005          7$: CLR     R5         :SET EXPECTED TO ZERO
(1) 013066 005227 000000 8$: INC     #0           :DELAY FOR ACT
(1) 013072 001375          BNE     8$            :
(1) 013074 012710 177777          MOV     #-1,(R0)     :SET ALL POSSIBLE BITS
(1) 013100 000005          RESET                :DO BUS INIT

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-50
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

(1) 013102 011004
 (1) 013104 001401
 (1) 013106 104002
 (1) 013110
 8755
 (1)
 (1)
 (1)
 (1)
 (1)
 (3)
 (6)
 (5) 013110 000004
 (3) 013112 012737 000005 001246
 (3) 013120 012737 013212 001362
 (1) 013126 013700 002010
 (1) 013132 104413
 (1) 013134 005005
 (1) 013136 012710 121600
 (1)
 (1) 013142 011004
 (1) 013144 001401
 (1) 013146 104002
 (1) 013150 012705 100040
 (1) 013154 052777 000017 166642
 (1) 013162 052710 000040
 (1) 013166 005002
 (1) 013170 011004
 (1) 013172 042704 001400
 (1) 013176 020504
 (1) 013200 001404
 (1) 013202 104414
 (1) 013204 005202
 (1) 013206 001370
 (1) 013210 104002
 (1) 013212

```

MOV (R0),R4 ;DID REGISTER CLEAR?
BEQ 9$ ;IF YES GET OUT
ERROR+ 2 ;REGISTER DID NOT CLEAR!
9$:
;***** TEST 5 *****
;THIS TEST VERIFIES THAT
;BITS 'RDONE,TRDY, BIT9, BIT8,
;AND SILOAL' ARE READ ONLY AND THAT TRDY IS
;ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.

```

```

;:* TEST 5
;*****
TST5: SCOPE
MOV #5,$TSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST6,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZVCSR,R0 ;SET ADDRESS TO R0
DEVICE.CLR ;DO A DEVICE CLEAR
CLR R5 ;SET EXPECTED TO 0
MOV #RDONE+TRDY+BIT9+BIT8+SILOAL,(R0)
MOV (R0),R4 ;WRITE THE BITS
BEQ 2$ ;READ BACK THE BITS
ERROR+ 2 ;BR IF NONE ARE SET.
;BITS WERE SET.
MOV #TRDY+MSENAB,R5 ;SET EXPECTED BIT
BIS #17,@DZVTCR ;SET TCR BITS FOR ALL LINES
BIS #MSENAB,(R0) ;SET SCAN ENABLE
CLR R2 ;SET COUNTER TO ZERO
MOV (R0),R4 ;READ THE REGISTER
BIC #BIT9!BIT8,R4 ;MASK OUT LINE NO.
CMP R5,R4 ;BIT SET?
BEQ 4$ ;BR IF YES
DELAY ;STALL TIME
INC R2 ;UPDATE COUNTER
BNE 3$ ;BR IF COUNTER NOT DONE.
ERROR+ 2 ;*TRDY NOT SET!
2$:
3$:
4$:

```

8756
 8757
 8758
 8759
 8760
 8762
 (5)
 (4) 013212 000004
 (2) 013214 012737 000006 001246
 (2) 013222 012737 013342 001362
 8763 013230 104413
 8764 013232 013700 002010
 8765 013236 012710 177757
 8766 013242 012705 050150
 8767 013246 011004
 8768 013250 020405
 8769 013252 001401
 8770 013254 104002
 8771 013256 105010
 8772 013260 105005

```

;***** TEST 6 *****
;THIS TEST VERIFIES THAT:
;TIE,SILOEN,RIE,MSENAB,AND MAINT ARE THE
;ONLY R/W BITS IN THE DZVCSR AND THAT
;SETTING 'DCLR' IN THE CSR WILL CLEAR THESE BITS.
;:* TEST 6
;*****
TST6: SCOPE
MOV #6,$TSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST7,NEXT ;POINT TO THE START OF THE NEXT TEST
DEVICE.CLR ;SET DCLR IN CSR
MOV DZVCSR,R0 ;SET UP FOR ERROR MESSAGE
MOV #^C<DCLR>,(R0) ;TRY TO SET ALL BITS EXCEPT DCLR
MOV #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
MOV (R0),R4 ;ACTUAL
CMP R4,R5 ;CMP EXPECTED VS ACTUAL
BEQ 1$ ;YES
ERROR+ 2 ;*NO
1$: CLRB (R0) ;CLEAR LOW BYTE OF CSR
CLRB R5 ;CLEAR LOW BYTE OF EXPECTED DATA

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-51
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

8773	013262	011004			MOV	(R0),R4	:READ CSR
8774	013264	020405			CMP	R4,R5	:DOES CSR COMPARE WITH EXPECTED?
8775	013266	001401			BEQ	3\$:BRANCH IF YES
8776	013270	104002			ERROR+	2	:IF NOT PRINT ERROR
8777	013272	012710	177757	3\$:	MOV	#^C<DCLR>,(R0)	:SET ALL CSR BITS POSSIBLE
8778	013276	105077	166510		CLRB	@HDZVCSR	:CLEAR HIGH BYTE OF CSR
8779	013302	012705	000150		MOV	#RIE!MSENAB!MAINT,R5	:SET EXPECTED IN R5
8780	013306	011004			MOV	(R0),R4	:READ CSR REGISTER
8781	013310	020405			CMP	R4,R5	:DOES ACTUAL=EXPECTED
8782	013312	001401			BEQ	4\$:IF YES CONTINUE
8783	013314	104002			ERROR+	2	:IF NO PRINT ERROR
8784	013316	012710	177757	4\$:	MOV	#^C<DCLR>,(R0)	:SET ALL POSSIBLE CSR BITS
8785	013322	005005			CLR	R5	:SET R5 TO EXPECTED RESULTS
8786	013324	052710	000020		BIS	#DCLR,(R0)	:DEVICE MASTER RESET
8787	013330	000240			NOP		
8788	013332	011004			MOV	(R0),R4	:ACTUAL
8789	013334	020405			CMP	R4,R5	:CMP ACTUAL VS EXPECTED
8790	013336	001401			BEQ	2\$:YES
8791	013340	104002			ERROR+	2	:*NO
8792	013342			2\$:			

8793
(1) :***** TEST 7 *****
(1) :*THIS TEST PERFORMS RESET TESTING AND
(1) :*TESTING OF READ ONLY REGISTER DZVRBUF
(3) :*AND TESTING OF WRITE ONLY REGISTER DZVLPR

(6) :::* TEST 7
(5) 013342 000004 :*****
(3) 013344 012737 000007 001246 TST7: SCOPE
(3) 013352 012737 013426 001362 MOV #7,\$TSTNM :LOAD THE NUMBER OF THIS TEST
(1) 013360 104413 DEVICE.CLR :POINT TO THE START OF THE NEXT TEST
(1) 013362 013700 002014 MOV DZVRBUF,R0 :CLEAR DZV11
(1) 013366 011005 MOV (R0),R5 :SET UP FOR ERROR MESSAGE
(1) 013370 042705 106000 BIC #DVALID!BIT11!BIT10,R5 :COPY PRESENT CONTENTS
(1) 013374 012777 177777 166416 MOV #-1,@DZVLPR :CLEAR ILLEGAL BITS
(1) 013402 011004 MOV (R0),R4 :TRY TO WRITE ALL 1'S
(1) 013404 020405 CMP R4,R5 :ACTUAL
(1) 013406 001401 BEQ 1\$:CMP ACTUAL VS EXPECTED
(1) 013410 104002 ERROR+ 2 :IF YES,GO CONTINUE PROCESSING
(1) 013412 005077 166402 1\$: CLR @DZVLPR ;TRY TO WRITE ALL ZEROES :*ERROR- BIT PATTERN NOT CORRECT
(1) 013416 011004 MOV (R0),R4 :READ REGISTER
(1) 013420 020405 CMP R4,R5 :CMP ACTUAL VS. EXPECTED
(1) 013422 001401 BEQ 2\$:BRANCH IF EQUAL
(1) 013424 104002 ERROR+ 2 :VALUES DID NOT COMPARE
(1) 013426 2\$:

8794
(1) :***** TEST 10 *****
(1) :*THIS TEST PERFORMS RESET TESTING AND
(1) :*TESTING OF READ ONLY REGISTER DZVMSR
(3) :*AND TESTING OF WRITE ONLY REGISTER DZVTDR

(6) :::* TEST 10
(5) 013426 000004 :*****
(3) 013430 012737 000010 001246 TST10: SCOPE
(3) 013436 012737 013512 001362 MOV #10,\$TSTNM :LOAD THE NUMBER OF THIS TEST
(1) 013444 104413 DEVICE.CLR :POINT TO THE START OF THE NEXT TEST
(1) 013446 013700 002030 MOV DZVMSR,R0 :CLEAR DZV11
(1) 013452 011005 MOV (R0),R5 :SET UP FOR ERROR MESSAGE
 :COPY PRESENT CONTENTS

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-52
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

(1) 013454 042705 170360          BIC      #170360,R5      ;CLEAR ILLEGAL BITS
(1) 013460 112777 177777 166346  MOVB     #-1,@DZVTDR   ;TRY TO WRITE ALL 1'S
(1) 013466 011004                MOV      (R0),R4      ;ACTUAL
(1) 013470 020405                CMP      R4,R5        ;CMP ACTUAL VS EXPECTED
(1) 013472 001401                BEQ      1$           ;IF YES,GO CONTINUE PROCESSING
(1) 013474 104002                ERROR+  2             ;*ERROR- BIT PATTERN NOT CORRECT
(1) 013476 005077 166332 1$:    CLR      @DZVTDR ;TRY TO WRITE ALL ZEROES
(1) 013502 011004                MOV      (R0),R4      ;READ REGISTER
(1) 013504 020405                CMP      R4,R5        ;CMP ACTUAL VS. EXPECTED
(1) 013506 001401                BEQ      2$           ;BRANCH IF EQUAL
(1) 013510 104002                ERROR+  2             ;VALUES DID NOT COMPARE
(1) 013512

```

8795
8796
8797
8798
8799
8800
8801
8802
8803
8804
8806
8807

```

***** TEST 11 *****
*VERIFY THAT SETTING "DTR" FOR A LINE WILL
*BRING UP "CO" AND "RING" FOR:
*THE SAME LINE IF IN EXTERNAL MODE
*THE STAGGERED LINE IF IN STAGGERED MODE.
*LINES ARE STAGGERED AS FOLLOWS:
*LINE0 WITH LINE1; LINE2 WITH LINE3.
*THIS TEST IS ONLY RUN IF AN H325,OR H329
*IS CONNECTED ON THE DZV UNDER TEST.

