IDENTIFICATION

PRODUCT CODE:	AC-A938D-MC
PRODUCT NAME:	CVDZBDO DZV11 DIAG PRT2
DATE RELEASED:	MARCH 1983
MAINTAINER:	DIAGNOSTIC ENGINEERING

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 (DVDZA,DVDZB,AND DVDZC) ONE SYSTEM MODULE FOR DEC X/11 (DZBA), AND AN OVERLAY FOR ITEP (DVDZD).

DVDZA TOGETHER WITH DVDZB WILL TEST ALL LOGICAL FUNCTIONS OF THE DZV11 INTERFACE MODULE.
DVDZC IS DESIGNED AS A NON-CHAINABLE STANDALONE DIAGNOSTIC PROVIDING THE OPERATOR WITH DIRECT CONTROL OVER THE TESTING OF ALL DZV11 EIA CABLES.

```
*****************************************************************
*                                                         ;;GPA *
* 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 *
*****************************************************************
```

1.1 HISTORY

REVISION C OF THIS DIAGNOSTIC CVDZB WAS DONE TO MAKE THIS DIAGNOSTIC USER FRIENDLY COMPATIBLE.

REVISION D WAS DONE ON 29-JUL-84 TO INCREASE TIMING PARAMETERS TO ALLOW PROGRAM TO RUN ON A J-11 PROCESSOR (ORION).
(LOCATION 15172 WAS = 2, IT WAS CHANGED TO 4).

2. REQUIREMENTS

2.1 EQUIPMENT

```
AN LSI11 CPU WITH MINIMUM 4K OF MEMORY.
ASR 33 (OR EQUIVALENT FOR CONSOLE)
DZV11           INTERFACE MODULE
M329            STAGGERED TURNAROUND CONNECTOR.
M325            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 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 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.
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 'SCOP1' 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 'SCOP1'). 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'.

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.
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, 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

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 DVDZB-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

$LPADR (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 PEFORMED.
$TSTNM (1246)	CONTAINS THE NUMBER OF THE TEST NOW BEING PEFORMED.
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.
$BASE (1174)	CONTAINS THE RECEIVER CSR OF THE CURRENT DZV11 UNDER TEST.

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 OCCURES, 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 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 INTERUPT ENABLE AND MSTSCAN ENABLE) ARE SET INTO THE DZVCSR. ALL TCR BITS ARE SET, A DELAY OCCURS, AND IF NO INTERUPT OCCURES (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 INTERUPT OCCURRED, THE ADDRESS TO WHICH THE DZV11 INTERUPTED 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 (DVDZA, AND DVDZB) CAN BE RUN AS STANDALONE DIAGNOSTICS OR IN EITHER OF THE APT MODES. DVDZC, 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:

$SWREG	-(1142)	USED AS THE SOFTWARE SWITCH REGISTER WHILE RUNNING UNDER APT.
$VECT1	-(1170)	USED TO SPECIFY THE FIRST VECTOR ADDRESS
$BASE	-(1174)	USED TO INDICATE BOTTOM ADDRESS OF DZV11 UNDER TEST
$DEVM	-(1176)	A BIT MAP REPRESENTING WHICH DZV11'S WILL BE TESTED
$CDW1	-(1200)	USED TO INDICATE WHICH LINES TO RUN ON ALL DZV11'S
$CDW2	-(1202)	USED TO INDICATE THE DEFAULT TEST MODE. SET TO 0 FOR INTERNAL TESTING, 200 FOR EXTERNAL LOOP BACK (H325 INSTALLED), OR SET TO 100000 FOR STAGGERED LOOP BACK TESTING (H329 INSTALLED).
$DDW0	-(1204)	EACH OF THE $DDW WORDS DESCRIBES THE PARAMETERS (LPR) FOR A PARTICULAR DZV11, GOING UP TO 16 DZV11'S

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 $BASE 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), JAN 19, 1977.

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

51              MISCELLANEOUS DEFINITIONS

63              GENERAL PURPOSE REGISTER DEFINITIONS

75              PRIORITY LEVEL DEFINITIONS

85              "SWITCH REGISTER" SWITCH DEFINITIONS

113             DATA BIT DEFINITIONS (BIT00 TO BIT15)

141             BASIC "CPU" TRAP VECTOR ADDRESSES

358                                     BITS 15-11=CPU TYPE
                                                11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                                11/70=06,PDQ=07,Q=10
                                        BIT 10=REAL TIME CLOCK
                                        BIT  9=FLOATING POINT PROCESSOR
                                        BIT  8=MEMORY MANAGEMENT

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

371                                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" A80

410             THIS TABLE CONTAINS .ARIOUS COMMON STORAGE LOCATIONS
                USED IN THE PROGRAM.

462             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 IS PERTINENT.
                NOTE1:  IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                NOTE2:  EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

468                     EM              ;;POINTS TO THE ERROR MESSAGE
                        DH              ;;POINTS TO THE DATA HEADER
                        DT              ;;POINTS TO THE DATA
                        DF              ;;POINTS TO THE DATA FORMAT
```

```
1010    INCREMENT THE PASS NUMBER (@PASS)
        IF THERES A MONITOR GO TO IT
        IF THERE ISN'T JUMP TO CYCLE

1072    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

1147    ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
        THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
        NOTE1:  @NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
        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 ASCIZ STRING
        OR
                TYPE
                MESADR


1931    ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
        IF BIT7 IN THE ENVIRONMENT MODE (@ENVM) BYTE IS SET,
        THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.

1963    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)


2071    ********************** TEST 1 ******************************
        THIS TEST VERIFIES OVERRUN AND SILO ALARM
        ONE LINE AT A TIME  - BASED UPON VALID LINES
        AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
        TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
        EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
        SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
        CHAR PULLED OUT OF THE SILO.
        ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.
        USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
        ON DZV LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
        USED TO SCOPE SILO ALARM PULSES, ETC.
```

```
2192    ********************** TEST 2 ******************************
        THIS TEST THAT "SILO ENABLE" WILL INHIBIT
        RECEIVER INTERRUPTS AND THAT ON THE
        16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
        INTERRUPT WITH "RIE" SET.
        THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
        ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.

2264    ********************** TEST 3 ******************************
        THIS TEST RUNS ALL LINES FULL BORE
        BASED UPON QUALIFIED LINES
        ..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
        TRANSMITTER

2397    ********************** TEST 4 ******************************
        DZV11 RELATIVE TIMING TEST.
        EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
        AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
        WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
        DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
        THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
         AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
        PARAMETERS ARE:
          EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                2400, 3600, 4800, 7200, 9600 BAUD.
          19.2 K BAUD - TWO STOP BITS AT
                SEVEN, SIX, FIVE BITS/PER/CHAR.
        AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
        THE NEXT SELECTED LINE IS THEN TESTED.
        WHEN RUNNING UNDER THE APT MANUFACTURING SYSTEM
        THIS TEST IS ONLY RUN THE FIRST PASS

2491    ********************** TEST 5 ******************************
        THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
        THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
        THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
        YOU ARE IN "STAGGERED" MODE.
        40(8) CHARS ARE USED FOR THIS TEST.
        ALL SELECTED LINES WILL BE ENABLED AT THE SAME TIME.
        THIS TEST FIRST CHECKS EVEN PARITY FOR ODD LINES AND
        ODD PARITY FOR EVEN LINES, THEN IT CHECKS THE REVERSE.
```

```
 6486
 8456
  (2)                    ;*COPYRIGHT (C) 1977,1984
  (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)                            ;"CVDZB0/<200>/FOUR LINE ASYNC MUX TESTS, PART 2 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
  (2)
  (1)                    .REM    !
  (1)                    ;SWITCH REGISTER OPTIONS
  (1)                    ;----------------------
  (1)
  (1)
  (1)                    SW15=100000             ;=1,HALT ON ERROR
  (1)                    SW14=40000              ;=1,LOOP ON CURRENT TEST
  (1)                    SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
  (1)                    SW12=10000              ;=1,DELETE TYPEOUT/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 INITAL START UP.
  (1)                    SW06=100                ;=1, DESELECT SPECIFIC DEVICES
  (1)                                            ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
  (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)          000C04    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
(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)
(2)                     ;*GENERAL PURPOSE REGISTER DEFINITIONS
(2)     000000          R0=     %0              ;;GENERAL REGISTER
(2)     000001          R1=     %1              ;;GENERAL REGISTER
(2)     000002          R2=     %2              ;;GENERAL REGISTER
(2)     000003          R3=     %3              ;;GENERAL REGISTER
(2)     000004          R4=     %4              ;;GENERAL REGISTER
(2)     000005          R5=     %5              ;;GENERAL REGISTER
(2)     000006          R6=     %6              ;;GENERAL REGISTER
(2)     000007          R7=     %7              ;;GENERAL REGISTER
(2)     000006          SP=     %6              ;;STACK POINTER
(2)     000007          PC=     %7              ;;PROGRAM COUNTER
(2)
(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)
(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
```

```
CVDZB-D MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-1
CVDZBD.P11    21-AUG-84 08:28          BASIC DEFINITIONS

   (2)
   (2)                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
   (2)          100000                BIT15=  100000
   (2)          040000                BIT14=  40000
   (2)          020000                BIT13=  20000
   (2)          010000                BIT12=  10000
   (2)          004000                BIT11=  4000
   (2)          002000                BIT10=  2000
   (2)          001000                BIT09=  1000
   (2)          000400                BIT08=  400
   (2)          000200                BIT07=  200
   (2)          000100                BIT06=  100
   (2)          000040                BIT05=  40
   (2)          000020                BIT04=  20
   (2)          000010                BIT03=  10
   (2)          000004                BIT02=  4
   (2)          000002                BIT01=  2
   (2)          000001                BIT00=  1
   (2)          001000                BIT9=   BIT09
   (2)          000400                BIT8=   BIT08
   (2)          000200                BIT7=   BIT07
   (2)          000100                BIT6=   BIT06
   (2)          000040                BIT5=   BIT05
   (2)          000020                BIT4=   BIT04
   (2)          000010                BIT3=   BIT03
   (2)          000004                BIT2=   BIT02
   (2)          000002                BIT1=   BIT01
   (2)          000001                BIT0=   BIT00
   (2)
   (2)                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
   (2)          000004                ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
   (2)          000010                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)
   (1)                                ;INSTRUCTION DEFINITIONS
   (1)                                ;----------------------
   (1)
   (1)          005746                PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
   (1)          005726                POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
   (1)          010046                PUSHR0=10046    ;SAVE R0 ON STACK
   (1)          012600                POPR0=12600     ;RESTORE R0 FROM STACK
   (1)          024646                PUSH2SP=24646   ;DECREMENT STACK TWICE
   (1)          022626                POP2SP=22626    ;INCREMENT STACK TWICE
   (1)          000200                MASK=BIT7       ;SET INTERRUPT MASK (INHIBIT FURTHER INTERRUPTS)
   (1)          000000                CLEAR=0         ;ALLOW INTERRUPTS (CLEAR PROCESSOR STATUS)
   (1)
```

```
(1)
(1)                                     ;DZV11 CONTROL AND STATUS REGISTER DEFINITIONS
(1)                                     ;(DZVCSR)       BIT DEFINITIONS
(1)                                     ;-----------------------------
(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)
(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)
(1)
(1)                                     ;DZVRBUF BIT DEFINITIONS
(1)                                     ;----------------------
(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)
(1)                                     ;DZVRBUF WORD DEFINITIONS
(1)                                     ;----------------------
(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)
(1)                                     ;DZVLPR WORD DEFINITIONS
(1)                                     ;----------------------
(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)
(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)
```

```
(1)  000100                PARITY=BIT6             ;PARITY ENABLED
(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)
(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)
(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)                                     ;TABLE OF LOOP AROUND FUNCTIONS (M325)
(1)                                     ;
(1)                                     ;       ----------------
(1)                                     ;       I               ↑
(1)                                     ;       V               ↑
(1)                                     ;       REC             TRANS
(1)                                     ;       DATA            DATA
(1)                                     ;
(1)                                     ;       ----------------
(1)                                     ;       I               ↑
(1)                                     ;       V               ↑
(1)                                     ;       CO              RTS
(1)                                     ;
(1)                                     ;       ----------------
(1)                                     ;       I               ↑
(1)                                     ;       V               ↑
(1)                                     ;       RING            DTR
```

```
(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 HALT".
(1)                                  ;---------------------------------------------------  -------------  --
(1)                                  ;;*****************************************************************
(1)
(1)          000000                  .=0
(1)                                  ;MUST NOT SET UP EMT TRAP CATCHER SINCE IT IS USED FOR UFD MONITOR LINKAGE
(1)          000034                  .=34                     ;SKIP OVER EMT TRAP
(1)                                          ;STANDARD INTERRUPT VECTORS
(1)                                          ;------------- -----------
(1)
(1)          000020                  .=20
(1)  000020  004464                          .SCOPE                    ;SCOPE LOOP HANDLER
(1)  000022  000200                          MASK                      ;HANDLE AT PRIORITY 7
(1)  000024  007566                          $PWRDN                    ;POWER FAIL HANDLER
(1)  000026  000340                          340                       ;SERVICE AT PRIORITY LEVEL 7
(1)          000034                  .=34
(1)  000034  006404                          .TRPSRV                   ;GENERAL HANDLER DISPATCH SERVICE
(1)  000036  000340                          340                       ;SERVICE AT PRIORITY LEVEL 7
(2)                                  .SBTTL  ACT11 HOOKS
(2)
(3)                                  ;;*****  ********************************************************
(2)                                  ;HOOKS REQUIRED BY ACT11
(2)          000040                          $SVPC=.                   ;SAVE PC
(2)          000046                          .=46
(2)  000046  004420                          $ENDAD                    ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
(2)          000052                          .=52
(2)  000052  000000                          .WORD   0                 ;;2)SET LOC.52 TO ZERO
(2)          000040                          .=$SVPC                   ;; RESTORE PC
(1)
(1)          000174                  .=174
(1)  000174  000000                  DISPREG:0                         ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
(1)  000176  000000                  SWREG:  0                         ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
(1)          000200                  .=200
(1)  000200  000137  002116                  JMP     .START            ;GO TO START OF PROGRAM
(1)
(2)
(2)          001000                  .=1000
(2)  001000  005200  053103  055104  MTITLE: .ASCIZ  <200><12>/CVDZBD/<200>/FOUR LINE ASYNC MUX TESTS, PART 2 OF 2/<200>
(2)
```

```
CVDZB-D MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-6
CVDZBD.P11    21-AUG-84 08:28          PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

   (3)          001120                          .=1120
   (4)                                  ;;*****************************************************************
   (4)                                  .SBTTL  APT MAILBOX-ETABLE
   (4)
   (5)                                  ;;*****************************************************************
   (4)                                  .EVEN
   (4)  001120                          $MAIL:                          ;;APT MAILBOX
   (4)  001120  000000                  $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
   (4)  001122  000000                  $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
   (4)  001124  000000                  $TESTN: .WORD   ATESTN   ;;TEST NUMBER
   (4)  001126  000000                  $PASS:  .WORD   APASS    ;;PASS COUNT
   (4)  001130  000000                  $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
   (4)  001132  000000                  $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
   (4)  001134  000000                  $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
   (4)  001136  000000                  $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
   (4)  001140                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
   (4)  001140     000                  $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
   (4)  001141     000                  $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
   (4)  001142  000000                  $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
   (4)  001144  000000                  $USWR:  .WORD   AUSWR    ;;USER SWITCHES
   (4)  001146  000000                  $CPUOP: .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                  $MAMS1: .BYTE   AMAMS1   ;;HIGH ADDRESS,M.S. BYTE
   (4)  001151     000                  $MTYP1: .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                  $MADR1: .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
   (4)                                  ;*                              MEM.LAST ADDR.-3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
   (4)  001154     000                  $MAMS2: .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S. BYTE
   (4)  001155     000                  $MTYP2: .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
   (4)  001156  000000                  $MADR2: .WORD   AMADR2   ;;MEM.LAST ADDRESS,BLK#2
   (4)  001160     000                  $MAMS3: .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
   (4)  001161     000                  $MTYP3: .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
   (4)  001162  000000                  $MADR3: .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
   (4)  001164     000                  $MAMS4: .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
   (4)  001165     000                  $MTYP4: .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
   (4)  001166  000000                  $MADR4: .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
   (4)  001170  000300                  $VECT1: .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
   (4)  001172  000000                  $VECT2: .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
   (4)  001174  160010                  $BASE:  .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
   (4)  001176  000001                  $DEVM:  .WORD   ADEVM    ;;DEVICE MAP
   (4)  001200  000017                  $CDW1:  .WORD   ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
   (4)  001202  000000                  $CDW2:  .WORD   ACDW2    ;;CONTROLLER DESCRIPTION WORD#2
   (4)  001204  017470                  $DDW0:  .WORD   ADDW0    ;;DEVICE DESCRIPTOR WORD#0
   (4)  001206  017470                  $DDW1:  .WORD   ADDW1    ;;DEVICE DESCRIPTOR WORD#1
   (4)  001210  017470                  $DDW2:  .WORD   ADDW2    ;;DEVICE DESCRIPTOR WORD#2
   (4)  001212  017470                  $DDW3:  .WORD   ADDW3    ;;DEVICE DESCRIPTOR WORD#3
   (4)  001214  017470                  $DDW4:  .WORD   ADDW4    ;;DEVICE DESCRIPTOR WORD#4
   (4)  001216  017470                  $DDW5:  .WORD   ADDW5    ;;DEVICE DESCRIPTOR WORD#5
```

```
(4)  001220  017470                    $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
(4)  001222  017470                    $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
(4)  001224  017470                    $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
(4)  001226  017470                    $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
(4)  001230  017470                    $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
(4)  001232  017470                    $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
(4)  001234  017470                    $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
(4)  001236  017470                    $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
(4)  001240  017470                    $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
(4)  001242  017470                    $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
(4)
(4)
(4)  001244                            $ETEND:
(4)
```

```
(3)                                     .SBTTL  COMMON TAGS
(3)
(4)                                     ;;*****************************************************************
(3)                                     ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(3)                                     ;*USED IN THE PROGRAM.
(3)
(3)  001244                             $CMTAG:                         ;;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     $REGAD: .WORD   0       ;;CONTAING THE ADDRESS FROM
(3)                                             ;;WHICH  ($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:    .ASCIZ  <12>    ;;LINE FEED
```

```
(3)                                     .SBTTL  ERROR POINTER TABLE
(3)
(3)                                     ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(3)                                     ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(3)                                     ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(3)                                     ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(3)                                     ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(3)
(3)                                     ;*      EM              ;;POINTS TO THE ERROR MESSAGE
(3)                                     ;*      DH              ;;POINTS TO THE DATA HEADER
(3)                                     ;*      DT              ;;POINTS TO THE DATA
(3)                                     ;*      DF              ;;POINTS TO THE DATA FORMAT
(3)
(3)
(3)  001362                             $ERRTB:
(2)
(2)                                             ;PROGRAM CONTROL PARAMETERS
(2)                                             ;-------------------------
(2)
(2)  001362  000000                     NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
(2)  001364  000000                     LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT TEST,TIGHT LOOP
(2)
(2)                                             ;PROGRAM VARIABLES
(2)                                             ;-----------------
(2)
(2)  001366  000017                     LINE:   17                      ;DEFAULT ALL FOUR LINES RUNNING
(2)  001370  017470                     PAR:    17470                   ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,NO PARIT
(2)  001372  000000                     MODE:   0                       ;DEFAULT MAINTENANCE MODE
(2)  001374  000000                     SAVLIN: 0                       ;LINE NUMBER
(2)  001376  000000                     XMTLIN: 0                       ;TRANSMISSION LINE NUMBER
(2)  001400  000000                     XMTCNT: 0                       ;COUNT OF WORDS IN A TRANSMISSION PATTERN
(2)  001402  000000                     REGIST: 0                       ;DEVICE ADDRESS STORAGE LOCATION
(2)  001404  000000                     SAVPC:  0                       ;PROGRAM COUNTER STORAGE
(2)  001406  000001                     DZVACTV:.BLKW   1               ;*DZV11'S SELECTED ACTIVE.
(2)  001410  000001                     SAVACTV:.BLKW   1               ;*A BIT MAP OF DZV11'S IN THE SYSTEM
(2)  001412  000001                     RUN:    1                       ;*POINTER ONE PAST RUNNING DEVICE.
(2)  001414  000001                     DZVNUM: .BLKB 1                 ;*OCTAL NUMBER OF DZV11'S IN THE SYSTEM.
(2)  001415     001                     SAVNUM: .BYTE 1                 ;*WORKABLE NUMBER.
(2)  001416  000001                     SAVNO:  .BLKB   1               ;*OCTAL NUMBER OF DZV11'S BEING TESTED
(2)          001420                     .EVEN
(2)  001420  001500                     ACTIVE: DZV.MAP                 ;TABLE POINTER.
```

```
(2)
(2)                                     ;PROGRAM CONTROL FLAGS
(2)                                     ;---------------------
(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                             STOP:
(2)                                     .SBTTL  APT PARAMETER BLOCK
(2)
(3)                                     ;;*****************************************************************
(2)                                     ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(3)                                     ;;*****************************************************************
(2)          001446                             .$X=.   ;;SAVE CURRENT LOCATION
(2)          000024                             .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
(2)  000024  000200                             200     ;;FOR APT START UP
(2)          000044                             .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
(2)  000044  001446                             $APTHDR ;;POINT TO APT HEADER BLOCK
(2)          001446                             .=.$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                     $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
(2)  001452  000132                     $TSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
(2)  001454  000137                     $PASTM: .WORD   95.     ;;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   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
(1)                                     ;DZV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
(1)                                     ;-----------------------------------------
(1)
(1)          001500                     .=1500
(1)  001500                             DZV.MAP:
(3)
(3)  001500  000001                     DZCR0:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 0
(3)  001502  000001                     DZVC0:  .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
```

```
CVDZB-D MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-11
CVDZBD.P11    21-AUG-84 08:28          APT PARAMETER BLOCK

   (3)  001520  000001                PAR1:    .BLKW   1       ;PARAMETERS
   (3)  001522  000001                MAINT1:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT2:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT3:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT4:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT5:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT6:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT7:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT10: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT11: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
   (3)
   (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                MAINT12: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
```

```
(3)
(3)  001656  000001                DZCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 13
(3)  001660  000001                DZVC13: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 13
(3)  001662  000001                LINE13: .BLKW   1       ;ALL LINES SELECTED
(3)  001664  000001                PAR13:  .BLKW   1       ;PARAMETERS
(3)  001666  0000C1                MANT13: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
(3)
(3)  001670  000001                DZCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 14
(3)  001672  000001                DZVC14: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 14
(3)  001674  000001                LINE14: .BLKW   1       ;ALL LINES SELECTED
(3)  001676  000001                PAR14:  .BLKW   1       ;PARAMETERS
(3)  001700  000001                MANT14: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
(3)
(3)  001702  000001                DZCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 15
(3)  001704  000001                DZVC15: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 15
(3)  001706  000001                LINE15: .BLKW   1       ;ALL LINES SELECTED
(3)  001710  000001                PAR15:  .BLKW   1       ;PARAMETERS
(3)  001712  000001                MANT15: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
(3)
(3)  001714  000001                DZCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 16
(3)  001716  000001                DZVC16: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 16
(3)  001720  000001                LINE16: .BLKW   1       ;ALL LINES SELECTED
(3)  001722  000001                PAR16:  .BLKW   1       ;PARAMETERS
(3)  001724  000001                MANT16: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
(3)
(3)  001726  000001                DZCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 17
(3)  001730  000001                DZVC17: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 17
(3)  001732  000001                LINE17: .BLKW   1       ;ALL LINES SELECTED
(3)  001734  000001                PAR17:  .BLKW   1       ;PARAMETERS
(3)  001736  000001                MANT17: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
(1)
(1)  001740  177777                DZV.END:        177777
```

```
CVDZB 0 MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-13
CVDZB0.P11    21-AUG-84 08:28          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)                                    ;----------------------------------------------------------------------
(1)  001742                            .TRPTAB:
(3)          104400                    ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
(2)  001742  006500                            .ADVANCE
(3)          104401                    SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
(2)  001744  004730                            .SCOP1
(3)          104402                    TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
(2)  001746  004754                            .TYPE
(3)          104403                    INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
(2)  001750  005604                            .INSTR
(3)          104404                    INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
(2)  001752  005710                            .INSTER
(3)          104405                    PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
(2)  001754  005730                            .PARAM
(3)          104406                    SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
(2)  001756  010424                            .SETFLG
(3)          104407                    SAV05=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
(2)  001760  006130                            .SAV05
(3)          104410                    RES05=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
(2)  001762  006170                            .RES05
(3)          104411                    CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
(2)  001764  006222                            .CONVRT
(3)          104412                    CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
(2)  001766  006226                            .CNVRT
(3)          104413                    DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
(2)  001770  006426                            .DEVICE.CLR
(3)          104414                    DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
(2)  001772  006460                            .DELAY
(3)          104415                    PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
(2)  001774  011434                            .PARMD
(3)          104416                    PAUCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
(2)  001776  010544                            .PAUCH
(3)          104417                    DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
(2)  002000  006446                            .DCLASM
(3)          104420                    SHIFT=TRAP+20           ;CALL TO ROTATE LINE POINTER
(2)  002002  006512                            .SHIFT
(3)          104421                    LPRSET=TRAP+21          ;CALL TO SET UP LPR DEVICE REGISTER
(2)  002004  006530                            .LPRSET
(3)          104422                    BUFSET=TRAP+22          ;CALL TO ZERO BUFFER AREA
(2)  002006  006570                            .BUFSET
(1)
(1)                                    ;----------------------------------------------------------------------
(1)                                    ;:**********************************************************************
```

```
(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)
```

```
(1)
(1)                                         ;TIME TABLE FOR RELATIVE TIMING TESTS
(1)                                         ;---------- ----------------------
(1)
(1)  002050                                 TMTBL:
(1)  002050  000000                         T50:    0
(1)  002052  000000                         T75:    0
(1)  002054  000000                         T110:   0
(1)  002056  000000                         T134:   0
(1)  002060  000000                         T150:   0
(1)  002062  000000                         T300:   0
(1)  002064  000000                         T600:   0
(1)  002066  000000                         T1200:  0
(1)  002070  000000                         T1800:  0
(1)  002072  000000                         T2000:  0
(1)  002074  000000                         T2400:  0
(1)  002076  000000                         T3600:  0
(1)  002100  000000                         T4800:  0
(1)  002102  000000                         T7200:  0
(1)  002104  000000                         T9600:  0
(1)  002106  000000                         TEIGHT:0
(1)  002110  000000                         TSEVEN: 0
(1)  002112  000000                         TSIX:   0
(1)  002114  000000                         TFIVE:  0
```

```
(1)                                             ;PROGRAM INITIALIZATION
(1)                                             ;LOCK OUT INTERRUPTS
(1)                                             ;SET UP PROCESSOR STACK
(1)                                             ;SET UP POWER FAIL VECTOR
(1)                                             ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
(1)                                             ;TYPE TITLE MESSAGE
(1)          001401                     UFDSET = 1
(1)
(1)  002116  032737  000040  000052  .START: BIT     @#PTS,#052      ;ARE WE UNDER UFD ?