```

::* TEST 11

```

(5)
(4) 013512 000004
(2) 013514 012737 000011 001246
(2) 013522 012737 013706 001362
8808 013530 005737 001372
8809 013534 001001
8810 013536 104400
8811 013540 012737 013630 001364 8$:
8812 013546 104413
8813 013550 013700 002030
8814 013554 005003
8815 013556 012702 000001
8816 013562 130237 001366 1$:
8817 013566 001003
8818 013570 005203 2$:
8819 013572 104420
8820 013574 000772
8821 013576 010204 3$:
8822 013600 105737 001372
8823 013604 100406
8824 013606 032703 000001
8825 013612 001402
8826 013614 006204
8827 013616 000401
8828 013620 006304 4$:
8829 013622 010405 5$:
8830 013624 000305
8831 013626 150405
8832 013630 150277 166172 10$:
8833 013634 104414
8834 013636 011004
8835 013640 020504

```

```

TST11: SCOPE
MOV      #11,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
MOV      #TST12,NEXT    ;POINT TO THE START OF THE NEXT TEST
TST      MODE           ;TEST TO SEE IF TESTING WITH
BNE      8$             ;CONNECTOR
ADVANCE   ;IF NO, GO TO NEXT TEST
MOV      #10$,LOCK      ;SET FOR TIGHT SCOPE LOOP
DEVICE.CLR ;SET DCLR IN CSR TO ZERO DEVICE
MOV      DZVMSR,R0      ;SET REGISTER
CLR      R3             ;ZERO LINE NUMBER
MOV      #1,R2          ;SET POINTER
BITB     R2,LINE        ;TEST THIS LINE?
BNE      3$             ;YES
INC      R3             ;LINE #
SHIFT    ;GET NEXT LINE
BR       1$             ;TEST NEXT LINE
MOV      R2,R4          ;SAVE BINARY BIT FOR LINE #
TSTB     MODE           ;RUNNING IN EXTERNAL MODE?
BMI      5$             ;IF YES SKIP STAGGERED SETUP
BIT      #BIT0,R3       ;IF EVEN LINE
BEQ      4$             ;GO GET ODD PARTNER
ASR      R4             ;OTHERWISE GET EVEN COMPANION
BR       5$             ;GO SETUP EXPECTED RESULTS
ASL      R4             ;FIND ODD PARTNER
MOV      R4,R5          ;LOAD R5 FOR EXPECTED
SWAB     R5             ;PLACE IN UPPER BYTE
BISB     R4,R5          ;SET FOR RING BITS
BISB     R2,@HDZVTCR    ;SET DTR BIT
DELAY    ;DELAY FOR CABLE LAG
MOV      (R0),R4        ;MOVE RESULTS OF MSR REGISTER TO R4
CMP      R5,R4          ;RESULTS=EXPECTED?

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-53
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

8836 013642 001401          BEQ      6$          :IF YES CONTINUE
8837 013644 104002          ERROR+  2          :IF NOT PRINT ERROR RESULTS
8838 013646 104401          6$: SCOPI         :IS SW09 SET?
8839 013650 012737 013656 001364  MOV      #1$,LOCK  :SET UP FOR NEXT TIGHT SCOPE
8840 013656 140277 166144 11$: BICB      R2,#HDZVTCR :CLEAR DTR BIT FOR LINE UNDER TEST
8841 013662 104414          DELAY                :DELAY FOR CABLE LAG
8842 013664 011004          MOV      (R0),R4    :LOAD MSR REGISTER INTO R4
8843 013666 001402          BEQ      7$          :IF CO AND RING CLEARED CONTINUE
8844 013670 005005          CLR      R5         :OTHERWISE SET EXPECTED FOR ERROR
8845 013672 104002          ERROR+  2          :PRINTOUT
8846 013674 104401          7$: SCOPI         :IS SW09 SET?
8847 013676 012737 013630 001364  MOV      #10$,LOCK :RESET TIGHT SCOPE LOOP
8848 013704 000731          BR       2$         :GET NEXT LINE
8849
8850          :***** TEST 12 *****
(1)          :* THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
(1)          :* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
(1)          :* FIED IN BITS 8-9 OF DZVCSR CORRESPOND
(1)          :* TO THE LINE SELECTED IN DZVTCR
(3)          :::* TEST 12
(6)          :*****
(5) 013706 000004          TST12: SCOPE
(3) 013710 012737 000012 001246  MOV      #12,$TSTNM :LOAD THE NUMBER OF THIS TEST
(3) 013716 012737 014040 001362  MOV      #TST13,NEXT :POINT TO THE START OF THE NEXT TEST
(1) 013724 104413          DEVICE.CLR         :ISSUE A 'DEVICE CLEAR' (RESET)
(1) 013726 012737 013762 001364  MOV      #2$,LOCK   :SET UP FOR TIGHT SCOPE LOOP
(1) 013734 005037 001374          CLR      SAVLIN    :INITIALIZE FOR ERROR PRINTOUT
(1) 013740 013700 002010          MOV      DZVCSR,R0 :SET POINTER
(1) 013744 012705 100040          MOV      #MSENAB!TRDY,R5 :START THE EXPECTED LINE NUMBER AT 0
(1) 013750 012702 000001          MOV      #1,R2     :USING R2 AS A BIT POINTER, POINT TO LINE 0
(1) 013754 130237 001366          1$: BITB      R2,LINE :IS THIS LINE SELECTED?
(1) 013760 001421          BEQ      6$         :IF NO, SKIP THE STARTUP
(1) 013762 050277 166036          2$: BIS       R2,#DZVTCR :SET THE GO BIT FOR THIS LINE
(1) 013766 052710 000040          BIS      #MSENAB,(R0) :START THE SCANNER
(1) 013772 005004          CLR      R4        :SET FOR DELAY
(1) 013774 005710          3$: TST      (R0)    :TX READY?
(1) 013776 100404          BMI     4$         :BR IF YES
(1) 014000 104414          DELAY                :DELAY
(1) 014002 005204          INC      R4        :COUNTER
(1) 014004 001373          BNE     3$         :BR IF <>0!
(1) 014006 104003          ERROR+  3          :*TX NOT READY!
(1) 014010 011004          4$: MOV      (R0),R4  :GET THE LINE POINTED TO BY THE SCANNER
(1) 014012 020405          CMP     R4,R5      :IS THE LINE NUMBER WHAT IT SHOULD BE?
(1) 014014 001401          PEQ     5$         :IF YES,GO WORK ON THE NEXT LINE
(1) 014016 104002          ERROR+  2          :*LINE NUMBER DID NOT MATCH TCR BIT
(1) 014020 104401          5$: SCOPI         :IS SW09 SET?
(1) 014022 104413          DEVICE.CLR         :SET DCLR IN CSR;SETUP FOR NEXT LINE
(1) 014024 062705 000400          6$: ADD      #400,R5  :POINT TO THE NEXT EXPECTED LINE
(1) 014030 104420          SHIFT                :POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
(1) 014032 005237 001374          INC     SAVLIN     :ADJUST FOR ERROR PRINTOUT
(1) 014036 000746          BR      1$        :IF NOT, GO DO THE NEXT LINE
8851          :***** TEST 13 *****
8852          :*TEST TO TRANSMIT ONE CHAR AND
8853          :*RECEIVE ONE CHAR ON ONE LINE
8854          :*AT A TIME. THE CHAR IS '252' AND
8855          :*ALL SELECTED LINES WILL BE TURNED ON .

```


CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-55
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