(1)  002124  001403                         BEQ     1$              ;NO,THEN SKIP THE INSTRUCTIONS
(1)  002126  104042                         EMT     42              ;GET DSRERR ADDRESS
(1)  002130  005060  000042                 CLR     42(R0)          ;INITIALIZE IT TO NO ERROR
(1)  002134                             1$:
(2)                                     ;;LCP/ORION ROUTINE TO SAVE EMTULATOR AND PRIORITY
(2)
(2)  002134  013737  000030  002152  EMTSAV: MOV     30,SAV30                ;;SAVE EMULATOR ADDRESS
(2)  002142  013737  000032  002154         MOV     32,SAV32                ;;SAVE EMULATOR PRIORITY LEVEL
(2)  002150  000402                         BR      VMKOR                   ;;GET AROUND TAG AREA
(2)  002152  000000                     SAV30: .WORD   0                       ;;PUT EMULATOR INFO HERE
(2)  002154  000000                     SAV32: .WORD   0                       ;;PUT PRIORITY LOCATION HERE
(2)  002156                             VMKOR:
(2)                                     ;*************************************************************************
(2)
(1)  002156  012737  006664  000030          MOV     #$ERROR,@#30    ;SET UP EMT VECTOR TO ERROR ROUTINE
(1)  002164  012737  000340  000032          MOV     #340,@#32       ;
(1)  002172  012706  001120                  MOV     #STACK,SP       ;SET UP STACK
(1)  002176  106427  000200                  MTPS    #MASK           ;LOCK OUT INTERRUPTS
(1)  002202  012737  007566  000024          MOV     #$PWRDN,@#24    ;SET UP POWER FAIL VECTOR
(1)  002210  005037  001126                  CLR     $PASS           ;CLEAR PASS COUNT
(1)  002214  105037  001247                  CLRB    $ERFLG          ;CLEAR ERROR FLAG
(1)  002220  012737  001500  001420          MOV     #DZV.MAP,ACTIVE ;GET MAP POINTER.
(1)  002226  012737  000001  001412          MOV     #1,RUN          ;POINT POINTER TO FIRST DEVICE.
(1)  002234  005037  001256                  CLR     $ERTTL          ;CLEAR ERROR COUNT
(1)  002240  005037  001262                  CLR     $ERRPC          ;CLEAR LAST ERROR POINTER
(1)  002244  005037  001246                  CLR     $TSTNM          ;SET UP FOR TEST 1
(1)  002250  012737  002116  001252          MOV     #.START,$LPADR  ;SET UP FOR POWER FAIL BEFORE
(1)                                                                  ;TESTING STARTS
(1)                                     ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
(1)  002256  012737  000176  001304          MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWR
(1)  002264  012737  000174  001306          MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
(1)  002272  004737  017332                  CALL    FALCON          ; CHECK FOR FALCON (KXT11)      ;;GPA
(1)  002276  001402                          BEQ     1000$           ; BR IF NOT                     ;;GPA
(1)  002300  004737  000570                  CALL    FALCINI         ; YES, INIT FOR FALCON.         ;;GPA
(1)  002304                             1000$:                                                       ;;GPA
(1)  002304  105737  001422                  TSTB    INIFLG          ;HAVE WE ALREADY BEEN HERE TODAY?
(1)  002310  001014                          BNE     10$             ;IF SO, SKIP PRINTING THE TITLE
(1)  002312  023727  000042  004420          CMP     @#42,@#ENDAD    ;IF RUNNING UNDER ACT
(1)  002320  001406                          BEQ     1$              ;DON'T PRINT TITLE
(1)  002322  032737  000040  000052          BIT     #BIT5,@#52      ;ARE WE UNDER UFD ?
(1)  002330  001002                          BNE     1$              ;YES,THEN SKIP TYPEOUT
(1)  002332  104402  001000                  TYPE    ,MYTITLE        ;PRINT THE DIAGNOSTIC'S TITLE
(1)  002336  105...7  001422             1$:  DECB    INIFLG          ;SET THE ONCE ONLY FLAG
(1)  002342  105737  001141             10$: TSTB    $ENVM           ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
(1)  002346  100004                          BPL     15$             ;IF NOT, GO CHECK FOR AUTO-SIZING
(1)  002350  004737  011436                  JSR     PC,SETAPT       ;OTHERWISE, GO DO APT SIZING FROM ETABLE
```

```
                        PROGRAM INITIALIZATION AND START UP.

(1)  002354  000137  003710                JMP     105$            ;GO PRINT DZV STATUS TABLE
(1)  002360  032737  000040  000052  15$:   BIT     #BIT5,@#52      ;ARE WE UNDER UFD ?
(1)  002366  001015                        BNE     17$             ;YES,THEN GO AUTOSIZE THE SETUP
(1)  002370  005737  000042                TST     @#42            ; CHAINED UNDER XXDP ??         ;;GPA
(1)  002374  001410                        BE      16$             ; BR IF NOT                     ;;GPA
(1)  002376  032737  000040  000052        BIT     #BIT5,@#52      ;ARE WE UNDER UFD ?
(1)  002404  001006                        BNE     17$             ;YES THEN GO CHECK THE SWITCHES
(1)  002406  004737  011436                CALL    SETAPT          ; YES, SET-UP FROM ETABLE       ;;GPA
(1)  002412  000137  003710                JMP     105$            ; AND PROCEED                   ;;GPA
(1)  002416  004737  007350          16$:  CALL    GETSWR          ; GET AN INITIAL SWR            ;;GPA
(1)  002422  032777  000001  176654  17$:  BIT     #SW00,@SWR      ;RESELECT ?
(1)  002430  001002                        BNE     20$             ;IF YES, GO SET UP THE INFORMATION
(1)  002432  000137  002734                JMP     55$             ;IF NO, SKIP THE INTERROGATION
(1)  002436  012700  001500          20$:  MOV     #DZV.MAP,R0     ;POINT TO THE BEGINNING OF THE MAP TABLE
(1)  002442  105037  001423                CLRB    MDRFLG          ;MAKE SURE A MAP GETS PRINTED
(1)  002446  005020                  25$:  CLR     (R0)+           ;CLEAR A TABLE LOCATION
(1)  002450  020027  001740                CMP     R0,#DZV.END     ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
(1)  002454  001374                        BNE     25$             ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
(1)  002456  105337  001422                DECB    INIFLG          ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED'
(1)
(1)                                 ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
(1)                                 ;TABLE AND SET UP THE DIAGNOSTIC.
(1)
(1)                                         ;GET THE BASE ADDRESS OF THE DZV11'S
(1)       002462                            GETCSR= .               ; POINTER FOR FALCON TWEAKER.   ;;GPA
(2)  002462  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
(2)  002464  003154                         91$                     ;POINTER TO MESSAGE TO BE PRINTED
(2)  002466  104405                         PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)  002470  160000                         160000                  ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002472  167770                         167770                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002474  001500                         DZCR0                   ;POINTER TO MAP LOCATION TO BE FILLED
(2)  002476     007                         .BYTE   7               ;MASK OF INVALID BITS FOR THIS PARAMETER
(2)  002477     001                         .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
(1)  002500  013737  001500  001174         MOV     DZCR0,#BASE     ;COPY BASE ADDRESS TO ETABLE
(1)
(1)                                         ;GET THE BASE VECTOR ADDRESS
(1)       002506                            GETVEC= .               ; POINTER FOR FALCON TWEAKER.   ;;GPA
(2)  002506  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
(2)  002510  003220                         92$                     ;POINTER TO MESSAGE TO BE PRINTED
(2)  002512  104405                         PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)  002514  000300                         300                     ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002516  000776                         776                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002520  001502                         DZVC0                   ;POINTER TO MAP LOCATION TO BE FILLED
(2)  002522     003                         .BYTE   3               ;MASK OF INVALID BITS FOR THIS PARAMETER
(2)  002523     001                         .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
(1)  002524  013737  001502  001170         MOV     DZVC0,#VECT1    ;COPY VECTOR TO ETABLE
(1)
(1)                                         ;GET THE MODE OF OPERATION (E,I,S)
(2)  002532  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
(2)  002534  003447                         96$                     ;POINTER TO THE MESSAGE TO BE PRINTED
(2)  002536  104406                         SETFLG                  ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
(2)  002540  001510                         MANTO                   ;THIS IS THE FLAG BEING SETUP
(1)
(1)                                         ;GET THE NUMBER OF DZV11'S RUNNING
(2)  002542  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
(2)  002544  003404                         95$                     ;POINTER TO MESSAGE TO BE PRINTED
```

```
(2)  002546  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)  002550  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002552  000020                          16.                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002554  001344                          $TMP1                   ;POINTER TO MAP LOCATION TO BE FILLED
(2)  002556     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
(2)  002557     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
(1)
(1)  002560  012737  000017  001504          MOV     #17,LINE0       ;SET UP DEFAULT LINES
(1)  002566  012737  017470  001506          MOV     #17470,PAR0     ;SET UP DEFAULT LPR PARAMETER
(1)                                                                  ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
(1)  002574  032777  000010  176502          BIT     #SW03,@SWR      ;DO YOU WANT PARAMETERS?
(1)  002602  001402                          BEQ     30$             ;IF NO, SKIP THE PARAMETER CALL
(1)  002604  004737  002764                  JSR     PC,65$          ;GET PARAMETERS
(1)  002610  012737  000001  001410  30$:    MOV     #1,SAVACTV      ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
(1)  002616  113737  001344  001414          MOVB    $TMP1,DZVNUM    ;COPY THE NUMBER OF DEVICES
(1)  002624  005337  001344          35$:    DEC     $TMP1           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
(1)  002630  001404                          BEQ     40$             ; SELECTED DEVICES
(1)  002632  000261                          SEC                     ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
(1)  002634  006137  001410                  ROL     SAVACTV         ;POINT TO THE NEXT DEVICE
(1)  002640  000771                          BR      35$             ;GO DO THIS PROCEDURE AGAIN
(1)  002642  013737  001410  001346  40$:    MOV     SAVACTV,$TMP2   ;# OF TIMES
(1)  002650  012700  001500                  MOV     #DZCR0,R0       ;SET A POINTER TO THE SPECIFIED INFORMATION
(1)  002654  012701  001512                  MOV     #DZCR1,R1       ;POINT R1 TO THE REST OF THE MAP TABLE
(1)  002660  012702  001204                  MOV     #$DDW0,R2       ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
(1)  002664  000241                          CLC                     ;INITIALIZE THE "C" BIT FOR A ROTATION
(1)  002666  006037  001346                  ROR     $TMP2           ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
(1)  002672  006237  001346          45$:    ASR     $TMP2           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
(1)  002676  103404                          BCS     50$             ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
(1)  002700  012711  177777                  MOV     #-1,(R1)        ;TERMINATE THE LIST
(1)  002704  000137  003652                  JMP     100$            ;GO TO THE NEXT BLOCK
(1)  002710  012011          50$:    MOV     (R0)+,(R1)      ;ADDRESS
(1)  002712  062721  000010                  ADD     #10,(R1)+       ;POINT TO THE NEXT DZV11 ADDRESS VALUE
(1)  002716  012011                          MOV     (R0)+,(R1)      ;VECTOR
(1)  002720  062721  000010                  ADD     #10,(R1)+       ;POINT TO THE NEXT VECTOR VALUE
(1)  002724  012021                          MOV     (R0)+,(R1)+     ;LINES
(1)  002726  012021                          MOV     (R0)+,(R1)+     ;PARAMETERS
(1)  002730  012021                          MOV     (R0)+,(R1)+     ;MAINTENANCE MODE
(1)  002732  000757                          BR      45$
(1)  002734  032777  000010  176342  55$:    BIT     #SW03,@SWR      ;ASK PARAMETERS ?
(1)  002742  001002                          BNE     60$             ;IF NO, GO DO AUTO SIZING
(1)  002744  000137  003652                  JMP     100$            ;GO SET UP FOR AUTO SIZING
(1)  002750  004737  002764          60$:    JSR     PC,65$          ;GO ASK PARAMETERS
(1)  002754  105337  001422                  DECB    INIFLG          ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
(1)  002760  000137  003710                  JMP     105$            ;GO TO THE NEXT BLOCK
(1)
(1)                                          ;GET THE ACTIVE LINES PARAMETER
(1)
(1)  002764                          65$:
(2)  002764  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
(2)  002766  003261                          93$                     ;POINTER TO MESSAGE TO BE PRINTED
(2)  002770  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)  002772  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002774  000017                          17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  002776  001504                          LINE0                   ;POINTER TO MAP LOCATION TO BE FILLED
(2)  003000     360                          .BYTE   360             ;MASK OF INVALID BITS FOR THIS PARAMETER
(2)  003001     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
```

```
                        PROGRAM INITIALIZATION AND START UP.

(1)  003002  105037  001423                CLRB    MDRFLG          ;MAKE SURE THE CHANGES ARE PRINTED
(1)
(1)                                        ;THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
(1)                                        ;IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED
(1)
(1)  003006  005737  001510                TST     MANTO           ;IS STAGGERED THE MODE OF OPERATION?
(1)  003012  100021                        BPL     85$             ;IF NOT, SKIP THIS SEGMENT
(1)  003014  013703  001504                MOV     LINE0,R3        ;GET A SCRATCH COPY OF THE ACTIVE LINES
(1)  003020  006003                70$:    ROR     R3              ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
(1)  003022  103410                        BCS     80$             ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
(1)  003024  001414                        BEQ     85$             ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
(1)  003026  006203                        ASR     R3              ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
(1)  003030  103373                        BCC     70$             ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
(1)  003032  104402  001356        75$:    TYPE    .$QUES          ;THIS IS AN INCORRECT PARAMETER
(1)  003036  104402  010350                TYPE    .MBADLN         ;LET THE USER KNOW ABOUT IT
(1)  003042  000750                        BR      65$             ;GO GET THE CORRECT PARAMETER
(1)  003044  001772                80$:    BEQ     75$             ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
(1)  003046  006203                        ASR     R3              ;GET THE NEXT FLAG
(1)  003050  103370                        BCC     75$             ;IF IT ISN'T SET, THERE'S AN ERROR
(1)  003052  000241                        CLC                     ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
(1)  003054  000761                        BR      70$             ;GO TEST THE NEXT PAIR OF FLAGS
(1)
(1)                                        ;GET THE LINE PARAMETER REGISTER ARGUMENT
(1)
(1)  003056                        85$:
(2)  003056  104403                        INSTR                   ;CALL THE STRING INPUT ROUTINE
(2)  003060  003334                        94$                     ;POINTER TO MESSAGE TO BE PRINTED
(2)  003062  104405                        PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(2)  003064  000000                        0                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  003066  000017                        17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(2)  003070  001506                        PAR0                    ;POINTER TO MAP LOCATION TO BE FILLED
(2)  003072     000                        .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
(2)  003073     001                        .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
(1)  003074  012702  001504                MOV     #LINE0,R2       ;POINT TO THE LINE SELECTION PARAMETER
(1)  003100  012703  001506                MOV     #PAR0,R3        ;POINT TO THE CHOSEN PARAMETERS
(1)  003104  011304                        MOV     (R3),R4         ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
(1)  003106  006304                        ASL     R4              ;ALIGN INDEX ON WORD BOUNDARY
(1)  003110  016437  017272  006476        MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
(1)  003116  000313                        SWAB    (R3)            ;PLACE IN HIGH BYTE
(1)  003120  052713  010070                BIS     #10070,(R3)     ;PLACE EXTRA PARAMETERS INTO LOC
(1)  003124  011262  000012        90$:    MOV     (R2),12(R2)     ;LOAD THE LINES
(1)  003130  011363  000012                MOV     (R3),12(R3)     ;LOAD THE PARAMETERS
(1)  003134  062702  000012                ADD     #12,R2          ;POINT TO THE NEXT SET
(1)  003140  062703  000012                ADD     #12,R3          ; .. OF BOTH PARAMETERS
(1)  003144  020527  001734                CMP     R3,#PAR17       ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
(1)  003150  001365                        BNE     90$             ;IF NOT, GO LOAD SOME MORE PARAMETERS
(1)  003152  000207                        RTS     PC              ;RETURN TO CALLING BLOCK
(1)  003154  030600  052123  041440  91$:  .ASCIZ  <200>/1ST CSR ADDRESS (160000:167770):  /
(1)  003220  030600  052123  053040  92$:  .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
(1)  003261     200  044514  042516  93$:  .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:17):  /
(1)  003334  042200  043105  052501  94$:  .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17):  /
(1)  003404  021600  047440  020106  95$:  .ASCIZ  <200>/# OF DZV11'S <IN OCTAL> (1:20):  /
(1)  003447     200  040515  047111  96$:  .ASCIZ  <200>/MAINTENANCE MODE/
(1)  003470  020200  042533  052130        .ASCII  <200>/ [EXTERNAL  <M325>     (E)]/
(1)  003524  020200  044533  052116        .ASCII  <200>/ [INTERNAL  <DZVCSR03-1>(I)]/
(1)  003561     200  055440  052123        .ASCIZ  <200>/ [STAGGERED <M329>     (S)]: /
```

```
CVDZB-D MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-20
CVDZBD.P11    21-AUG-84 08:28          PROGRAM INITIALIZATION AND START UP.

    (1)  003620  042600  052116  051105  97$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
    (1)          003652                  .EVEN
    (1)  003652                          100$:
    (1)  003652  122737  000377  001422          CMPB    #377,INIFLG     ;ONLY DO AUTO SIZE ON 1ST START
    (1)  003660  001013                          BNE     105$            ;
    (1)  003662  032777  000200  175414          BIT     #BIT7,@SWR      ;BIT7=1??
    (1)  003670  001007                          BNE     105$            ;BR IF NO AUTO SIZE
    (1)  003672  005737  017346                  TST     KXTFLAG         ; KXT11 ??                      ;;GPA
    (1)  003676  001402                          BEQ     1001$           ; SKIP NEXT IF NOT              ;;GPA
    (1)  003700  000137  002436                  JMP     20$             ; YES, DON'T AUTO-SIZE          ;;GPA
    (1)  003704                          1001$:                                                          ;;GPA
    (1)  003704  004737  011564                  JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
    (1)  003710  105737  001423          105$:   TSTB    HDRFLG          ;HAS THE TABLE BEEN TYPED YET?
    (1)  003714  001021                          BNE     120$            ;IF SO, DON'T TYPE IT AGAIN
    (1)  003716  105337  001423                  DECB    HDRFLG          ;INDICATE THAT THE TABLE WILL BE TYPED
    (1)  003722  104402  010322                  TYPE    ,XHEAD          ;TYPE MAP HEADER
    (1)  003726  012700  001500                  MOV     #DZV.MAP,R0     ;SET POINTER
    (1)  003732  010037  001344          110$:   MOV     R0,@TMP1        ;POINT TO THE MAP LOCATION
    (1)  003736  012037  001346                  MOV     (R0)+,@TMP2     ;SET DATA
    (1)  003742  022737  177777  001346          CMP     #-1,@TMP2       ;END OF LIST?
    (1)  003750  001403                          BEQ     120$            ;BR IF YES
    (1)  003752  104411                  115$:   CONVRT                  ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
    (1)  003754  010412                          XSTATQ                  ;CONVERT THE DATA AT THIS ADDRESS
    (1)  003756  000765                          BR      110$            ;GO PRINT THE NEXT PARAMETER
    (1)  003760  013737  001410  001406  120$:   MOV     SAVACTV,DZVACTV ;COPY BIT MAP OF ACTIVE DEVICES
    (1)  003766  113737  001414  001416          MOVB    DZVNUM,SAVNO    ;COPY NO. OF DEVICES IN THE SYSTEM
    (1)  003774  032777  000100  175302          BIT     #SW06,@SWR      ;DESELECT SPECIFIC DEVICES??
    (1)  004002  001431                          BEQ     135$            ;BR IF NO.
    (1)  004004                          121$:
    (2)  004004  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
    (2)  004006  010240                          MNEW                    ;POINTER TO MESSAGE TO BE PRINTED
    (2)  004010  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
    (2)  004012  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
    (2)  004014  177777                          177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
    (2)  004016  001406                          DZVACTV                 ;POINTER TO MAP LOCATION TO BE FILLED
    (2)  004020     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
    (2)  004021     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
    (1)  004022  023737  001406  001410          CMP     DZVACTV,SAVACTV ;IS VALUE VALID?
    (1)  004030  101403                          BLOS    122$            ;IF YES BRANCH
    (1)  004032  104402  010112                  TYPE    ,MERR3          ;IF NOT TYPE ERROR
    (1)  004036  000762                          BR      121$            ;THEN REASK QUESTION
    (1)  004040  105037  001416          122$:   CLRB    SAVNO           ;INITIALIZE NO. OF ACTIVE DEVICES
    (1)  004044  013737  001406  001344          MOV     DZVACTV,@TMP1   ;COPY BIT MAP OF ACTIVE DEVICES
    (1)  004052  006237  001344          126$:   ASR     @TMP1           ;ROTATE OUT AN ACTIVE BIT
    (1)  004056  103002                          BCC     127$            ;IF NOT ACTIVE SKIP RECORDING IT
    (1)  004060  105237  001416                  INCB    SAVNO           ;INCREMENT NO. OF ACTIVE DEVICES
    (1)  004064  001372                  127$:   BNE     126$            ;IF NOT DONE GO CONTINUE
    (1)  004066  032777  000020  175210  135$:   BIT     #SW04,@SWR      ;CHECK TO SEE IF DELAY COUNT CHANGES
    (1)  004074  001407                          BEQ     140$            ;IF NOT, GO CLEAR VECTOR AREA
    (2)  004076  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
    (2)  004100  003620                          97$                     ;POINTER TO MESSAGE TO BE PRINTED
    (2)  004102  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
    (2)  004104  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
    (2)  004106  177777                          177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
    (2)  004110  006476                          DLYCNT                  ;POINTER TO MAP LOCATION TO BE FILLED
    (2)  004112     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
```

PROGRAM INITIALIZATION AND START UP.

```
(2)  004113     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
(1)  004114  012700  000300          140$:   MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
(1)  004120  012701  000302                  MOV     #302,R1         ;VECTOR AREA. 300-776
(1)  004124  010120                  145$:   MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
(1)  004126  005021                          CLR     (R1)+           ;IN VECTOR AREA.
(1)  004130  022021                          CMP     (R0)+,(R1)+     ;POP POINTERS
(1)  004132  005737  017346                  TST     KXTFLAG         ; IF FALCON...                  ;;GPA
(1)  004136  001403                          BEQ     1002$                                           ;;GPA
(1)  004140  020027  000400                  CMP     R0,#400         ;...QUIT AT 400.                ;;GPA
(1)  004144  000402                          .+2                     ; SKIP NEXT                     ;;GPA
(1)  004146                          1002$:                                                          ;;GPA
(1)  004146  022700  001000                  CMP     #1000,R0        ;ALL DONE??
(1)  004152  001364                          BNE     145$            ;BR IF NO.
(1)
(1)                                  ;TEST START AND RESTART
(1)                                  ;----------------------
(1)
(1)  004154  012706  001120          .BEGIN: MOV     #STACK,SP       ;SET UP STACK
(1)  004160  106427  000200                  MTPS    #MASK           ;LOCK OUT INTERRUPTS
(1)  004164  005737  000042                  TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
(1)  004170  001015                          BNE     2$              ;BR IF YES
(1)  004172  032777  000004  175104          BIT     #BIT2,@#SWR     ;CHECK FOR LOCK ON TEST
(1)  004200  001406                          BEQ     1$              ;BR IF NO LOCK DESIRED.
(1)  004202  104402  010136                  TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
(1)  004206  012737  000240  004476          MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
(1)  004214  000403                          BR      2$              ;CONTINUE ALONG.
(1)  004216  013737  004724  004476  1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
(1)  004224  012737  010724  001252  2$:     MOV     #CYCLE,@LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
(1)  004232  113737  001416  001415          MOVB    SAVNO,SAVNUM    ;COPY NO. OF ACTIVE DEVICES
(1)  004240  104402  010027                  TYPE    ,MR             ;TYPE "RUNNING"
(1)  004244  000177  175002                  JMP     @@LPADR         ;START TESTING
```

```
                                        END OF PASS ROUTINE

 8457                                           ;END OF PASS
  (2)                                           ;TYPE NAME OF TEST
  (2)                                           ;UPDATE PASS COUNT
  (2)                                           ;CHECK FOR EXIT TO ACT-11
  (2)                                           ;RESTART TEST
  (3)                                   .SBTTL  END OF PASS ROUTINE
  (3)
  (4)                                   ;;**********************************************************************
  (3)                                   ;*INCREMENT THE PASS NUMBER ($PASS)
  (3)                                   ;*IF THERES A MONITOR GO TO IT
  (3)                                   ;*IF THERE ISN'T JUMP TO CYCLE
  (3)
  (3)  004250                           $EOP:
  (5)  004250  000004                           SCOPE
  (5)  004252  005037  001262                   CLR     $ERRPC          ;CLEAR LAST ERROR PC
  (5)  004256  105037  001247                   CLRB    $ERFLG          ;CLEAR ERROR FLAG
  (5)  004262  104402  010003                   TYPE    .MEPASS         ;TYPE END PASS
  (5)  004266  104402  010165                   TYPE    .MCSRX          ;TYPE CSR
  (5)  004272  104412  004434                   CNVRT   .XCSR           ;SHOW IT
  (5)  004276  104402  010173                   TYPE    .MVECX          ;TYPE VECTOR
  (5)  004302  104412  004442                   CNVRT   .XVEC           ;SHOW IT
  (5)  004306  005237  001126                   INC     $PASS           ;RAISE PASS COUNT
  (5)  004312  104402  010201                   TYPE    .MPASSX         ;TYPE PASSES
  (5)  004316  104412  004450                   CNVRT   .XPASS          ;SHOW IT
  (5)  004322  005337  001126                   DEC     $PASS           ;RESTORE PASS COUNT
  (5)  004326  104402  010212                   TYPE    .MERRX          ;TYPE ERRORS
  (5)  004332  104412  004456                   CNVRT   .XERR           ;SHOW IT
  (5)  004336  005237  001130                   INC     $DEVCT          ;INC DEVCNT FOR APT
  (5)  004342  105337  001415                   DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
  (5)  004346  001030                           BNE     $DOAGN          ;BR IF NO.
  (5)  004350  113737  001416  001415           MOVB    SAVNO,SAVNUM    ;RESTORE THE COUNT
  (3)  004356  005037  001354                   CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (3)  004362  005237  001126                   INC     $PASS           ;;INCREMENT THE PASS NUMBER
  (3)  004366  042737  100000  001126           BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
  (3)  004374  005327                           DEC     (PC)+           ;;LOOP?
  (3)  004376  000001                   $EOPCT: .WORD   1
  (3)  004400  003013                           BGT     $DOAGN          ;;YES
  (3)  004402  012737                           MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (3)  004404  000001                   $ENDCT: .WORD   1
  (3)  004406  004376                           $EOPCT
  (3)  004410  013700  000042           $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (3)  004414  001405                           BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  (3)  004416  000005                           RESET                   ;;CLEAR THE WORLD
  (3)  004420  004710                   $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (3)  004422  000240                           NOP                     ;;SAVE ROOM
  (3)  004424  000240                           NOP                     ;;FOR
  (3)  004426  000240                           NOP                     ;;ACT11
  (3)  004430                           $DOAGN:
  (3)  004430  000137                           JMP     @(PC)+          ;;RETURN
  (3)  004432  010724                   $RTNAD: .WORD   CYCLE
  (2)
  (2)  004434  000001                   XCSR:   1
  (2)  004436     006     002                   .BYTE   6,2
  (2)  004440  002010                           DZVCSR
  (2)  004442  000001                   XVEC:   1
  (2)  0044..     003     002                   .BYTE   3,2
```

```
CVDZB D MACY11 30A(1052)  21 AUG-84  08:31  PAGE 26-23
CVDZBD.P11    21-AUG-84 08:28          END OF PASS ROUTINE

(2)  004446  002040                          DZVRIV
(2)  004450  000001                  XPASS:  1
(2)  004452     006      002                 .BYTE   6,2
(2)  004454  001126                          $PASS
(2)  004456  000001                  XERR:   1
(2)  004460     006      002                 .BYTE   6,2
(2)  004462  001256                          $ERTTL
(2)
(2)                                          ;SCOPE LOOP AND ITERATION HANDLER
(2)                                          ;-------------------------------
(2)
(3)                                  .SBTTL  SCOPE HANDLER ROUTINE
(3)
(4)                                  ;;*********************************************************************
(3)                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(3)                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(3)                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(3)                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(3)                                  ;*SW14=1          LOOP ON TEST
(3)                                  ;*SW11=1          INHIBIT ITERATIONS
(3)                                  ;*CALL
(3)                                  ;*      SCOPE           ;;SCOPE=IOT
(3)
(3)  004464                          $SCOPE:
(5)  004464  005037  001262          .SCOPE: CLR     $ERRPC          ;CLEAR LAST ERROR PC.