8898 014254 017704 165534      12$:  MOV    @DZVRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
8899 014260 020405                CMP    R4,R5             ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
8900 014262 001722                BEQ    5$                ;IF YES, GO DO THE NEXT LINE
8901 014264 104006                ERROR+ 6                ;*NO DATA/CONTENTS DID NOT COMPARE
8902 014266 000720                BR     5$                ;GO BACK AND WAIT TO CLEAR TCR BIT
8903 014270 104401      13$:  SCOP1                ;CHECK TO SEE IF SWITCH NINE IS SET
8904 014272 105037 001425        CLRB   DONFLG           ;SET UP FOR NEXT LINE
8905 014276 005077 165522        CLR    @DZVTCR         ;CLEAR PREVIOUS TCR BIT
8906 014302 005237 001374      15$:  INC    SAVLIN          ;SET LINE INDICATOR FOR NEXT LINE
8907 014306 104420                SHIFT                ;CALCULATE NEXT LINE
8908 014310 000702                BR     3$                ;GET GET STARTED
8909
8910                                ;TIGHT SCOPE LOOP FOR THIS TEST. LOOP TRANSMITS CHARACTERS ONLY
8911
8912 014312 005777 165472      16$:  TST    @DZVCSR         ;IS TRANSMITTER READY?
8913 014316 100375                BPL    16$              ;IF NOT, WAIT FOR IT
8914 014320 110177 165510        MOVB   R1,@DZVTDR      ;LOAD THE CHARACTER
8915 014324 104401                SCOP1                ;LOOP AGIN IF SW05=1
8916 014326 000760                BR     13$             ;OTHERWISE, GO PICK UP THE TEST NORMALLY
8917
8918                                ;***** TEST 14 *****
8919                                ;*THIS TEST VERIFIES THAT EACH RECEIVING LINE CAN BE
8920                                ;*DISABLED BY SETTING RCVON (BIT12 IN THE LPR REGISTER)
8921                                ;*TO ZERO FOR EACH LINE.
8922                                ;*THIS TEST ALSO VERIFIES THAT THE SILO CAN BE
8923                                ;*EMPTIED BY ISSUING A DEVICE MASTER CLEAR.
8924
8925                                ;:* TEST 14
8926                                ;*****
(5)
(4) 014330 000004
(2) 014332 012737 000014 001246  TST14: SCOPE
(2) 014340 012737 014652 001362    MOV    #14,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
8926 014346 105037 001425                MOV    #TST15,NEXT    ;POINT TO THE START OF THE NEXT TEST
8927 014352 005037 001374                CLRB   DONFLG         ;CLEAR TEST CONTROL FLAG
8928 014356 104417                CLR    SAVLIN         ;CLEAR LINE INDICATOR
8929                                DCLASM                ;ISSUE A DEVICE MASTER CLEAR
8930                                ;AND SET MAINT BIT IF NECESSARY
8931 014360 013701 001370                MOV    PAR,R1         ;SAVE DEFAULT PARAMETERS
8932 014364 042737 010000 001370        BIC    #RCVON,PAR     ;DISABLE RECEIVER IN DEFAULT PAR.
8933 014372 104421                LPRSET                ;LOAD PARAMETERS IN LPR REGISTER
8934 014374 010137 001370                MOV    R1,PAR         ;RESTORE DEFAULT PARAMETERS
8935 014400 012701 000252                MOV    #252,R1        ;LOAD A CHARAC. INTO R1
8936 014404 013702 001366                MOV    LINE,R2        ;COPY AN IMAGE OF THE ACTIVE LINES
8937 014410 010277 165410                MOV    R2,@DZVTCR    ;SET TCR BITS FOR ALL ACTIVE LINES
8938 014414 052777 000040 165366        BIS    #MSENAB,@DZVCSR ;SET MASTER SCAN ENABLE
8939 014422 005005                CLR    R5             ;INIT DELAY COUNTER
8940 014424 005777 165360      1$:  TST    @DZVCSR         ;IS TRANS READY SET?
8941 014432 104414                BMI    3$             ;BRANCH IF YES
8942 014434 005205                DELAY                ;WAIT FOR TRDY TO SET
8943 014436 001372                INC    R5             ;INCREMENT DELAY COUNTER
8944 014440 104003                BNE    2$             ;RETURN TO CHECK TRDY
8945 014442 117705 165344      3$:  ERROR+ 3                ;TRDY FAILED TO SET!
8946 014446 012703 000001                MOVB   @HDZVCSR,R5    ;MOVE LINE NO. TO R5
8947 014452 042705 177774                MOV    #1,R3         ;INJT TCR POINTER
8948 014456 001403                BIC    #^C<3>,R5      ;ISOLATE LINE NO.
8949 014460 106303                BEQ    31$            ;IF LINE 0 BRANCH
8950 014462 005305                ASLB   R3             ;SHIFT R3 POINTER TO NEXT LINE
                                DEC    R5             ;DECREMENT LINE NO.

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-56
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

8951 014464 001375
8952 014466 030302
8953 014470 001007
8954 014472 140377 165326
8955 014476 001351
8956 014500 105737 001425
8957
8958 014504 001037
8959 014506 000404
8960 014510 110177 165320
8961 014514 040302
8962 014516 000741
8963 014520 005077 165300
8964 014524 005005
8965 014526 105777 165256
8966 014532 100002
8967 014534 104020
8968 014536 000403
8969 014540 104414
8970 014542 005205
8971 014544 001370
8972 014546 017704 165242
8973 014552 100007
8974 014554 000304
8975 014556 042704 177774
8976 014562 010437 001374
8977 014566 104017
8978 014570 000766
8979 014572 105237 001425
8980 014576 013701 001370
8981 014602 000673
8982
8983 014604 005005
8984 014606 104414
8985 014610 005205
8986 014612 001375
8987 014614 104413
8988 014616 000240
8989 014620 000240
8990 014622 105777 165162
8991 014626 100003
8992 014630 005037 001374
8993 014634 104020
8994 014636 017704 165152
8995 014642 100003
8996 014644 005037 001374
8997 014650 104017
8998 014652
8999
9000
9001
9002
9003
9004
9006
(5)

```

```

31$: BNE 30$ ;WHEN R5=0, R3 POINTS TO LINE TCR
BIT R3,R2 ;HAS CHARACTER BEEN SENT?
BNE 4$ ;BRANCH IF NO
BICB R3,@DZVTCR ;IF YES THEN CLEAR TCR BIT
BNE 1$ ;IF ALL CHARAC. SENT DROP THROUGH
TSTB DONFLG ;IF NO MORE ACTIVE IS THIS SECOND
;TIME HERE?
BNE 10$ ;IF YES SKIP TO SECOND PART OF TEST
BR 5$ ;IF FIRST TIME HERE GO ZERO TCR BITS
4$: MOVB R1,@DZVTDR ;LOAD CHAR. INTO BUFFER
BIC R3,R2 ;INDICATE CHARAC. SENT ON THIS LINE
BR 1$ ;GO BACK AND WAIT FOR TRDY TO SET
5$: CLR @DZVTCR ;CLEAR OUT TCR BITS
CLR R5 ;INIT DELAY COUNTER
6$: TSTB @DZVCSR ;IS RECEIV. DONE SET?
BPL 7$ ;IF NOT THEN WAIT TO SEE IF IT WILL
ERROR+ 20 ;REC DONE SHOULD NOT SET!
BR 8$ ;GO FIND WHICH LINE RECEIVED
7$: DELAY ;STALL FOR RECEIVER
INC R5 ;INCREMENT DELAY COUNTER
BNE 6$ ;IF NOT DONE GO RETEST REC DONE
8$: MOV @DZVRBUF,R4 ;READ REC. BUFFER
BPL 9$ ;IS DVALID SET?
SWAB R4 ;IF YES GET LINE NO.
BIC #^C<3>,R4 ;ISOLATE LINE NO.
MOV R4,SAVLIN ;SET UP LINE NO. FOR ERROR REPORT
ERROR+ 17 ;DVALID SHOULD NOT BE SET
BR 8$ ;GO CHECK FOR ANY OTHER CHAR. IN SILO
9$: INCB DONFLG ;INDICATE THAT FIRST PART OF TEST IS DONE
MOV PAR,R1 ;SAVE DEFAULT LINE PARAM.
BR 100$ ;NOW GO RELOAD LPR REGISTER TO
;TURN RECEIVERS ON
10$: CLR R5 ;ZERO DELAY COUNTER
11$: DELAY ;WAIT FOR ALL CHARAC. TO BE RECEIVED
INC R5 ;INCREASE DELAY COUNT
BNE 11$ ;CONT. DELAY IF NOT FINISHED
DEVICE.CLR ;ISSUE A MASTER CLEAR
NOP
NOP
12$: TSTB @DZVCSR ;NOW IS RECEIV. DONE SET?
BPL 12$ ;BRANCH IF NO
CLR SAVLIN ;CLEAR LINE NO FOR ERROR REPORT
ERROR+ 20 ;REC. DONE SHOULD NOT BE SET!
MOV @DZVRBUF,R4 ;READ REC. BUFFER
BPL 13$ ;IS DVALID SET? IT SHOULDN'T BE
CLR SAVLIN ;DEVICE. CLR DID NOT ZERO SILO
ERROR+ 17 ;PRINT OUT THE ERROR.(LINE NO. IS IRRELEVANT)
13$:
;***** TEST 15 *****
; * THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
; * CHARACTERS (FLAG MODE) AND THE RECEIVER RECEIVES (FLAG MODE)
; * (ONE LINE AT A TIME BASED UPON VALID LINES)
; * THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
;::* TEST 15
;:*****

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-57
 CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

(4) 014652 000004          TST15: SCOPE
(2) 014654 012737 000015 001246  MOV      #15,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
(2) 014662 012737 015142 001362  MOV      #TST16,NEXT   ;POINT TO THE START OF THE NEXT TEST
(1) 014670 012737 014756 001364  MOV      #5$,LOCK     ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
9007 014676 104417          DCLASH                ;SET DCLR AND SET MNTFLG
9008 014700 104421          LPRSET                ;LOAD LPR REGISTER FOR ALL LINES
9009 014702 005037 001374  CLR      SAVLIN        ;INIT FOR FIRST LINE
9010 014706 104422          BUFSET                ;ZERO BUFFER AREA
9011 014710 105037 001425  CLRB     DONFLG        ;ZERO TCR BIT HANDLER FLAG
9012 014714 012702 000001  MOV      #1,R2         ;LINE POINTER
9013 014720 052777 000040 165062  BIS      #MSENAB,@DZVCSR ;START SCANNER
9014 014726 030237 001366 3$:     BIT      R2,LINE    ;VALID LINE ?
9015 014732 001477          BEQ      15$          ;NO SET UP NEXT LINE
9016 014734 010277 165064  MOV      R2,@DZVTCR    ;SET TCR BIT
9017 014740 013700 001374  MOV      SAVLIN,R0     ;ADJUST BUFFER POINTER
9018 014744 006300          ASL      R0           ;OFFSET
9019 014746 105777 165036 4$:     TSTB     @DZVCSR      ;IS REC DONE = 0 ?
9020 014752 100001          BPL      5$          ;IF YES, ALLOW TIME FOR TRDY TO SET
9021 014754 104020          ERROR+ 20           ;*REC DONE SHOULD = 0
9022 014756 005005          CLR      R5           ;USE R5 AS TIMER WAITING FOR TRDY TO SET
9023 014760 005777 165024 6$:     TST      @DZVCSR      ;IS THE TRANSMITTER READY?
9024 014764 100404          BMI      7$          ;IF SO, GO TRANSMIT A CHARACTER
9025 014766 104414          DELAY                    ;WAIT A LITTLE BIT
9026 014770 005205          INC      R5           ;UP THE LOCAL COUNTER.TIME EXCEEDED?
9027 014772 001372          BNE     6$           ;IF NOT, GO TRY AGAIN
9028 014774 104003          ERROR+ 3            ;*TRDY FAILED TO SET!
9029 014776 105737 001425 7$:     TSTB     DONFLG      ;ALL CHARAL. TRANS.?
9030 015002 001047          BNE     14$          ;IF YES GO ZERO TCR BIT
9031 015004 116077 001426 165022  MOVB     TDO(R0),@DZVTDR ;LOAD CHARACTER
9032 015012 013705 001374  MOV      SAVLIN,R5     ;MAKE EXPECTED LINE #
9033 015016 005737 001372  TST      MODE         ;IS THIS TEST IN STAGGERED MODE?
(1) 015022 100006          BPL      10$         ;IF NOT, SKIP STAGGERED SETUP