(5)  004470  022716  012276                  CMP     @TST1+2,(SP)    ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
(5)  004474  001413                          BEQ     $XTSTR          ;IF SO, DON'T LOOP ON IT
(5)  004476  000406                  TTST:   BR      1$              ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
(5)  004500  105777  174604                  TSTB    @#TKS           ;KEYBOARD DONE?
(5)  004504  100067                          BPL     $OVER           ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
(5)  004506  017766  174600  177776          MOV     @#TKB,-2(SP)    ;CLEAR DONE BIT
(3)  004514  032777  040000  174562  1$:     BIT     #BIT14,@#SWR    ;;LOOP ON PRESENT TEST?
(3)  004522  001060                          BNE     $OVER           ;;YES IF SW14=1
(3)                                  ;@@@@@START OF CODE FOR THE XOR TESTER@@@@@
(3)  004524  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(3)                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(3)  004526  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(3)  004532  012737  004552  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(3)  004540  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
(3)  004544  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(3)  004550  000436                          BR      $SVLAD          ;;GO TO THE NEXT TEST
(3)  004552  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(3)  004554  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(3)  004560  000441                          BR      $OVER           ;;LOOP ON THE PRESENT TEST
(3)  004562                          6$:;@@@@@END OF CODE FOR THE XOR TESTER@@@@@
(3)  004562  105737  001247          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(3)  004566  001404                          BEQ     3$              ;;BR IF NO
(3)  004570  105037  001247          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(3)  004574  005037  001354                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(3)  004600  032777  004000  174476  3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
(3)  004606  001011                          BNE     1$              ;;BR IF YES
(3)  004610  005737  001126                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
(3)  004614  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
(3)  004616  005237  001250                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
(3)  004622  023737  001354  001250          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
(3)  004630  002015                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
```

```
(3)  004632  012737  000001  001250  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
(3)  004640  013737  004726  001354          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
(3)  004646  105237  001246          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
(3)  004652  113737  001246  001124          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
(3)  004660  011637  001252                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
(3)  004664  013777  001246  174414  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(3)  004672  013716  001252                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(5)  004676  004737  007322                  JSR     PC,SERV.G       ;FIND OUT IF ^G WAS TYPED
(5)  004702  105037  001424                  CLRB    MNTFLG          ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
(5)  004706  005737  001372                  TST     MODE            ;HAS THE MODE BEEN CHANGED?
(5)  004712  001003                          BNE     4$              ;IF NOT INTERNAL , GO DO A TEST
(5)  004714  112737  000010  001424          MOVB    #MAINT,MNTFLG   ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
(5)  004722  000002                  4$:     RTI                     ;GO DO THE TEST
(5)  004724  000406                  BRW:    406
(3)  004726  000005                  $MXCNT: 5                       ;;MAX. NUMBER OF ITERATIONS
(1)
(1)                                          ;CHECK FOR FREEZE ON CURRENT DATA
(1)                                          ;-------------------------------
(1)
(1)  004730  032777  001000  174346  .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
(1)  004736  001405                          BEQ     1$              ;BR IF NOT SET.
(1)  004740  005737  001364                  TST     LOCK            ;IS THERE A TIGHT LOOP SPECIFIED?
(1)  004744  001402                          BEQ     1$              ;IF NO, RETURN
(1)  004746  013716  001364                  MOV     LOCK,(SP)       ;IF YES, GOTO THE ADDRESS IN LOCK.
(1)  004752  000002                  1$:     RTI                     ;GO BACK.
(1)
(1)  004754  032737  000040  000052  .TYPE:  BIT     #BIT5,@#52      ;ARE WE UNDER UFD ?
(1)  004762  001004                          BNE     1$              ;YES,THEN SKIP ALL TYPEOUTS
(1)  004764  032777  010000  174312          BIT     #SW12,@SWR      ;INHIBIT ALL PRINTOUT??
(1)  004772  001403                          BEQ     $TYPE           ;IF NOT, GO TYPE
(1)  004774  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER MESSAGE POINTER
(1)  005000  000002                          RTI                     ;RETURN TO WHERE PROCEDURE WAS INVOKED
(2)                                  .SBTTL  TYPE ROUTINE
(2)
(3)                                  ;;**********************************************************************
(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)
(2)  005002  105737  001323          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(2)  005006  100002                          BPL     1$              ;;BR IF YES
(2)  005010  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
(2)  005012  000430                          BR      3$              ;;LEAVE
(2)  005014  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
(2)  005016  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
(2)  005022  122737  000001  001140          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
```

```
(2)  005030  001011                        BNE     62$                ;;NO,GO CHECK FOR APT CONSOLE
(2)  005032  132737  000100  001141        BITB    @APTSPOOL,$ENVM    ;;SPOOL MESSAGE TO APT
(2)  005040  001405                        BEQ     62$                ;;NO,GO CHECK FOR CONSOLE
(2)  005042  010037  005052                MOV     R0,61$             ;;SETUP MESSAGE ADDRESS FOR APT
(2)  005046  004737  005344                JSR     PC,$ATY3           ;;SPOOL MESSAGE TO APT
(2)  005052  000000                61$:    .WORD   0                  ;;MESSAGE ADDRESS
(2)  005054  132737  000040  001141  62$:  BITB    @APTCSUP,$ENVM     ;;APT CONSOLE SUPPRESSED
(2)  005062  001003                        BNE     60$                ;;YES,SKIP TYPE OUT
(2)  005064  112046                2$:     MOVB    (R0)+,-(SP)        ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(2)  005066  001005                        BNE     4$                 ;;BR IF IT ISN'T THE TERMINATOR
(2)  005070  005726                        TST     (SP)+              ;;IF TERMINATOR POP IT OFF THE STACK
(2)  005072  012600                60$:    MOV     (SP)+,R0           ;;RESTORE R0
(2)  005074  062716  000002        3$:     ADD     #2,(SP)            ;;ADJUST RETURN PC
(2)  005100  000002                        RTI                        ;;RETURN
(2)  005102  122716  000011        4$:     CMPB    @HT,(SP)           ;;BRANCH IF <HT>
(2)  005106  001430                        BEQ     8$
(2)  005110  122716  000200                CMPB    @CRLF,(SP)         ;;BRANCH IF NOT <CRLF>
(2)  005114  001006                        BNE     5$
(2)  005116  005726                        TST     (SP)+              ;;POP  <CR><LF> EQUIV
(2)  005120  104402                        TYPE                       ;;TYPE A CR AND LF
(2)  005122  001357                        $CRLF
(2)  005124  105037  005332                CLRB    $CHARCNT           ;;CLEAR CHARACTER COUNT
(2)  005130  000755                        BR      2$                 ;;GET NEXT CHARACTER
(2)  005132  004737  005214        5$:     JSR     PC,$TYPEC          ;;GO TYPE THIS CHARACTER
(2)  005136  123726  001322        6$:     CMPB    $FILLC,(SP)+       ;;IS IT TIME FOR FILLER CHARS.?
(2)  005142  001350                        BNE     2$                 ;;IF NO GO GET NEXT CHAR.
(2)  005144  013746  001320                MOV     $NULL,-(SP)        ;;GET # OF FILLER CHARS. NEEDED
(2)                                                                   ;;AND THE NULL CHAR.
(2)  005150  105366  000001        7$:     DECB    1(SP)              ;;DOES A NULL NEED TO BE TYPED?
(2)  005154  002770                        BLT     6$                 ;;BR IF NO--GO POP THE NULL OFF OF STACK
(2)  005156  004737  005214                JSR     PC,$TYPEC          ;;GO TYPE A NULL
(2)  005162  105337  005332                DECB    $CHARCNT           ;;DO NOT COUNT AS A COUNT
(2)  005166  000770                        BR      7$                 ;;LOOP
(2)
(2)                                ;HORIZONTAL TAB PROCESSOR
(2)
(2)  005170  112716  000040        8$:     MOVB    #' ,(SP)           ;;REPLACE TAB WITH SPACE
(2)  005174  004737  005214        9$:     JSR     PC,$TYPEC          ;;TYPE A SPACE
(2)  005200  132737  000007  005332        BITB    #7,$CHARCNT        ;;BRANCH IF NOT AT
(2)  005206  001372                        BNE     9$                 ;;TAB STOP
(2)  005210  005726                        TST     (SP)+              ;;POP SPACE OFF STACK
(2)  005212  000724                        BR      2$                 ;;GET NEXT CHARACTER
(2)  005214                        $TYPEC:
(2)  005214  105777  174070                TSTB    @$TKS              ;;CHAR IN KYBD BUFFER?                    ;MJD001
(2)  005220  100022                        BPL     10$                ;;BR IF NOT                               ;MJD001
(2)  005222  017746  174064                MOV     @$TKB,-(SP)        ;;GET CHAR                                ;MJD001
(2)  005226  042716  177600                BIC     #177600,(SP)       ;;STRIP EXTRANEOUS BITS                   ;MJD001
(2)  005232  122716  000023                CMPB    #$XOFF,(SP)        ;;WAS CHAR XOFF                           ;MJD001
(2)  005236  001012                        BNE     102$               ;;BR IF NOT                               ;MJD001
(2)  005240                        101$:                                                                        ;MJD001
(2)  005240  105777  174044                TSTB    @$TKS              ;;WAIT FOR CHAR                           ;MJD001
(2)  005244  100375                        BPL     101$                                                         ;MJD001
(2)  005246  117716  174040                MOVB    @$TKB,(SP)         ;;GET CHAR                                ;MJD001
(2)  005252  042716  177600                BIC     #177600,(SP)       ;;STRIP IT                                ;MJD001
(2)  005256  122716  000021                CMPB    #$XON,(SP)         ;;WAS IT XON?                             ;MJD001
(2)  005262  001366                        BNE     101$               ;;BR IF NOT                               ;MJD001
```

```
(2)  005264                          102$:                                                          ;MJD001
(2)  005264  005726                          TST     (SP)+          ;;FIX STACK                    ;MJD001
(2)  005266                          10$:                                                           ;MJD001
(2)  005266  105777  174022                  TSTB    @#TPS          ;;WAIT UNTIL PRINTER IS READY
(2)  005272  100375                          BPL     10$                                           ;MJD001
(2)  005274  116677  000002  174014          MOVB    2(SP),@#TPB    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(2)  005302  122766  000015  000002          CMPB    #CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
(2)  005310  001003                          BNE     1$             ;;BRANCH IF NO
(2)  005312  105037  005332                  CLRB    $CHARCNT       ;;YES--CLEAR CHARACTER COUNT
(2)  005316  000406                          BR      $TYPEX         ;;EXIT
(2)  005320  122766  000012  000002  1$:     CMPB    #LF,2(SP)      ;;IS CHARACTER A LINE FEED?
(2)  005326  001402                          BEQ     $TYPEX         ;;BRANCH IF YES
(2)  005330  105227                          INCB    (PC)+          ;;COUNT THE CHARACTER
(2)  005332  000000                  $CHARCNT:.WORD  0              ;;CHARACTER COUNT STORAGE
(2)  005334  000207                  $TYPEX: RTS     PC
(2)
(2)                                  .SBTTL  APT COMMUNICATIONS ROUTINE
(2)
(3)                                  ;;*****************************************************************
(2)  005336  112737  000001  005602  $ATY1:  MOVB    #1,$FFLG       ;;TO REPORT FATAL ERROR
(2)  005344  112737  000001  005600  $ATY3:  MOVB    #1,$MFLG       ;;TO TYPE A MESSAGE
(2)  005352  000403                          BR      $ATYC
(2)  005354  112737  000001  005602  $ATY4:  MOVB    #1,$FFLG       ;;TO ONLY REPORT FATAL ERROR
(2)  005362                          $ATYC:
(4)  005362  010046                          MOV     R0,-(SP)       ;;PUSH R0 ON STACK
(4)  005364  010146                          MOV     R1,-(SP)       ;;PUSH R1 ON STACK
(2)  005366  105737  005600                  TSTB    $MFLG          ;;SHOULD TYPE A MESSAGE?
(2)  005372  001450                          BEQ     5$             ;;IF NOT:  BR
(2)  005374  122737  000001  001140          CMPB    @#APTENV,$ENV  ;;OPERATING UNDER APT?
(2)  005402  001031                          BNE     3$             ;;IF NOT:  BR
(2)  005404  132737  000100  001141          BITB    @#APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
(2)  005412  001425                          BEQ     3$             ;;IF NOT:  BR
(2)  005414  017600  000004                  MOV     @4(SP),R0      ;;GET MESSAGE ADDR.
(2)  005420  062766  000002  000004          ADD     #2,4(SP)               ;;BUMP RETURN ADDR.
(2)  005426  005737  001120          1$:     TST     $MSGTYPE       ;;SEE IF DONE W/ LAST XMISSION?
(2)  005432  001375                          BNE     1$             ;;IF NOT:  WAIT
(2)  005434  010037  001134                  MOV     R0,$MSGAD      ;;PUT ADDR IN MAILBOX
(2)  005440  105720                  2$:     TSTB    (R0)+          ;;FIND END OF MESSAGE
(2)  005442  001376                          BNE     2$
(2)  005444  163700  001134                  SUB     $MSGAD,R0      ;;SUB START OF MESSAGE
(2)  005450  006200                          ASR     R0             ;;GET MESSAGE LNGTH IN WORDS
(2)  005452  010037  001136                  MOV     R0,$MSGLGT     ;;PUT LENGTH IN MAILBOX
(2)  005456  012737  000004  001120          MOV     #4,$MSGTYPE    ;;TELL APT TO TAKE MSG.
(2)  005464  000413                          BR      5$
(2)  005466  017637  000004  005512  3$:     MOV     @4(SP),4$      ;;PUT MSG ADDR IN JSR LINKAGE
(2)  005474  062766  000002  000004          ADD     #2,4(SP)               ;;BUMP RETURN ADDRESS
(4)  005502  013746  177776                  MOV     177776,-(SP)   ;;PUSH 177776 ON STACK
(2)  005506  004737  005002                  JSR     PC,$TYPE       ;;CALL TYPE MACRO
(2)  005512  000000                  4$:     .WORD   0
(2)  005514                          5$:
(2)  005514  105737  005602          10$:    TSTB    $FFLG          ;;SHOULD REPORT FATAL ERROR?
(2)  005520  001416                          BEQ     12$            ;;IF NOT:  BR
(2)  005522  005737  001140                  TST     $ENV           ;;RUNNING UNDER APT?
(2)  005526  001413                          BEQ     12$            ;;IF NOT:  BR
(2)  005530  005737  001120          11$:    TST     $MSGTYPE       ;;FINISHED LAST MESSAGE?
(2)  005534  001375                          BNE     11$            ;;IF NOT:  WAIT
```

```
(2)  005536  017637  000004  001122          MOV     @4(SP),$FATAL   ;;GET ERROR #
(2)  005544  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(2)  005552  005237  001120                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
(2)  005556  105037  005602          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
(2)  005562  105037  005601                  CLRB    $LFLG           ;;CLEAR LOG FLAG
(2)  005566  105037  005600                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
(4)  005572  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(4)  005574  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(2)  005576  000207                          RTS     PC              ;;RETURN
(2)  005600     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
(2)  005601     000          $LFLG:  .BYTE   0               ;;LOG FLAG
(2)  005602     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
(2)          005604                  .EVEN
(2)          000200          APTSIZE=200
(2)          000001          APTENV=001
(2)          000100          APTSPOOL=100
(2)          000040          APTCSUP=040
(1)
(1)                                  ;STRING INPUT ROUTINE
(1)                                  ;-------------------------
(1)
(1)  005604  010346          .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
(1)  005606  010446                  MOV     R4,-(SP)        ;SAVE R4 ON STACK
(1)  005610  017637  000004  005626  MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
(1)  005616  062766  000002  000004  ADD     #2,4(SP)        ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
(1)  005624  104402          .INST1: TYPE                    ;PRINT THE MESSAGE
(1)  005626  000000          .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
(1)  005630  012704  010620          MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
(1)  005634  012703  000007          MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
(1)  005640  105777  173444  1$:     TSTB    @$TKS           ;HAS A CHARACTER BEEN RECEIVED?
(1)  005644  100375                  BPL     1$              ;IF NO, KEEP WAITING FOR IT
(1)  005646  117714  173440          MOVB    @$TKB,(R4)      ;IF YES, SAVE IT IN THE INPUT BUFFER
(1)  005652  142714  000200          BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
(1)  005656  122427  000015          CMPB    (R4)+,#15       ;IS THIS CHARACTER A LINE FEED?
(1)  005662  001417                  BEQ     INSTR2          ;IF SO, TERMINATE THE INPUT SEQUENCE
(1)  005664  105777  173424  2$:     TSTB    @$TPS           ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
(1)  005670  100375                  BPL     2$              ;IF WE CAN'T, WAIT UNTIL WE CAN
(1)  005672  017777  173414  173416  MOV     @$TKB,@$TPB     ;ECHO THE CHARACTER BACK
(1)  005700  005303                  DEC     R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
(1)  005702  001356                  BNE     1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
(1)  005704  012604                  MOV     (SP)+,R4        ;IF WE HAVE 7, RESTORE R4
(1)  005706  012603                  MOV     (SP)+,R3        ;RESTORE R3
(1)  005710  010346          .INSTE: MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
(1)  005712  010446                  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
(1)  005714  104402  001356          TYPE    ,$QUES          ;PRINT A QUESTION MARK... WHAT'S GOING ON?
(1)  005720  000741                  BR      .INST1          ;GO PRINT THE MESSAGE AGAIN
(1)  005722  012604          INSTR2: MOV     (SP)+,R4        ;RESTORE R4
(1)  005724  012603                  MOV     (SP)+,R3        ;RESTORE R3
(1)  005726  000002                  RTI                     ;RETURN TO THE MAIN PROCEDURE
(1)
(1)                                  ;CONVERT ASCII STRING TO OCTAL
(1)                                  ;---------------------------
(1)
(1)  005730  010546          .PARAM: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
(1)  005732  010446                  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
(1)  005734  016605  000004          MOV     4(SP),R5        ;GET THE SETUP INFORMATION POINTER
```

```
(1)  005740  012537  006120                MOV     (R5)+,LOLIM     ;SET THE LOW LIMIT FOR THE INPUT
(1)  005744  012537  006122                MOV     (R5)+,HILIM     ;SET THE HIGH LIMIT FOR THE INPUT
(1)  005750  012537  006124                MOV     (R5)+,DEVADR    ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
(1)  005754  112537  006126                MOVB    (R5)+,LOBITS    ;GET THE MASK OF THE INCORRECT BITS
(1)  005760  112537  006127                MOVB    (R5)+,ADRCNT    ;GET THE COUNT OF ITEMS TO BE STORED
(1)  005764  010366  000004                MOV     R5,4(SP)        ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
(1)  005770  005005                PARAM1: CLR     R5              ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
(1)  005772  012704  010620                MOV     #INBUF,R4       ;POINT TO THE INPUT BUFFER
(1)  005776  122714  000015                CMPB    #15,(R4)        ;IS THIS CHARACTER A CARRIAGE RETURN?
(1)  006002  001420                        BEQ     PARERR          ;IF SO, PRINT THE MESSAGE AGAIN
(1)  006004  121427  000060        1$:     CMPB    (R4),#60        ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
(1)  006010  002415                        BLT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
(1)  006012  121427  000067                CMPB    (R4),#67        ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
(1)  006016  003012                        BGT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
(1)  006020  142714  000060                BICB    #60,(R4)        ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
(1)  006024  152405                        BISB    (R4)+,R5        ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
(1)  006026  122714  000015                CMPB    #15,(R4)        ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
(1)  006032  001406                        BEQ     LIMITS          ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
(1)  006034  006305                        ASL     R5              ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
(1)  006036  006305                        ASL     R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
(1)  006040  006305                        ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
(1)                                                                ;NEXT THREE BITS
(1)  006042  000760                        BR      1$              ;GO GET THE NEXT CHARACTER
(1)  006044  104404                PARERR: INSTER                  ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
(1)  006046  000750                        BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN
(1)
(1)                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
(1)                                        ;-------------------------------------
(1)
(1)  006050  020537  006122        LIMITS: CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
(1)  006054  101373                        BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
(1)  006056  020537  006120                CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
(1)  006062  103770                        BLO     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
(1)  006064  133705  006126                BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
(1)  006070  001365                        BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
(1)
(1)                                        ;STORE NUMBER AT SPECIFIED ADDRESS
(1)
(1)  006072  013704  006124                MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
(1)  006076  010524                1$:     MOV     R5,(R4)+        ;STORE THE RESULT
(1)  006100  062705  000002                ADD     #2,R5           ;CALCULATE THE NEXT DATUM
(1)  006104  105337  006127                DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
(1)  006110  001372                        BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
(1)  006112  012604                        MOV     (SP)+,R4        ;RESTORE R4
(1)  006114  012605                        MOV     (SP)+,R5        ;RESTORE R5
(1)  006116  000002                        RTI                     ;RETURN TO THE MAIN PROGRAM
(1)
(1)  006120  000000                LOLIM:  0                       ;LOWEST ACCEPTABLE VALUE
(1)  006122  000000                HILIM:  0                       ;HIGHEST ACCEPTABLE
(1)  006124  000000                DEVADR: 0                       ;LOCATION WHERE RESULT WILL BE STORED
(1)  006126     000                LOBITS: .BYTE   0               ;INCORRECT BITS MASK
(1)  006127     000                ADRCNT: .BYTE   0               ;COUNT OF ITEMS TO BE STORED
(1)
(1)                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
(1)                                        ;-------------------------------------
(1)
```

```
(1)  006130  016637  000004  001404  .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
(1)
(1)                                          ;SAVE R0-R5
(1)
(1)  006136  010537  001340          SV05:   MOV     R5,$REG5        ;SAVE R5
(1)  006142  010437  001336                  MOV     R4,$REG4        ;SAVE R4
(1)  006146  010337  001334                  MOV     R3,$REG3        ;SAVE R3
(1)  006152  010237  001332                  MOV     R2,$REG2        ;SAVE R2
(1)  006156  010137  001330                  MOV     R1,$REG1        ;SAVE R1
(1)  006162  010037  001326                  MOV     R0,$REG0        ;SAVE R0
(1)  006166  000002                          RTI                     ;LEAVE.
(1)
(1)                                          ;RESTORE R0-R5
(1)
(1)  006170  013700  001326          .RES05: MOV     $REG0,R0        ;RESTORE R0
(1)  006174  013701  001330                  MOV     $REG1,R1        ;RESTORE R1
(1)  006200  013702  001332                  MOV     $REG2,R2        ;RESTORE R2
(1)  006204  013703  001334                  MOV     $REG3,R3        ;RESTORE R3
(1)  006210  013704  001336                  MOV     $REG4,R4        ;RESTORE R4
(1)  006214  013705  001340                  MOV     $REG5,R5        ;RESTORE R5
(1)  006220  000002                          RTI                     ;LEAVE
(1)
(1)                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
(1)                                          ;-----------------------------------------------------
(1)
(1)  006222  104402  001357          .CONVR: TYPE    ,$CRLF          ;PRINT A CARRIAGE RETURN
(1)  006226  010046                  .CNVRT: MOV     R0,-(SP)        ;SAVE R0
(1)  006230  010146                          MOV     R1,-(SP)        ;SAVE R1
(1)  006232  010346                          MOV     R3,-(SP)        ;SAVE R3
(1)  006234  010446                          MOV     R4,-(SP)        ;SAVE R4
(1)  006236  010546                          MOV     R5,-(SP)        ;SAVE R5
(1)  006240  017601  000012                  MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
(1)  006244  062766  000002  000012          ADD     #2,12(SP)       ;POINT TO WHERE MAIN PROGRAM WILL RESUME
(1)  006252  012137  006376                  MOV     (R1)+,WRDCNT    ;GET NUMBER OF WORDS TO BE PRINTED
(1)  006256  112105                  1$:     MOVB    (R1)+,R5        ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
(1)  006260  112100                          MOVB    (R1)+,R0        ;GET THE NUMBER OF SPACES TO PRINT
(1)  006262  013104                          MOV     @(R1)+,R4       ;COPY THE WORD TO BE CONVERTED
(1)  006264  110537  006400                  MOVB    R5,CHRCNT       ;COPY THE CHARACTER COUNT
(1)  006270  010403                  3$:     MOV     R4,R3           ;COPY THE ARGUMENT WORD AGAIN
(1)  006272  042703  177770                  BIC     #^C<7>,R3       ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
(1)  006276  062703  000060                  ADD     #060,R3         ;MAKE AN ASCII CHARACTER OUT OF THEM
(1)  006302  110346                          MOVB    R3,-(SP)        ;SAVE THAT CHARACTER
(1)  006304  006004                          ROR     R4              ;MOVE THE NEXT THREE BITS INTO PLACE
(1)  006306  006204                          ASR     R4              ;MOVE THEM AGAIN
(1)  006310  006204                          ASR     R4              ;AND FINALLY A THIRD TIME
(1)  006312  005305                          DEC     R5              ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
(1)                                                                  ;BUILT?
(1)  006314  001365                          BNE     3$              ;IF NO, GO BUILD THE NEXT ONE.
(1)  006316  012703  010662                  MOV     #MDATA,R3       ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
(1)  006322  112623                  4$:     MOVB    (SP)+,(R3)+     ;STORE THE CHARACTER, STARTING WITH THE MOST
(1)  006324  105337  006400                  DECB    CHRCNT          ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
(1)  006330  001374                          BNE     4$              ;IF NO, GO TRANSFER ANOTHER
(1)  006332  105700                          TSTB    R0              ;ARE ANY SPACES TO BE PRINTED?