;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
(1)
(1)
(1) 015024 006205          ASR      R5           ;GET THE LAST BIT INTO THE CARRY BIT
(1) 015026 103402          BCS      8$          ;IF IT IS SET, GO CLEAR IT
(1) 015030 000261          SEC                    ;IF IT IS CLEAR SET IT HERE
(1) 015032 000401          BR      9$          ;SKIP THE CLEARING
(1) 015034 000241          8$:     CLC                    ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
(1) 015036 006105          9$:     ROL      R5           ;GET THE NEW BIT BACK INTO R5
9034 015040 000305          10$:    SWAB     R5           ;MOVE THE LINE NUMBER TO THE UPPER BYTE
9035 015042 156005 001426  BISB     TDO(R0),R5     ;ADD CHARACTER
9036 015046 052705 100000  BIS      #DVALID,R5   ;ADD DATA VALID
9037 015052 005003          CLR      R3           ;
9038 015054 105777 164730 11$:    TSTB     @DZVCSR      ;REC DONE?
9039 015060 100404          BMI     12$          ;IF YES GO CHECK CHAR.
9040 015062 104414          DELAY                    ;IF NOT WAIT FOR REC.
9041 015064 005203          INC      R3           ;DELAY LOOP TIMER
9042 015066 001372          BNE     11$          ;DELAY FINISHED?
9043 015070 104004          ERROR+ 4            ;*RDONE FAILED TO SET!
9044 015072 017704 164716 12$:    MOV      @DZVRBUF,R4   ;LOAD THE VALUE ACTUALLY RECEIVED
9045 015076 020405          CMP      R4,R5        ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
9046 015100 001401          BEQ     13$          ;IF YES, GO DO THE NEXT LINE
9047 015102 104006          ERROR+ 6            ;*NO DATA/CONTENTS DID NOT COMPARE
9048 015104 104401          13$:    SCOP1                ;CHECK TO SEE IF SWITCH NINE IS SET
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-58
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

9049	015106	105260	001426	INCB	TDO(R0)	:INCREMENT BINARY PATTERN FOR THIS LINE
9050	015112	001315		BNE	4\$:GO 'ROUND AGAIN FOR NEXT CHARACTER
9051	015114	105237	001425	INCB	DONFLG	:INDICATE ALL CHAR. SENT
9052	015120	000712		BR	4\$:BRANCH TO CLEAR TCR BIT
9053	015122	005077	164676	14\$: CLR	@DZVTCR	:CLEAR TCR REGISTER
9054	015126	105037	001425	CLRB	DONFLG	:INIT FOR NEXT LINE
9055	015132	005237	001374	15\$: INC	SAVLIN	:INC EXPECTED LINE
9056	015136	104420		SHIFT		:SHIFT THE LINE POINTER. ARE WE ALL DONE?
9057	015140	000672		BR	3\$:IF NO, GO AROUND AGAIN FOR NEXT LINE

9058
9059
9060
9061
9062
9063
9064
9065
9066
9067
(5)
(4)
(2)
(2)

```

:***** TEST 16 *****
:*THIS TEST WILL PROVE THAT:
:* 1) THE TRANSMITTER "BREAK BIT" WORKS
:* 2) THE RECEIVER CAN FLAG "FRAMING ERRORS"
:* 3) THE RECEIVER CAN FLAG "PARITY ERRORS"
:*ONLY ONE LINE AT A TIME WILL BE EXERCISED.

```

```

:.* TEST 16
:*****
TST16: SCOPE
MOV #16,$TSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST17,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV #5$,LOCK ;SET FOR LOOP
CLR SAVLIN ;INIT LINE INDIC. FOR ERROR PRINTOUT
MOV #1,R2 ;LINE POINTER
1$: BIT R2,LINE ;VALID LINE?
BEQ 9$ ;IF NOT SET FOR NEXT LINE
DCLASH ;SET DCLR IN CSR AND SET MNTFLG
MOV PAR,R1 ;PICK UP PARAMETERS
BIS #ODDPAR!PARITY,PAR ;FORCE ODD PARITY
LPRSET ;LOAD LPR REGISTER
MOV R1,PAR ;RESET PAR TO ORIGINAL VALUE
BIS #MSENAB,@DZVCSR ;START SCANNER
MOV SAVLIN,R5 ;MAKE EXPECTED DATA
TST MODE ;IS THIS TEST IN STAGGERED MODE?
BPL 4$ ;IF NOT, SKIP STAGGERED SETUP

;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
(1) 015142 000004
(1) 015144 012737 000016 001246
(1) 015152 012737 015344 001362
9068 015160 012737 015270 001364
9069 015166 005037 001374
9070 015172 012702 000001
9071 015176 030237 001366
9072 015202 001454
9073 015204 104417
9074 015206 013701 001370
9075 015212 052737 000300 001370
9076 015220 104421
9077 015222 010137 001370
9078 015226 052777 000040 164554
9079 015234 013705 001374
9080 015240 005737 001372
(1) 015244 100006
(1)
(1)
(1)
(1) 015246 006205
(1) 015250 103402
(1) 015252 000261
(1) 015254 000401
(1) 015256 000241
(1) 015260 006105
9081 015262 000305
9082 015264 052705 130000
9083 015270 005003
9084 015272 110277 164540
9085 015276 105777 164506
9086 015302 100404
9087 015304 104414
9088 015306 005203
9089 015310 001372
9090 015312 104004
9091 015314 017704 164474

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-59
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

9092 015320 020405          CMP      R4,R5          ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
9093 015322 001401          BEQ      8$              ;IF YES, GO CLEAN UP
9094 015324 104006          ERROR+  6              ;*DATA/CONTENTS FAILED TO COMPARE
9095 015326 105077 164504 8$:  CLRB    @DZVTDR        ;CLEAR BREAK BITS
9096 015332 104401          SCOP1                    ;LOOP?
9097 015334 005237 001374 9$:  INC     SAVLIN          ;INC LINE #
9098 015340 104420          SHIFT                    ;SET R2 TO NEXT LINE
9099 015342 000715          BR      1$              ;GO BACK AND TEST NEXT LINE
9100                                     ;***** TEST 17 *****
(1)                                     ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
(1)                                     ;*WHILE THE PROCESSOR STATUS DOES NOT ALLOW INTERRUPTS
(1)                                     ;*BUT WILL INTERRUPT IF THE PROCESSOR STATUS
(1)                                     ;*ALLOWS INTERRUPTS.
(3)                                     ;:* TEST 17
(6)                                     ;*****
(5) 015344 000004          TST17: SCOPE
(3) 015346 012737 000017 001246  MOV     #17,$STSTNM     ;LOAD THE NUMBER OF THIS TEST
(3) 015354 012737 015740 001362  MOV     #TST20,NEXT    ;POINT TO THE START OF THE NEXT TEST
(1) 015362 104417          DCLASM                    ;SET DCLR IN CSR AND SET MAINT BIT
(1)                                     ;IF NECESSARY (INTERNAL MODE)
(1) 015364 104421          LPRSET                    ;SET UP LPR REGISTER
(1) 015366 005037 001374  CLR     SAVLIN          ;INIT LINE INDIC. FOR ERROR
(1) 015372 105037 001425  CLRB   DONFLG          ;INIT TCR BIT HANDLER FLAG
(1) 015376 113777 001366 164420  MOVB   LINE,@DZVTCR    ;SET ALL VALID TCR BITS
(1) 015404 106427 000200  MTPS   #MASK           ;SET CPU STATUS TO DZV11 PRIO.
(1) 015410 012777 000200 164424  MOV     #MASK,@DZVRIS  ;SET RECEIVER STATUS
(1) 015416 012777 000200 164422  MOV     #MASK,@DZVTIS  ;SET TRANSMITTER STATUS
(1) 015424          1$:  MOV     #6$,@DZVTIV    ;SET UP THE TRANSMITTER INTERRUPT VECTOR
(2) 015424 012777 015512 164412  MOV     #7$,@DZVRIV    ;SET UP THE RECEIVER INTERRUPT VECTOR
(2) 015432 012777 015534 164400  MOV     #MASK,@DZVRIS  ;SET THE INTERRUPT VECTOR STATUS
(2) 015440 012777 000200 164374  MOV     #MASK,@DZVTIS  ;SET TRANSMITTER INTERRUPT PRIORITY
(2) 015446 012777 000200 164372  BIS    #TIE!MSENAB,@DZVCSR ;ENABLE THE DEVICE
(2) 015454 052777 040040 164326  CLR    R5              ;INIT DELAY COUNTER
(1) 015462 005005          CLR    R5              ;TRDY SET?
(1) 015464 005777 164320 4$:  TST    @DZVCSR        ;IF NOT GO DO DELAY
(1) 015470 100003          BPL    5$              ;WAIT FOR INTERRUPT
(1) 015472 000240          NOP
(1) 015474 000240          NOP
(1) 015476 000420          BR     8$              ;GO CLEAR TIE BIT
(1) 015500 104414          5$:  DELAY                    ;DELAY ROUTINE CALL
(1) 015502 005205          INC    R5              ;INC DELAY COUNTER
(1) 015504 001367          BNE    4$              ;DELAY FINISHED?
(1) 015506 104003          ERROR+ 3              ;*TRDY NOT SET!
(1) 015510 000413          BR     8$              ;GO CLEAR TIE
(1) 015512 022626          6$:  POP2SP                    ;REMOVE THE INTERRUPT FROM THE STACK
(1) 015514 042777 040000 164266  BIC    #TIE,@DZVCSR    ;DON'T LET ANY MORE INTERRUPTS OCCUR
(1) 015522 105737 001425          TSTB   DONFLG          ;PROCESSOR ALLOWING INTER?
(1) 015526 001013          BNE    10$             ;IF YES NO ERROR
(1) 015530 104010          ERROR+ 10             ;IF NOT PRINT ERROR
(1) 015532 000413          BR     9$              ;RETURN TO THE NORMAL FLOW
(1) 015534 104012          7$:  ERROR+ 12             ;*RECEIVER SHOULD NOT INTERRUPT
(1) 015536 022626          POP2SP                    ;POP FOR FAKE RTI
(1) 015540 042777 040000 164242 8$:  BIC    #TIE,@DZVCSR    ;RESET TRANSMITTER INTERKRYPT ENABLE
(1) 015546 105737 001425          TSTB   DONFLG          ;INTERRUPTS ENABLED?
(1) 015552 001403          BEQ    9$              ;IF NOT GET OUT
(1) 015554 104007          ERROR+ 7              ;IF YES TRANS FAILED TO INTER.

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-60
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

(1) 015556 106427 000000 10$: MTPS #CLEAR ;ALLOW INTERRUPTS
(1) 015562 9$:
(2) 015562 012777 015666 164254 MOV #11$,@DZVTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
(2) 015570 012777 015672 164242 MOV #12$,@DZVRIV ;SET UP THE RECEIVER INTERRUPT VECTOR
(2) 015576 012777 000200 164236 MOV #MASK,@DZVRIS ;SET THE INTERRUPT VECTOR STATUS
(2) 015604 012777 000200 164234 MOV #MASK,@DZVTIS ;SET TRANSMITTER INTERRUPT PRIORITY
(2) 015612 052777 000140 164170 BIS #RIE!MSENAB,@DZVCSR ;ENABLE THE DEVICE
(1) 015620 113777 001426 164206 MOVB TDO,@DZVTDR ;LOAD BUFFER WITH ANY CHAR.
(1) 015626 005005 CLR R5 ;INIT DELAY ACCUMULATOR
(1) 015630 105777 164154 13$: TSTB @DZVCSR ;REC. DONE?
(1) 015634 100003 BPL 14$ ;IF NOT DELAY
(1) 015636 000240 NOP ;WAIT FOR INTERRUPT
(1) 015640 000240 NOP
(1) 015642 000404 BR 18$
(1) 015644 104414 14$: DELAY ;DELAY FOR INTERRUPT
(1) 015646 005205 INC R5 ;INCREMENT DELAY COUNTER
(1) 015650 001367 BNE 13$ ;DELAY FINISHED?
(1) 015652 104004 ERROR+ 4 ;*NO RX DONE! (NOT SET)
(1) 015654 105737 001425 18$: TSTB DONFLG ;PROCESSOR ALLOWING INTERRUPTS?
(1) 015660 001411 BEQ 15$ ;IF NOT DON'T PRINT ERROR
(1) 015662 104011 ERROR+ 11 ;RECEIVER FAILED TO INTERRUPT
(1) 015664 000407 BR 15$ ;CONTINUE TEST
(1) 015666 104010 11$: ERROR+ 10 ;TRANSMITTER SHOULD NOT INTER.
(1) 015670 000404 BR 16$ ;CONT TEST
(1) 015672 105737 001425 12$: TSTB DONFLG ;PROCESSOR ALLOWING INTERRUPTS?
(1) 015676 001001 BNE 16$ ;IF YES DON'T PRINT ERROR
(1) 015700 104012 ERROR+ 12 ;*RECEIVER SHOULD NOT INTERRUPT
(1) 015702 022626 16$: POP2SP ;POP FOR FAKE RTI
(1) 015704 042777 040100 164076 15$: BIC #RIE!TIE,@DZVCSR ;CLEAR INTERRUPTS
(1) 015712 105737 001425 TSTB DONFLG ;SECOND TIME THROUGH?
(1) 015716 001005 BNE 17$ ;IF YES LEAVE TEST
(1) 015720 105237 001425 INCB DONFLG ;IF NO INDICATE SECOND TEST PASS
(1) 015724 106427 000000 MTPS #CLEAR ;ALLOW INTERRUPTS
(1) 015730 000635 BR 18$ ;RESTART TEST
(1) 015732 106427 000200 17$: MTPS #MASK ;DON'T ALLOW INTERRUPTS
(1) 015736 104413 DEVICE.CLR ;CLEAR DEVICE, LEAVE TEST

```

```

9101
9102
9103
9104
9105
9106
9107
9108
9109
9111
::* TEST 20
:***** TEST 20 *****
:*THIS TEST VERIFIES THAT THE RECEIVER WILL
:*INTERRUPT BEFORE THE TRANSMITTER EVEN
:*THOUGH THE TRANSMITTER WAS ENABLED
:*FIRST. SET PS TO HIGH (MASK INTERRUPTS);
:*GET RDONE AND TRDY TO SET;
:*SET TX IE AND RX IE;
:*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST

```

```

(5)
(4) 015740 000004
(2) 015742 012737 000020 001246 TST20: SCOPE
(1) 015750 012737 004246 001362 MOV #20,$TSTNM ;LOAD THE NUMBER OF THIS TEST
9112 015756 104417 DCLASH ;POINT TO THE END-OF-PASS HANDLER
9113 015760 104421 LPRSET ;SET DCLR IN CSR AND MNTFLG
9114 015762 005037 001374 CLR SAVLIN ;LOAD PAR REGISTER FOR ALL LINES
9115 015766 012777 016176 164044 MOV #8,@DZVRIV ;INIT. ERROR LINE INDIC.
9116 015774 012777 000200 164040 MOV #MASK,@DZVRIS ;SETUP INTERRUPT STUFF
9117 016002 012777 016264 164034 MOV #12$,@DZVTIV ;

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-61
 CVDZAD.P11 11-MAR-83 10:06 DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

9118 016010 012777 000200 164030      MOV      #MASK,@DZVTIS      ;
9119 016016 052777 000040 163764      BIS      #MSENAB,@DZVCSR
9120 016024 012702 000001                MOV      #1,R2              ;LINE POINTER
9121 016030 030237 001366      3$:     BIT      R2,LINE      ;VALID LINE ?
9122 016034 001515                BEQ      14$                ;IF NOT GO TO NEXT LINE
9123 016036 106427 000200      4$:     MTPS     #MASK
9124 016042 110277 163756      MOVB     R2,@DZVTCR        ;SET TCR BIT
9125 016046 005777 163742      TST      @DZVRBUF          ;VALID DATA?
9126 016052 100001                BPL      .+4                ;IT BETTER NOT BE SET
9127 016054 104017                ERROR+  17                  ;DATA VALID SHOULD NOT BE SET
9128 016056 105777 163726      5$:     TSTB     @DZVCSR      ;RECEIVER DONE ?
9129 016062 100001                BPL      .+4
9130 016064 104020                ERROR+  20                  ;RECEIVER DONE BIT SHOULD NOT BE SET
9131 016066 005005                CLR      R5
9132 016070 005004                CLR      R4
9133 016072 005777 163712      99$:    TST      @DZVCSR        ;WAIT FOR TRDY
9134 016076 100404                BMI      100$              ;BR IF READY
9135 016100 104414                DELAY
9136 016102 005204                INC      R4
9137 016104 001372                BNE      99$
9138 016106 104003                ERROR+  3
9139 016110 105077 163720      100$:   CLRB     @DZVTDR        ;TRDY FAILED TO SET
9140 016114 005004                CLR      R4                ;SEND A ZERC CHARACTER
9141 016116 105777 163666      6$:     TSTB     @DZVCSR      ;IS RDONE SET?
9142 016122 100404                BMI      7$
9143 016124 104414                DELAY
9144 016126 005204                INC      R4
9145 016130 001372                BNE      6$
9146 016132 104004                ERROR+  4
9147 016134 005777 163650      7$:     TST      @DZVCSR        ;*RDONE FAILED TO SET!
9148 016140 100401                BMI      .+4                ;TRANS DONE BIT = 1 ?
9149 016142 104003                ERROR+  3                    ;YES
9150                ;*NO TRANS DONE FAILED TO SET
9151                ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
9152 016144 052777 040000 163636      BIS      #TIE,@DZVCSR      ;SET INTERRUPT ENABLES
9153 016152 052777 000100 163630      BIS      #RIE,@DZVCSR
9154 016160 106427 000000                MTPS     #CLEAR            ;ALLOW THE INTERRUPTS
9155 016164 000240                NOP
9156 016166 000240                NOP
9157 016170 104007                ERROR+  7                    ;*TRANSMITTER FAILED TO INTERRUPT
9158 016172 104011                ERROR+  11                  ;*RECEIVER FAILED TO INTERRUPT
9159 016174 000435                BR       14$                ;GET OUT
9160
9161                ;RECEIVER INTERRUPT ROUTINE
9162 016176 017704 163612      8$:     MOV      @DZVRBUF,R4      ;ACTUAL
9163 016202 010403                MOV      R4,R3
9164 016204 000303                SWAB     R3
9165 016206 042703 177770                BIC      #^C<7>,R3          ;STRIP JUNK
9166 016212 005737 001372                TST      MODE              ;IS THIS TEST IN STAGGERED MODE?
(1) 016216 100006                BPL      11$                ;IF NOT, SKIP STAGGERED SETUP
(1)
(1)                ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
(1)
(1) 016220 006203                ASR      R3                  ;GET THE LAST BIT INTO THE CARRY BIT
(1) 016222 103402                BCS     9$                  ;IF IT IS SET, GO CLEAR IT
(1) 016224 000261                SEC
    
```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 10-62
 CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