(1)  006334  001404                          BEQ     6$              ;IF NO, DON'T SET UP ANY
(1)  006336  112723  000040          5$:     MOVB    #040,(R3)+      ;ADD A SPACE TO THE OUTPUT BUFFER
(1)  006342  105300                          DECB    R0              ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
```

```
CVDZB-0 MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-30
CVDZB0.P11    21-AUG-84 08:28          APT COMMUNICATIONS ROUTINE

   (1)  006344  001374                          BNE     5$                ;IF YES, GO ADD ANOTHER SPACE
   (1)  006346  105013                  6$:     CLRB    (R3)              ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
   (1)  006350  104402  010662                  TYPE    ,MDATA            ;PRINT THE STRING WE JUST BUILT
   (1)  006354  005337  006376                  DEC     WRDCNT            ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
   (1)  006360  001336                          BNE     1$                ;IF YES, GO CONVERT THEM
   (1)  006362  012605                          MOV     (SP)+,R5          ;RESTORE R5
   (1)  006364  012604                          MOV     (SP)+,R4          ;RESTORE R4
   (1)  006366  012603                          MOV     (SP)+,R3          ;RESTORE R3
   (1)  006370  012601                          MOV     (SP)+,R1          ;RESTORE R1
   (1)  006372  012600                          MOV     (SP)+,R0          ;RESTORE R0
   (1)  006374  000002                          RTI                       ;RETURN TO THE MAIN PROGRAM
   (1)  006376  000000                  WRDCNT: 0
   (1)  006400     000                  CHRCNT: .BYTE                     ;NUMBER OF CHARACTERS TO PRINT
   (1)  006401     000                  SPACNT: .BYTE   0                 ;NUMBER OF SPACES TO PRINT
   (1)
   (1)  006402  000000                  BINWRD: 0
   (1)
   (1)
   (1)                                          ;TRAP DISPATCH SERVICE
   (1)                                          ;ARGUMENT OF TRAP IS EXTRACTED
   (1)                                          ;AND USED AS OFFSET TO OBTAIN POINTER
   (1)                                          ;TO SELECTED SUBROUTINE
   (1)
   (1)  006404  010046                  .TRPSR: MOV     R0,-(SP)          ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
   (1)  006406  016600  000002                  MOV     2(SP),R0          ;GET TRAP ADDRESS
   (1)  006412  005740                          TST     -(R0)             ;GET TRAP
   (1)  006414  111000                          MOVB    (R0),R0           ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
   (1)  006416  006300                          ASL     R0                ;POSITION OFFSET FOR TABLE INDEXING
   (1)  006420  016000  001742                  MOV     .TRPTAB(R0),R0    ;PLACE INDEXED ADDRESS OF TABLE IN R0
   (1)  006424  000200                          RTS     R0                ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0
   (1)
   (1)                                          ;DEVICE CLEAR ROUTINE
   (1)                                          ;ISSUE A DEVICE CLEAR
   (1)                                          ;-------------------
   (1)  006426                          .DEVICE.CLR:
   (1)  006426  052777  000020  173354          BIS     #DCLR,@DZVCSR     ;SET DCLR
   (1)  006434  032777  000020  173346  1$:     BIT     #DCLR,@DZVCSR     ;DID IT CLEAR?
   (1)  006442  001374                          BNE     1$                ;BR IF NO
   (1)  006444  000002                          RTI                       ;EXIT ROUTINE
   (1)
   (1)                                          ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
   (1)                                          ;-----------------------------------------------------------
   (1)  006446  104413                  .DCLASM:DEVICE.CLR                ;ISSUE A DEVICE CLEAR
   (1)  006450  153777  001424  173332          BISB    MNTFLG,@DZVCSR    ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
   (1)  006456  000002                          RTI                       ;RETURN TO CALLING ROUTINE
   (1)
   (1)  006460                          .DELAY:
   (1)  006460  010046                          MOV     R0,-(SP)          ;SAVE R0
   (1)  006462  013700  006476                  MOV     DLYCNT,R0         ;SET COUNT
   (1)  006466  005300                  1$:     DEC     R0                ;DELAY
   (1)  006470  001376                          BNE     1$                ;
   (1)  006472  012600                          MOV     (SP)+,R0          ;RESTORE R0
   (1)  006474  000002                          RTI                       ;LEAVE ROUTINE
   (1)  006476  000001                  DLYCNT: .WORD     1               ;PATCHABLE LOC FOR MORE TIME
   (1)
   (1)                                          ;ADVANCE TO NEXT TEST HANDLER
```

```
(1)                                             ;- - --------- ------ - -   -
(1)
(1)  006500  013716  001362          .ADVANCE:MOV     NEXT,(SP)       ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
(1)  006504  005037  001364                   CLR     LOCK            ;RESET TIGHT LOOP ADDRESS
(1)  006510  000002                           RTI                     ;CHECK TO SEE IF OLD TEST GETS REPEATED
(1)
(1)                                           ;ROUTINE TO SHIFT LINE POINTER
(1)                                           ;AND SWITCH TESTS IF NECESSARY
(1)                                           ;-----------------------------
(1)  006512  106302                   .SHIFT: ASLB    R2              ;POINT TO THE NEXT LINE
(1)  006514  032702  000020                   BIT     #BIT4,R2        ;HAVE WE PASSED ALL LINE POINTERS?
(1)  006520  001402                           BEQ     1$              ;IF NOT, RETURN TO THE TEST
(1)  006522  022626                           POP2SP                  ;REMOVE THE TRAP CALL FROM THE STACK
(1)  006524  104400                           ADVANCE                 ;GO TO THE NEXT TEST
(1)  006526  000002                   1$:     RTI                     ;RETURN TO THE PRESENT TEST
(1)
```

```
(1)                                             ;LINE PARAMETER REGISTER SETUP ROUTINE
(1)
(1)  006530  010146                 .LPRSET:MOV     R1,-(SP)        ;SAVE CONTENTS OF R1
(1)  006532  010246                         MOV     R2,-(SP)        ;SAVE CONTENTS OF R2
(1)  006534  013701  001370                 MOV     PAR,R1          ;MOVE DEFAULT PARAM. INTO R1
(1)  006540  012702  000001                 MOV     #1,R2   ;INIT. FOR LINE 1
(1)  006544  010177  173250         1$:     MOV     R1,@DZVLPR      ;LOAD PARAM. REGISTER
(1)  006550  005201                         INC     R1              ;SET R1 FOR NEXT LINE
(1)  006552  106302                         ASLB    R2              ;SET R2 FOR NEXT LINE
(1)  006554  032702  000020                 BIT     #BIT4,R2        ;ALL LINES DONE?
(1)  006560  001771                         BEQ     1$              ;IF NO LOAD NEXT LINE
(1)  006562  012602                         MOV     (SP)+,R2        ;RELOAD R2
(1)  006564  012601                         MOV     (SP)+,R1        ;RELOAD R1
(1)  006566  000002                         RTI                     ;RETURN
(1)
(1)                                             ;ROUTINE TO ZERO DATA BUFFER
(1)
(1)  006570  010046                 .BUFSET:MOV     R0,-(SP)        ;SAVE CONTENTS OF R0
(1)  006572  012700  001426                 MOV     #TDO,R0         ;SET R0 TO TOP OF BUFFER
(1)  006576  005020                 1$:     CLR     (R0)+           ;CLEAR BUFFER LOCATION
(1)  006600  022700  001446                 CMP     #STOP,R0        ;IS BUFFER ALL CLEARED
(1)  006604  001374                         BNE     1$              ;IF NOT CLEAR NEXT LOCATION
(1)  006606  012600                         MOV     (SP)+,R0        ;RELOAD R0
(1)  006610  000002                         RTI                     ;RETURN
(1)
(2)
(3)                                     ;;*****************************************************************
(2)                                     .SBTTL  ABORT ROUTINE FOR LCP/ORION UFD MODE
(3)                                     ;;*****************************************************************
(2)
(2)          000040                     UFD=BIT5
(2)  006612  032737  000040  000052 ABORT:  BIT     @UFD,52         ;TEST FOR USER FRIENDLY MODE
(2)  006620  001420                         BEQ     ABORT2          ;IF NOT UFD THEN CONTINUE NORMAL OPERATION
(2)
(2)
(2)  006622  013737  002152  000030         MOV     SAV30,30        ;RESTORE EMT LOCATION (30)
(2)  006630  013737  002154  000032         MOV     SAV32,32        ;RESTORE EMT PRIORITY LOCATION (32)
(2)  006636  104042                         EMT     +42             ;GET DCA LOCATION INTO R0 FROM MONITOR
(2)  006640  012760  177777  000042         MOV     #-1,42(R0)      ;SET A -1 INTO LOCATION DRSERR IN MONITOR
(2)  006646  013700  000042         ABORT1: MOV     42,R0           ;AND PUT THE MONITOR RETURN ADDRESS IN R0
(2)  006652  005037  000042                 CLR     42              ;CLEAR MONITOR RETURN FLAG
(2)  006656  000137  004420                 JMP     $ENDAD          ;RETURN TO MONITOR-DO NOT PUSH STACK HERE
(2)  006662  000207                 ABORT2: RTS     PC              ;IF NOTUFD RETURN TO MAINLINE
(2)
(2)
(3)                                     ;;*****************************************************************
(1)
(1)                                             ;ERROR HANDLER
(1)                                             ;-------------
(1)
(1)  006664  004737  006612         $ERROR: JSR     PC,ABORT        ;ARE WE UNDER UFD ?
(1)  006670  004737  007322                 JSR     PC,SERV.G       ;FIND OUT IF <^G> WAS HIT
(1)  006674  032777  010000  172402         BIT     #SW12,@SHR      ;BELL ON ERROR?
(1)  006702  001406                         BEQ     XBX             ;BR IF NO BELL
(1)  006704  105777  172404                 TSTB    @#TPS           ;TTY READY.
(1)  006710  100003                         BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
```

```
(1)  006712  112777  000207  172376          MOVB    #207,@#TPB      ;PUSH A BELL AT THE TTY.
(1)  006720  032777  020000  172356  XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
(1)  006726  001113                          BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
(1)  006730  021637  001262                  CMP     (SP),@ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
(1)  006734  001404                          BEQ     1$              ;BR IF YES
(1)  006736  011637  001262                  MOV     (SP),@ERRPC     ;RECORD BEING HERE
(1)  006742  105037  001247                  CLRB    @ERFLG          ;PREPARE HEADER
(1)  006746  104407                  1$:     SAV05                   ;SAVE ALL PROC REGISTERS
(1)  006750  011605                          MOV     (SP),R5         ;GET THE PC OF ERROR
(1)  006752  162705  000002                  SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
(1)  006756  011504                          MOV     (R5),R4         ;GET ERROR INSTRUCTION
(1)  006760  110437  001260                  MOVB    R4,@ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
(1)  006764  006304                          ASL     R4              ;MULT BY TWO
(1)  006766  061504                          ADD     (R5),R4         ;DOUBLE IT
(1)  006770  006304                          ASL     R4              ;MULT AGAIN
(1)  006772  042704  177001                  BIC     #177001,R4      ;CLEAR JUNK
(1)  006776  062704  015422                  ADD     #.ERRTAB,R4     ;GET POINTER
(1)  007002  012437  007126                  MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
(1)  007006  012437  007140                  MOV     (R4)+,DATAHD    ;GET DATA HEADER
(1)  007012  011437  007152                  MOV     (R4),DATABP     ;GET DATA TABLE
(1)  007016  105737  001247                  TSTB    @ERFLG          ;TYPE HEADER
(1)  007022  001403                          BEQ     TYPMSG          ;BR IF YES
(1)  007024  005737  007152                  TST     DATABP          ;DOES DATA TABLE EXIST?
(1)  007030  001044                          BNE     TYPDAT          ;BR IF YES.
(1)  007032  104402  001357          TYPMSG: TYPE    .@CRLF          ;TYPE A CARRIAGE RETURN
(1)  007036  104402  001357                  TYPE    .@CRLF          ;AND TYPE ANOTHER
(1)  007042  005737  001364                  TST     LOCK
(1)  007046  001402                          BEQ     2$
(1)  007050  104402  010235                  TYPE    .MASTEK
(1)  007054  104402  010223          1$:     TYPE    .MTSTN
(1)  007060  104412  007314                  CNVRT   .XTSTN          ;SHOW IT
(1)  007064  104402  010315                  TYPE    .MERRPC         ;TYPE PC.
(1)  007070  104412  007306                  CNVRT   .ERTABO         ;SHOW IT
(1)  007074  104402  010165                  TYPE    .MCSRX
(1)  007100  104412  004434                  CNVRT   .XCSR
(1)  007104  104402  001357                  TYPE    .@CRLF          ;GIVE A CR/LF
(1)  007110  112737  177777  001247          MOVB    #-1,@ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
(1)  007116  005737  007126                  TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
(1)  007122  001402                          BEQ     WTBS.FM         ;BR IF NO.
(1)  007124  104402                          TYPE                    ;TYPE
(1)  007126  000000                  ERRMSG: 0                       ;    ERROR MESSAGE
(1)  007130                          WTBS.FM:                        ;
(1)  007130  005737  007140                  TST     DATAHD          ;DATA HEADER?
(1)  007134  001402                          BEQ     TYPDAT          ;BR IF NO
(1)  007136  104402                          TYPE                    ;TYPE
(1)  007140  000000                  DATAHD: 0                       ;    DATA HEADER
(1)  007142  005737  007152          TYPDAT: TST     DATABP          ;DATA TABLE?
(1)  007146  001402                          BEQ     RESREG          ;BR IF NO.
(1)  007150  104411                          CONVRT                  ;SHOW
(1)  007152  000000                  DATABP: 0                       ;    DATA TABLE
(1)  007154  104410                  RESREG: RES05                   ;RESTORE PROC REGISTERS
(1)  007156  122737  000001  001140  HALTS:  CMPB    @APTENV,#ENV    ;IS APT RUNNING?
(1)  007164  001007                          BNE     15$             ;SKIP APT CALL IF NOT
(1)  007166  113737  001260  007200          MOVB    @ITEMB,5$       ;COPY ERROR NUMBER
(1)  007174  004737  005354                  JSR     PC,@ATY4        ;CALL APT SERVICE
(1)  007200  000000                  5$:     .WORD   0               ;ERROR NUMBER STUCK HERE
```

```
CVDZB-D MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-34
CVDZBD.P11    21-AUG-84 08:28          ABORT ROUTINE FOR LCP/ORION UFD MODE

(1)  007202  000777                  10$:    BR      10$             ;LOCK UP HERE
(1)  007204  022737  004420  000042  15$:    CMP     @#ENDAD,#042    ;CHECK TO SEE IF IN ACT-11 MODE
(1)  007212  001403                          BEQ     20$             ;IF SO, HANDLE ACCORDINGLY
(1)  007214  005777  172064                  TST     @SWR            ;HALT ON ERROR?
(1)  007220  100006                          BPL     EXITER          ;BR IF NO HALT ON ERROR
(1)  007222  016677  000002  172056  20$:    MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
(1)  007230  004737  006612                  JSR     PC,ABORT        ;CHECK IF WE ARE UNDER UFD?