(1)	016226	000401			BR	10\$;SKIP THE CLEARING
(1)	016230	000241		9\$:	CLC			;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
(1)	016232	006103		10\$:	ROL	R3		;GET THE NEW BIT BACK INTO R3
9167	016234	020337	001374	11\$:	CMP	R3,SAVLIN		;IS THIS A VALID LINE
9168	016240	001401			BEQ	.+4		;YES
9169	016242	104015			ERROR+	15		;*INVALID LINE
9170	016244	042704	177400		BIC	#^C<377>,R4		;STRIP JUNK
9171	016250	120504			CMPB	R5,R4		;DATA COMPARE ?
9172	016252	001401			BEQ	.+4		;YES
9173	016254	104005			ERROR+	5		;*DATA DOES NOT COMPARE
9174	016256	040277	163542		BIC	R2,@DZVTCR		;CLEAR TCR BIT
9175	016262	000401			BR	13\$;GO GET OUT OF INTERRUPT MODE
9176								;TRANSMITTER INTERRUPT SVC ROUTINE
9177	016264	104011		12\$:	ERROR+	11		;THE RECEIVER INTERRUPT FAILED
9178								;TO OVERRIDE THE TRANSMITTER
9179	016266	022626		13\$:	POP2SP			;REMOVE THE INTERRUPT VECTOR FROM THE STACK
9180	016270	005237	001374	14\$:	INC	SAVLIN		;ADJUST FOR NEXT LINE
9181	016274	104420			SHIFT			;GET THE NEXT POINTER. IF DONE, ADVANCE
9182	016276	000137	016030		JMP	3\$;OTHERWISE GO DO THE NEXT LINE

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 11
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

Line No.	Code 1	Code 2	Code 3	Code 4	Code 5
9184					
9185					
9186	016302	000000			
9187	016304	000000			
9188	016306	000000			
9189					
9190	016310	016450			
9191	016312	017266			
9192	016314	017406			
9193					
9194	016316	016523			
9195	016320	017312			
9196	016322	017420			
9197					
9198	016324	016551			
9199	016326	017345			
9200	016330	017436			
9201					
9202	016332	016610			
9203	016334	017345			
9204	016336	017436			
9205					
9206	016340	016637			
9207	016342	017357			
9208	016344	017444			
9209					
9210	016346	016666			
9211	016350	017357			
9212	016352	017444			
9213					
9214	016354	016725			
9215	016356	017345			
9216	016360	017436			
9217					
9218	016362	016766			
9219	016364	017345			
9220	016366	017436			
9221					
9222	016370	017030			
9223	016372	017345			
9224	016374	017436			
9225					
9226	016376	017066			
9227	016400	017345			
9228	016402	017436			
9229					
9230	016404	000000			
9231	016406	000000			
9232	016410	000000			
9233					
9234	016412	000000			
9235	016414	000000			
9236	016416	000000			
9237					
9238	016420	017125			
9239	016422	000000			

Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
.ERRTAB:					
			0		:ERROR 0
			0		
			0		
			EM1		:ERROR
			DH1		
			DT1		
			EM2		:ERROR 2
			DH2		
			DT2		
			EM3		:ERROR 3
			DH3		
			DT3		
			EM4		:ERROR 4
			DH3		
			DT3		
			EM5		:ERROR 5
			DH4		
			DT4		
			EM6		:ERROR 6
			DH4		
			DT4		
			EM7		:ERROR 7
			DH3		
			DT3		
			EM10		:ERROR 10
			DH3		
			DT3		
			EM11		:ERROR 11
			DH3		
			DT3		
			EM12		:ERROR 12
			DH3		
			DT3		
			0		
			0		
			0		
			0		
			0		
			0		
			EM15		:ERROR 15
			0		

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 11-1
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

9240	016424	000000	0	
9241				
9242	016426	000000	0	
9243	016430	000000	0	
9244	016432	000000	0	
9245				
9246	016434	017167	EM17	;ERROR 17
9247	016436	017345	DH3	
9248	016440	017436	DT3	
9249				
9250	016442	017225	EM20	
9251	016444	017345	DH3	
9252	016446	017436	DT3	

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 12
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

```

9254
9255
9259 016450 047200 020117 052502 EM1: .ASCIZ <200>/NO BUS REPLY RESPONSE FROM DZV11 REGISTER/
9260 016523 200 042522 044507 EM2: .ASCIZ <200>/REGISTER R/W FAILURE?
9261 016551 200 051124 047101 EM3: .ASCIZ <200>/TRANSMIT READY (TRDY) NOT SET/
9262 016610 051200 041505 044505 EM4: .ASCIZ <200>/RECEIVER DONE NOT SET/
9263 016637 200 040504 040524 EM5: .ASCIZ <200>/DATA COMPARISON ERROR/
9264 016666 042200 053132 030461 EM6: .ASCIZ <200>/DZV11 *RECEIVER BUFFER* ERROR/
9265 016725 200 051124 047101 EM7: .ASCIZ <200>/TRANSMITTER FAILED TO INTERRUPT/
9266 016766 052600 042516 050130 EM10: .ASCIZ <200>/UNEXPECTED TRANSMITTER INTERRUPT/
9267 017030 051200 041505 044505 EM11: .ASCIZ <200>/RECEIVER FAILED TO INTERRUPT/
9268 017066 052600 042516 050130 EM12: .ASCIZ <200>/UNEXPECTED RECEIVER INTERRUPT/
9269 017125 200 041501 044524 EM15: .ASCIZ <200>/ACTION DETECTED ON INVALID LINE./
9270 017167 200 040504 040524 EM17: .ASCIZ <200>/DATA VALID SHOULD NOT BE SET/
9271 017225 200 042522 042503 EM20: .ASCIZ <200>/RECEIVER DONE SHOULD NOT BE SET/
9272
9273 017266 052200 040522 020120 DH1: .ASCIZ <200>/TRAP PC DZV11 REG/
9274 017312 042600 050130 041505 DH2: .ASCIZ <200>/EXPECTED FOUND REGISTER/
9275 017345 200 044514 042516 DH3: .ASCIZ <200>/LINE NO./
9276 017357 200 054105 042520 DH4: .ASCIZ <200>/EXPECTED FOUND LINE/
9277
9278 .EVEN
9282
9283 017406 000002 DT1: .DATA TABLES FOR ERROR MESSAGES
9284 017410 006 003 2
9285 017412 001330 .BYTE 6,3
9286 017414 006 001 $REG1
9287 017416 001326 .BYTE 6,1
9288 $REG0
9289 017420 000003 DT2: 3
9290 017422 006 004 .BYTE 6,4
9291 017424 001340 $REG5
9292 017426 006 001 .BYTE 6,1
9293 017430 001336 $REG4
9294 017432 006 001 .BYTE 6,1
9295 017434 001326 $REG0
9296
9297 017436 000001 DT3: 1
9298 017440 003 001 .BYTE 3,1
9299 017442 001374 SAVLIN
9300
9301 017444 000003 DT4: 3
9302 017446 006 004 .BYTE 6,4
9303 017450 001340 $REG5
9304 017452 006 001 .BYTE 6,1
9305 017454 001336 $REG4
9306 017456 003 001 .BYTE 3,1
9307 017460 001374 SAVLIN
9308 .EVEN
9309
9317
9318
9319
9320 017462 002450 DLYTBL: 2450
9321 017464 001560
9322 017466 001120

```

:TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES

:-----

```

:TIME FOR 50 BAUD
:TIME FOR 75 BAUD
:TIME FOR 110 BAUD

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 12-1
CVDZAD.P11 11-MAR-83 10:06

DZV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

9323	017470	000750	750	:TIME FOR 134 BAUD
9324	017472	000660	660	:TIME FOR 150 BAUD
9325	017474	000330	330	:TIME FOR 300 BAUD
9326	017476	000150	150	:TIME FOR 600 BAUD
9327	017500	000060	60	:TIME FOR 1200 BAUD
9328	017502	000040	40	:TIME FOR 1800 BAUD
9329	017504	000030	30	:TIME FOR 2000 BAUD
9330	017506	000020	20	:TIME FOR 2400 BAUD
9331	017510	000010	10	:TIME FOR 3600 BAUD
9332	017512	000001	1	:TIME FOR 4800 BAUD
9333	017514	000001	1	:TIME FOR 7200 BAUD
9334	017516	000001	1	:TIME FOR 9600 BAUD
9335	017520	000001	1	:TIME OF DELAY FOR 19200 BAUD

9336
9337
9338

:DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
:FOR ALL TESTS TO FUNCTION CORRECTLY ON A LSI11.

DZV11 DEVICE DIAGNOSTICS. COPYRIGHT 1977,1981 DIGITAL EQUIP. CORP.

9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357 017522 005227 177777
9358 017526 001002
9359 017530 004737 000400
9360 017534 005727
9361 017536 000000
9362 017540 000207
9363
9364 017542
9365 000400
9366 000400 005037 017536
9367 000404 013746 000004
9368 000410 012737 000504 000004
9369 000416 012700 160010
9370 000422 005720
9371 000424 000240
9372 000426 020027 174000
9373 000432 103773
9374 000434 010037 017536
9375 000440 012700 000040
9376 000444 040037 000006
9377 000450 040037 000016
9378 000454 040037 000022
9379 000460 040037 000032
9380 000464 040037 000036
9381 000470 012737 170000 000140
9382 000476 012637 000004
9383 000502 000207
9384
9385 000504 012716 000512
9386 000510 000002
9387 000512 012637 000004
9388 000516 012700 000402
9389 000522 013701 000376
9390 000526 010602
9391 000530 012704 000570
9392 000534 014446
9393 000536 020427 000546
9394 000542 101374
9395 000544 010607

```
.SBTTL FALCON (KXT-11) UPGRADE ROUTINES.                ;;GPA
:
: THE FOLLOWING ROUTINES HAVE BEEN ADDED TO ALLOW DIAGNOSTIC(S)
: TO RUN ON A FALCON (KXT-11) BASED SYSTEM.
: TO DETERMINE WHETHER WE'RE A FALCON OR NOT, WE'LL SIZE THE 1ST 3/4 OF
: THE I/O PAGE (28K TO 31K). FALCON HAS 2KW LOCAL RAM AT 28K(+4) TO 30K
: AND A MACRO-ODT AT 30K TO 31K. CONSEQUENTLY, ALL I/O DEVICES MUST
: BE PLACED BETWEEN 174000 AND 177776. ADDITIONALLY, WE'LL STRAP THE
: EMT AND TRAP SERVICE LEVEL TO PRI6, AND SET THE HALT VECTOR SO THAT
: WE CAN STOP THE SUCKER !!
:
: TO MINIMIZE THE IMPACT OF THESE CHANGES ON FINAL PROGRAM SIZE, THE
: BULK OF THIS CODE IS PLACED IN THE FLOATING VECTOR SPACE (400-776).
: IF THE CPU AT HAND IS A FALCON (KXT11), IT STAYS THERE (NO HARM DONE).
: OTHERWISE, THE AREA IS RESTORED TO ITS ORIGINAL "TRAP-CATCHER" STATE.
:
FALCON: INC      #-1                ; ONCE-ONLY !!!                ;;GPA
        BNE      1$                  ;                               ;;GPA
        CALL     KXTCHK                ; EXECUTE FALCON CHECK          ;;GPA
1$:     TST      (PC)+                ; TEST FALCON FLAG...         ;;GPA
KXTFLAG: 0                               ; ...NZ = FALCON...           ;;GPA
        RETURN                       ; ...AND RETURN TO CALLER...  ;;GPA
:
        $SVPC= .                      ;                               ;;GPA
        . = 400                        ; RESTORE FROM 374:376 AT END  ;;GPA
KXTCHK: CLR      KXTFLAG              ; ASSUME NOT FALCON.          ;;GPA
        MOV      @#4,-(SP)            ; SAVE ERROR VECTOR.         ;;GPA
        MOV      #2$,@#4              ; SET A TRAP CATCHER.        ;;GPA
1$:     MOV      #160010,R0           ; FALCON RAM STARTS AT 28K+4. ;;GPA
        TST      (R0)+                ;                               ;;GPA
        240                               ;                               ;;GPA
        CMP      R0,#174000           ; SIZE TO 31K.                ;;GPA
        BLO      1$                  ;                               ;;GPA
        MOV      R0,KXTFLAG           ; MUST BE FALCON, SET THE FLAG ;;GPA
        MOV      #40,R0               ; GET PRI1 BIT...             ;;GPA
        BIC      R0,@#6               ; ...AND LOWER BUS-ERROR...  ;;GPA
        BIC      R0,@#16              ; ...BPT...                   ;;GPA
        BIC      R0,@#22              ; ...IOT...                   ;;GPA
        BIC      R0,@#32              ; ...EMT...                   ;;GPA
        BIC      R0,@#36              ; ...AND TRAP SERVICE TO PRI6 ;;GPA
        MOV      #170000,@#140        ; ENABLE "BREAK" HALT.       ;;GPA
        MOV      (SP)+,@#4            ; RESTORE ERROR VECTOR...    ;;GPA
        RETURN                       ; ...AND RETURN.             ;;GPA
:
2$:     MOV      #3$, (SP)            ; TRAP -- NOT A FALCON...     ;;GPA
        RTI                               ; ...CONTINUE.                ;;GPA
3$:     MOV      (SP)+,@#4            ; RESET ERROR VECTOR          ;;GPA
        MOV      #402,R0              ; SET-UP TO RESTORE FLOATING... ;;GPA
        MOV      @#376,R1             ; ...VECTORS (400 - 776).    ;;GPA
        MOV      SP,R2                ; SAVE STACK POINTER IN R2   ;;GPA
        MOV      #6$,R4               ;                               ;;GPA
4$:     MOV      -(R4),-(SP)          ; PUSH THE RESTORE CODE...    ;;GPA
        CMP      R4,#5$              ; ...ONTO THE STACK.         ;;GPA
        BHI      4$                  ;                               ;;GPA
        MOV      SP,PC                ; AND EXECUTE IT.            ;;GPA
```

FALCON (KXT-11) UPGRADE ROUTINES.

::GPA

```

9397
9398
9399
9400 000546 010060 177776
9401 000552 010110
9402 000554 022020
9403 000556 020027 000776
9404 000562 101771
9405 000564 010206
9406 000566 000207
9407 000570
9408
9409
9410
9411
9412
9413
9414
9415
9416 000570 023727 001174 160010
9417 000576 001003
9418 000600 012737 174040 001174
9419 000606 023727 001170 000300
9420 000614 001003
9421 000616 012737 000370 001170
9422 000624 012737 000670 002462
9423 000632 012737 174000 002466
9424 000640 012737 177770 002470
9425 000646 012737 000732 002506
9426 000654 005037 002512
9427 000660 012737 000370 002514
9428 000666 000207
9429
9430 000670 030600 052123 041440
      000676 051123 040440 042104
      000704 042522 051523 024040
      000712 033461 030064 030060
      000720 030472 033467 033467
      000726 024460 000040
9431 000732 030600 052123 053040
      000740 041505 047524 020122
      000746 042101 051104 051505
      000754 020123 030050 030060
      000762 031472 030067 020051
      000770 020040 000040
9432
9433
9434
9438
9439
9440 017542
9444

```

```

:
: THIS CODE IS RELOCATED TO AND EXECUTED IN THE STACK AREA.
:
5$:  MOV     R0,-2(R0)      ; RESTORE .+2...
      MOV     R1,(R0)      ; ...HALT (OR IOT).
      CMP     (R0)+,(R0)+
      CMP     R0,#776
      BLOS   5$           ; LOOP 'TIL DONE
      MOV     R2,SP        ; THEN RESTORE SP...
      RETURN              ; ...AND RETURN TO CALLER
:
6$:
:
: IF FALCON, THIS AREA IS FREE FOR ANY PROGRAM UNIQUE
: CHANGES OR DATA STRUCTURES.
: BE SURE THAT IT DOESN'T GET SCREWED UP !!
:
: INIT $BASE AND $VECT1 AND TWEAK THE "$GETPAR" CALLING
: SEQUENCE TO ACCEPT THE VALID FALCON RANGE.
:
FALCINI: CMP     $BASE,#ABASE ; IS $BASE VIRGIN ??
        BNE     1$         ; SKIP NEXT IF NOT
        MOV     #174040,$BASE ; YES, SET ENGINEERING DEFAULT
1$:     CMP     $VECT1,#AVECT1 ; IS $VECT1 VIRGIN ??
        BNE     2$         ; SKIP NEXT IF NOT
        MOV     #370,$VECT1 ; YES, SET ENGINEERING DEFAULT
2$:     MOV     #3$,GETCSR+2 ; SUBSTITUE CSR TEXT...
        MOV     #174000,GETCSR+6
        MOV     #177770,GETCSR+10 ; ...AND VALID RANGE.
        MOV     #4$,GETVEC+2 ; SUBSTITUTE VECTOR TEXT...
        CLR     GETVEC+6
        MOV     #370,GETVEC+10 ; ...AND VALID RANGE.
        RETURN          ; RETURN TO CALLER.
:
3$:     .ASCIZ <200>'1ST CSR ADDRESS (174000:177770) '
:
4$:     .ASCIZ <200>'1ST VECTOR ADDRESS (000:370) '
:
        .EVEN
$FREE= <1000-.>/2 ; FREE WORDS LEFT.
        .=$$VPC
CORMAX:
.END

```

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 14
 CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

ABASE = 160010	8387#	8730	9416
ABORT 006610	8731#		
ABORT1 006644	8731#		
ABORT2 006660	8731#		
ACDW1 = 000017	8392#	8730	
ACDW2 = 000000	8730		
ACPUOP= 000000	8730		
ACTIVE 001420	8730#*	8731*	
ADDW0 = 017470	8390#	8730	
ADDW1 = 017470	8390#	8730	
ADDW10= 017470	8390#	8730	
ADDW11= 017470	8390#	8730	
ADDW12= 017470	8390#	8730	
ADDW13= 017470	8390#	8730	
ADDW14= 017470	8390#	8730	
ADDW15= 017470	8390#	8730	
ADDW2 = 017470	8390#	8730	
ADDW3 = 017470	8390#	8730	
ADDW4 = 017470	8390#	8730	
ADDW5 = 017470	8390#	8730	
ADDW6 = 017470	8390#	8730	
ADDW7 = 017470	8390#	8730	
ADDW8 = 017470	8390#	8730	
ADDW9 = 017470	8390#	8730	
ADEVCT= 000000	8730		
ADEVN = 000001	8393#	8730	
ADRCNT 006125	8731#*		
ADVANC= 104400	8730#	8731	8736 8810
AENV = 000000	8730		
AENVN = 000000	8730		
AFATAL= 000000	8730		
AMADR1= 000000	8730		
AMADR2= 000000	8730		
AMADR3= 000000	8730		
AMADR4= 000000	8730		
AMAMS1= 000000	8730		
AMAMS2= 000000	8730		
AMAMS3= 000000	8730		
AMAMS4= 000000	8730		
AMSGAD= 000000	8730		
AMSGLG= 000000	8730		
AMSGTY= 000000	8730		
AMTYP1= 000000	8730		
AMTYP2= 000000	8730		
AMTYP3= 000000	8730		
AMTYP4= 000000	8730		
APASS = 000000	8730		
APRIOR= 000000	8730		
APTCSU= 000040	8731#		
APTENV= 000001	8731#		
APTSIZ= 000200	8731#		
APTSPO= 000100	8731#		
ASWREG= 000000	8730		
ATESTN= 000000	8730		
AUNIT = 000000	8730		
AUSWR = 000000	8730		

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 14-1
 CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

AUTO.S	011562	8730	8731#														
AVECT1=	000300	8391#	8730	9419													
AVECT2=	000000	8730															
BINWRD	006400	8731#															
BIT0	= 000001	8730#	8824														
BIT00	= 000001	8730#															
BIT01	= 000002	8730#															
BIT02	= 000004	8730#															
BIT03	= 000010	8730#															
BIT04	= 000020	8730#															
BIT05	= 000040	8730#															
BIT06	= 000100	8730#															
BIT07	= 000200	8730#															
BIT08	= 000400	8730#															
BIT09	= 001000	8730#															
BIT1	= 000002	8730#															
BIT10	= 002000	8730#	8793														
BIT11	= 004000	8730#	8731	8793													
BIT12	= 010000	8730#															
BIT13	= 020000	8730#															
BIT14	= 040000	8730#	8731														
BIT15	= 100000	8730#															
BIT2	= 000004	8730#															
BIT3	= 000010	8730#															
BIT4	= 000020	8730#	8731														
BIT5	= 000040	8730#	8731														
BIT6	= 000100	8730#															
BIT7	= 000200	8730#	8731														
BIT8	= 000400	8730#	8755														
BIT9	= 001000	8730#	8755														
BPTVEC=	000014	8730#															
BRK0	= 000400	8730#															
BRK1	= 001000	8730#															
BRK2	= 002000	8730#															
BRK3	= 004000	8730#															
BRW	004722	8730	8731#														
BUFSET=	104422	8730#	9010														
CHRCNT	006376	8731#*															
CLEAR	= 000000	8730#	9100*	9154*													
CNVRT	= 104412	8730#	8731														
CONVRT=	104411	8730#	8731														
CORMAX	017542	9440#	9441														
COO	= 000400	8730#															
CO1	= 001000	8730#															
CO2	= 002000	8730#															
CO3	= 004000	8730#															
CR	= 000015	8730#	8731														
CRLF	= 000200	8730#	8731														
CSRMAP	011570	8731#															
CYCLE	010722	8730	8731#														
DATABP	007150	8731#*															
DATAHD	007136	8731#*															
DCLASH=	104417	8730#	8862	8928	9007	9073	9100	9112									
DCLR	= 000020	8730#	8731	8744	8765	8777	8784	8786									
DDISP	= 177570	8730#															
DELAY	= 104414	8730#	8755	8833	8841	8850	8878	8894	8941	8969	8984	9025	9040	9087			

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 14-7
 CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

STOP	001446	8730#	8731				
SV05	006134	8731#					
SWR	001304	8730#*	8731*				
SWREG	000176	8730#					
SW0	= 000001	8730#					
SW00	= 000001	8730#					
SW01	= 000002	8730#	8731				
SW02	= 000004	8730#					
SW03	= 000010	8730#					
SW04	= 000020	8730#					
SW05	= 000040	8730#					
SW06	= 000100	8730#					
SW07	= 000200	8730#					
SW08	= 000400	8730#	8731				
SW09	= 001000	8730#	8731				
SW1	= 000002	8730#					
SW10	= 002000	8730#	8731				
SW11	= 004000	8730#					
SW12	= 010000	8730#	8731				
SW13	= 020000	8730#	8731				
SW14	= 040000	8730#					
SW15	= 100000	8730#					
SW2	= 000004	8730#					
SW3	= 000010	8730#					
SW4	= 000020	8730#					
SW5	= 000040	8730#					
SW6	= 000100	8730#					
SW7	= 000200	8730#					
SW8	= 000400	8730#					
SW9	= 001000	8730#					
S110	= 001000	8730#					
S1200	= 003400	8730#					
S134	= 001400	8730#					
S150	= 002000	8730#					
S1800	= 004000	8730#					
S19200	= 007400	8730#					
S2000	= 004400	8730#					
S2400	= 005000	8730#					
S300	= 002400	8730#					
S3600	= 005400	8730#					
S4800	= 006000	8730#					
S50	= 000000	8730#					
S600	= 003000	8730#					
S7200	= 006400	8730#					
S75	= 000400	8730#					
S9600	= 007000	8730#					
TBITVE	= 000014	8730#					
TCR0	= 000001	8730#	8754				
TCR1	= 000002	8730#	8754				
TCR2	= 000004	8730#	8754				
TCR3	= 000010	8730#	8754				
TD0	001426	8730#	8731	9031	9035	9049*	9100
TD1	001430	8730#					
TD2	001432	8730#					
TD3	001434	8730#					
TEIGHT	002106	8730#					

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 14-8
 CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

TFIVE	002114	8730#				
TIE =	040000	8730#	8753	8766	9100	9152
TKVEC =	000060	8730#				
TLAST =	015740	8731	9314#			
TLO =	000000	8730#				
TL1 =	000400	8730#				
TL2 =	001000	8730#				
TL3 =	001400	8730#				
TMTBL	002050	8730#				
TPVEC =	000064	8730#				
TRAPVE =	000034	8730#				
TRDY =	100000	8730#	8755	8850		
TRTVEC =	000014	8730#				
TRO	001436	8730#				
TR1	001440	8730#				
TR2	001442	8730#				
TR3	001444	8730#				
TSEVEN	002110	8730#				
TSIX	002112	8730#				
TST1	012272	8731	8736#			
TST10	013426	8793	8794#			
TST11	013512	8794	8807#			
TST12	013706	8807	8850#			
TST13	014040	8850	8861#			
TST14	014330	8861	8925#			
TST15	014652	8925	9006#			
TST16	015142	9006	9067#			
TST17	015344	9067	9100#			
TST2	012462	8736	8742#			
TST20	015740	9100	9111#	9314		
TST21 =	***** U	9111				
TST3	012526	8742	8753#			
TST4	012704	8753	8754#			
TST5	013110	8754	8755#			
TST6	013212	8755	8762#			
TST7	013342	8762	8793#			
TTST	004474	8730*	8731#			
TWOSTO =	000040	8730#				
TYPDAT	007140	8731#				
TYPE =	104402	8730#	8731			
TYPMSG	007030	8731#				
T110	002054	8730#				
T1200	002066	8730#				
T134	002056	8730#				
T150	002060	8730#				
T1800	002070	8730#				
T2000	002072	8730#				
T2400	002074	8730#				
T300	002062	8730#				
T3600	002076	8730#				
T4800	002100	8730#				
T50	002050	8730#				
T600	002064	8730#				
T7200	002102	8730#				
T75	002052	8730#				
T9600	002104	8730#				

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 14-9
 CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

UFD = 000040	8731#			
UFDSET= 000001	8730#			
VECMAP 012102	8731#			
VMKOR 002156	8730#			
WRDCNT 006374	8731#*			
WTBS.F 007126	8731#			
XBX 006716	8731#			
XCSR 004432	8731#			
XERR 004454	8731#			
XHEAD 010320	8730	8731#		
XMTCNT 001400	8730#			
XMTLIN 001376	8730#			
XPASS 004446	8731#			
XSTATQ 010410	8730	8731#		
XTSTN 007312	8731#			
XVEC 004440	8731#			
XX = 160210	8730#			
YY = 000500	8730#			
ZZ = 000020	8730#			
SAPTHD 001446	8730#			
SASTAT= ***** U	8731			
SATYC 005360	8731#			
SATY1 005334	8731#			
SATY3 005342	8731#			
SATY4 005352	8731#			
SAUTOB 001300	8730#			
SBASE 001174	8730#*	8731*	9416	9418*
SBDADR 001266	8730#			
SBDDAT 001272	8730#			
SCDW1 001200	8730#	8731		
SCDW2 001202	8730#	8731*		
SCHARC 005330	8731#*			
SCMTAG 001244	8730#			
SCM1 = 000006	8730#			
SCM2 = 000014	8730#			
SCM3 = 000006	8730#			
SCM4 = 000005	8730#			
SCPUOP 001146	8730#			
SCRLF 001357	8730#	8731		
SDDW0 001204	8730#	8731		
SDDW1 001206	8730#			
SDDW10 001230	8730#			
SDDW11 001232	8730#			
SDDW12 001234	8730#			
SDDW13 001236	8730#			
SDDW14 001240	8730#			
SDDW15 001242	8730#			
SDDW2 001210	8730#			
SDDW3 001212	8730#			
SDDW4 001214	8730#			
SDDW5 001216	8730#			
SDDW6 001220	8730#			
SDDW7 001222	8730#			
SDDW8 001224	8730#			
SDDW9 001226	8730#			
SDEVCT 001130	8730#	8731*		

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 15-1
 CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- MACRO NAMES

\$UNIBU	8395#	8736													
\$VARIA	6566#	8730													
\$XZ	7941#	8736	8737	8741	8753	8754	8755	8756	8761	8793	8794	8796	8805	8850	8851
	8860	8918	8924	9000	9005	9060	9066	9100	9102	9110					
\$SCMRE	8730#														
\$SCMTM	8730#														
\$SESCA	1697#	8730#													
\$SNEWT	1652#	8730#	8736	8742	8753	8754	8755	8762	8793	8794	8807	8850	8861	8925	9006
	9067	9100	9111												
\$SSKIP	1730#	8730#													
.EQUAT	191#	6518#	8730												
.HEADE	67#	6517#	8730												
.KT11	333#														
.SETUP	1213#	6517#													
.SWRHI	108#														
.SACT1	5090#	6519#	8730												
.SAPTB	5133#	6519#	8730#												
.SAPTH	5388#	6519#	8730												
.SAPTY	5560#	6519#	8731												
.SASTA	5433#														
.SCATC	917#	6517#													
.SCMTA	1026#	8730#													
.SDB2D	4726#														
.SDB2O	4847#														
.SDIV	4630#														
.SEOP	2185#	6517#	8731												
.SERRO	2664#	6518#													
.SERRT	2919#														
.SMULT	4568#														
.SPOWE	4244#	6518#	8731												
.SRAND	4318#														
.SRDDE	3918#														
.SRDOC	3828#														
.SREAD	3427#														
.SR2AZ	4988#														
.SSAVE	3992#														
.SSB2D	4809#														
.SSB2O	4908#														
.SSCOP	2419#	6518#	8731												
.SSIZE	4370#														
.SSUPR	4945#														
.STRAP	4093#	6518#													
.STYPB	3321#														
.STYPD	3245#														
.STYPE	3005#	6517#	8731												
.STYPO	3150#														
.S4OCA	955#														
.1170	511#														

. ABS. 017542 000

ERRORS DETECTED: 0

CVDZAD,CVDZAD/NL:TOC/CRF=SYSMC.SML,CVDZAD.P11

CVDZA-D MACY11 30(1046) 11-MAR-83 10:13 PAGE 15-2
CVDZAD.P11 11-MAR-83 10:06 CROSS REFERENCE TABLE -- MACRO NAMES

RUN-TIME: 16 20 1 SECONDS
RUN-TIME RATIO: 415/38=10.9
CORE USED: 54K (108 PAGES)