(1)  007234  000000                          HALT                    ;HALT
(1)  007236  005237  001256          EXITER: INC     $ERTTL          ;UPDATE ERROR COUNT
(1)  007242  004737  007322                  JSR     PC,SERV.G       ;FIND OUT IF ^G WAS TYPED
(1)  007246  032777  000400  172030          BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
(1)  007254  001007                          BNE     1$              ;BR IF YES
(1)  007256  032777  002000  172020          BIT     #SW10,@SWR      ;GOTO NEXT TEST?
(1)  007264  001407                          BEQ     2$              ;BR IF NO
(1)  007266  013737  001362  001252          MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
(1)  007274  012706  001120          1$:     MOV     #STACK,SP       ;RESET SP
(1)  007300  000177  171746                  JMP     @$LPADR         ;GOTO SPECIFIED TEST
(1)  007304  000002                  2$:     RTI                     ;RETURN
(1)  007306  000001                  ERTABO: 1
(1)  007310     006     002                  .BYTE   6,2
(1)  007312  001404                          SAVPC
(1)  007314  000001                  XTSTN:  1
(1)  007316     002     002                  .BYTE   2,2
(1)  007320  001246                          $TSTNM
(1)  007322  017746  171764          SERV.G: MOV     @$TKB,-(SP)     ;OTHERWISE, GET THE LAST CHARACTER TYPED
(1)  007326  042716  000200                  BIC     #BIT7,(SP)      ;STRIP PARITY(EIGHTH) BIT
(1)  007332  122726  000007                  CMPB    #7,(SP)+        ;IS IT ^G?
(1)  007336  001076                          BNE     6$              ;IF NOT, IGNORE INPUT
(1)  007340  032777  004000  171742          BIT     #4000,@$TKS     ;RX BUSY?
(1)  007346  001365                          BNE     SERV.G          ;BR IF YES
(1)          007350          GETSWR: .                                                       ;;GPA
(1)  007350  017737  171730  007556          MOV     @SWR,90$        ;SAVE (SWR).
(1)  007356  104402  007536          1$:     TYPE    ,89$            ;TYPE HEADER FOR OLD SWITCH REGISTER
(1)  007362  104412  007550                  CNVRT   ,88$            ;TYPE THE NUMBER ITSELF
(1)  007366  104402  007560                  TYPE    ,91$            ;AFTER HAVING CONVERTED IT TO ASCII
(1)  007372  105037  007564                  CLRB    92$             ;CLEAR SWR CHANGE FLAG
(1)  007376  005077  171702                  CLR     @SWR            ;CLEAR THE SOFTWARE SWITCH REGISTER
(1)  007402  105777  171702          3$:     TSTB    @$TKS           ;WAIT FOR DONE.
(1)  007406  100375                          BPL     3$              ;CONTINUE WAITING FOR IT
(1)  007410  017746  171676                  MOV     @$TKB,-(SP)     ;PUT THE CHARACTER ON THE STACK
(1)  007414  042716  000200                  BIC     #BIT7,(SP)      ;STRIP PARITY BIT
(1)  007420  122726  000015                  CMPB    #15,(SP)+       ;IS IT THE CARRIAGE RETURN CHAR?
(1)  007424  001433                          BEQ     4$              ;IF SO, GO PRINT CRLF
(1)  007426  105777  171662          2$:     TSTB    @$TPS           ;IS THE OUTPUT BUFFER AVAILABLE
(1)  007432  100375                          BPL     2$              ;IF NOT, WAIT FOR IT TO BE READY
(1)  007434  105237  007564                  INCB    92$             ;INDICATE THAT THE SWR WAS CHANGED
(1)  007440  014677  171652                  MOV     -(SP),@$TPB     ;PLACE THE CHARACTER THERE(ECHO BACK)
(1)  007444  000241                          CLC                     ;GET READY TO ROTATE
(1)  007446  006177  171632                  ROL     @SWR            ;MOVE THE EXISTING BITS OVER
(1)  007452  006177  171626                  ROL     @SWR            ;TO MAKE ROOM FOR THE INCOMING
(1)  007456  006177  171622                  ROL     @SWR            ;THREE BITS FROM THIS CHARACTER
(1)  007462  103735                          BCS     1$              ;ERROR
(1)  007464  022627  000060                  CMP     (SP)+,#60       ;IS IT LOWER THAN 0?
(1)  007470  002732                          BLT     1$              ;IF SO, GO ASK AGAIN
(1)  007472  026627  177776  000067          CMP     -2(SP),#67      ;IS IT HIGHER THAN 7?
(1)  007500  003326                          BGT     1$              ;IF SO, GO ASK AGAIN
```

```
                                ABORT ROUTINE FOR LCP/ORION UFD MODE

(1)  007502  042746  177770            BIC     #^C<7>,-(SP)    ;ISOLATE INFORMATION BITS
(1)  007506  052677  171572            BIS     (SP)+,@SWR      ;ADD THEM TO THE SWITCH REGISTER
(1)  007512  000733                    BR      3$              ;GO CHECK FOR THE NEXT CHARACTER
(1)  007514  105737  007564      4$:   TSTB    92$             ;HAS THE SWR BEEN CHANGED?
(1)  007520  001003                    BNE     5$              ;IF YES GO TYPE CRLF
(1)  007522  013777  007556  171554    MOV     90$,@SWR        ;IF NOT RESTORE SWR
(1)  007530  104402  001357      5$:   TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN AND LINE FEED
(1)  007534  000207              6$:   RTS     PC              ;RETURN TO CALLING PROCEDURE
(1)
(1)  007536  020200  051450  051127  89$:   .ASCIZ  <200>? (SWR)=/?
(1)  007544  036451  000057
(1)                                    .EVEN
(1)  007550  000001              88$:  1
(1)  007552     006     000            .BYTE   6,0
(1)  007554  007556                    90$
(1)  007556  000000              90$:  .WORD   0
(1)  007560  036457  000057      91$:  .ASCIZ  ?/=/?
(1)  007564     000              92$:  .BYTE   0
(1)          007566                    .EVEN
(2)                                    .SBTTL  POWER DOWN AND UP ROUTINES
(2)
(3)                                    ;;*****************************************************************
(2)                                    ;POWER DOWN ROUTINE
(2)  007566  012737  007732  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
(2)  007574  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
(4)  007602  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(4)  007604  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(4)  007606  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(4)  007610  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(4)  007612  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
(4)  007614  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(4)  007616  017746  171462                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
(2)  007622  010637  007736                  MOV     SP,$SAVR6       ;;SAVE SP
(2)  007626  012737  007640  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
(2)  007634  000000                          HALT
(2)  007636  000776                          BR      .-2             ;;HANG UP
(2)
(3)                                    ;;*****************************************************************
(2)                                    ;POWER UP ROUTINE
(2)  007640  012737  007732  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
(2)  007646  013706  007736                  MOV     $SAVR6,SP       ;;GET SP
(2)  007652  005037  007736                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
(2)  007656  005237  007736          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
(2)  007662  001375                          BNE     1$              ;;OF  WORD
(4)  007664  012677  171414                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
(4)  007670  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
(4)  007672  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
(4)  007674  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
(4)  007676  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(4)  007700  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(4)  007702  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(2)  007704  012737  007566  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
(2)  007712  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
(2)  007720  104402                          TYPE                    ;;REPORT THE POWER FAILURE
(2)  007722  007740              $PWRMG: .WORD   MPFAIL          ;;POWER FAIL MESSAGE POINTER
(2)  007724  012716                          MOV     (PC)+,(SP)      ;;RESTART AT RESTART
```

```
CVDZB-D MACY11 30A(1052)  21-AUG-84  08:31  PAGE 26-36
CVDZBO.P11    21-AUG-84 08:28          POWER DOWN AND UP ROUTINES

   (2)  007726  011270                 $PWRAD: .WORD   RESTART          ;;RESTART ADDRESS
   (2)  007730  000002                         RTI
   (2)  007732  000000                 $ILLUP: HALT                     ;;THE POWER UP SEQUENCE WAS STARTED
   (2)  007734  000776                         BR      .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
   (2)  007736  000000                 $SAVR6: 0                        ;;PUT THE SP HERE
   (2)  007740  050200  051127  043040 MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
   (2)  010003     200  047105  020104 MEPASS: .ASCIZ  <200>/END PASS CVDZB-D  /
   (2)  010027     200  052522  047116 MR:    .ASCIZ  <200>/RUNNING   /
   (2)  010043     200  051120  043517 MERR2:  .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
   (2)  010112  044600  051516  043125 MERR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
   (2)  010136  046200  041517  020113 MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
   (2)  010165     103  051123  020072 MCSRX:  .ASCIZ   /CSR: /
   (2)  010173     126  041505  020072 MVECX:  .ASCIZ  /VEC: /
   (2)  010201     120  051501  042523 MPASSX: .ASCIZ  /PASSES: /
   (2)  010212  051105  047522  051522 MERRX:  .ASCIZ  /ERRORS: /
   (2)  010223     124  051505  020124 MTSTN:  .ASCIZ  /TEST NO: /
   (2)  010235     052  000040        MASTEK: .ASCIZ  /* /
   (2)  010240  052200  050131  020105 MNEW:   .ASCIZ  <200>/TYPE A BIT MAP OF DZV11'S DESIRED ACTIVE:  /
   (2)  010315     120  035103  000040 MERRPC: .ASCIZ  /PC: /
   (2)  010322  046600  050101  047440 XHEAD:  .ASCIZ  <200>/MAP OF DZV11 STATUS/<200>
   (2)  010350  044600  046114  043505 MBADLN: .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
   (2)                                 .EVEN
   (2)  010412  000002                 XSTATQ: 2
   (2)  010414     006     003                 .BYTE   6,3
   (2)  010416  001344                         $TMP1
   (2)  010420     006     002                 .BYTE   6,2
   (2)  010422  001346                         $TMP2
   (1)                                 .EVEN
   (2)                                         ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
   (2)                                         ;-------------------------------------------------------------
   (2)                                         ;E=EXTERNAL LOOP BACK
   (2)                                         ;I=INTERNAL LOOP BACK
   (2)                                         ;S=STAGGERED LOOP BACK
   (2)  010424  017605  000000        .SETFLG:MOV     @(SP),R5         ;PICK UP ADDRESS OF TAG
   (2)  010430  042737  000040  010620        BIC     #40,INBUF        ;STRIP LOWER CASE
   (2)  010436  122737  000105  010620        CMPB    #'E,INBUF        ;IS IT EXTERNAL LOOP BACK ?
   (2)  010444  001005                        BNE     4$               ;NO
   (2)  010446  013715  010536                MOV     1$,(R5)          ;YES STORE INFO
   (2)  010452  105037  001424                CLRB    MNTFLG           ;SET MAINT BIT =0
   (2)  010456  000422                        BR      7$               ;GET OUT
   (2)  010460  122737  000111  010620  4$:   CMPB    #'I,INBUF        ;IS IT INTERNAL LOOP BACK ?
   (2)  010466  001006                        BNE     5$               ;NO
   (2)  010470  013715  010540                MOV     2$,(R5)          ;YES STORE INFO
   (2)  010474  112737  000010  001424        MOVB    #MAINT,MNTFLG    ;SET UP THE MAINTENANCE FLAG LOADER
   (2)  010502  000410                        BR      7$               ;GET OUT
   (2)  010504  122737  000123  010620  5$:   CMPB    #'S,INBUF        ;IS IT STAGGERED LOOP BACK ?
   (2)  010512  001007                        BNE     6$               ;WHAT ?
   (2)  010514  013715  010542                MOV     3$,(R5)          ;YES STORE INFO
   (2)  010520  105037  001424                CLRB    MNTFLG           ;ZERO BITS
   (2)  010524  062716  000002        7$:     ADD     #2,(SP)          ;POP AROUND
   (2)  010530  000002                        RTI
   (2)  010532  104404                6$:     INSTER                   ;RETRY
   (2)  010534  000733                        BR      .SETFLG          ;DITTO
   (2)  010536  000200                1$:     .WORD   200              ;EXTERNAL = E
   (2)  010540  000000                2$:     .WORD   0                ;INTERNAL = I
   (2)  010542  100000                3$:     .WORD   100000           ;STAGGERED = S
```

(2)

```
(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)  010544  017605  000000          .PAWCH:MOV     @(SP),R5
(2)  010550  142737  000040  010620          BICB    #40,INBUF       ;SET FOR LOWER CASE INPUT
(2)  010556  122737  000105  010620          CMPB    #'E,INBUF       ;IS IT "E" ?
(2)  010564  001002                          BNE     1$
(2)  010566  105015                          CLRB    (R5)            ;000
(2)  010570  000406                          BR      2$
(2)  010572  122737  000103  010620  1$:     CMPB    #'C,INBUF       ;IS IT "C" ?
(2)  010600  001005                          BNE     3$
(2)  010602  112715  177777                  MOVB    #-1,(R5)        ;3177
(2)  010606  062716  000002          2$:     ADD     #2,(SP)
(2)  010612  000002                          RTI
(2)  010614  104404                  3$:     INSTER                  ;RETRY
(2)  010616  000752                          BR      .PAWCH
(2)
(2)                                             ;BUFFERS FOR INPUT-OUTPUT
(2)
(2)  010620  000000                  INBUF:  0
(2)          010662                  .=.+40
(2)                                  ; TEMP: 0               ; TEMP AREA UNUSED.                     ;;GPA
(2)                                  ; .=.+40                ; DELETED TO CONSERVE SPACE.            ;;GPA
(2)  010662  000000                  MDATA:  0
(2)          010724                  .=.+40
(2)
```

```
(2)
(2)                                             ;
(2)                                             ;ROUTINE USED TO "CYCLE" THROUGH UP TO SIXTEEN DZV11'S
(2)                                             ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
(2)                                             ;AND RUNS THE SPECIFIED DZV11'S.   THIS ROUTINE *MUST*
(2)                                             ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
(2)                                             ;SETUP NECESSARY.
(2)                                             ;
(2)
(2)  010724  005737  001406          CYCLE:  TST     DZVACTV         ;ARE ANY DZV11'S TO BE TESTED?
(2)  010730  001006                          BNE     1$              ;BR IF OK.
(2)  010732  104402  010043                  TYPE    ,MERR2          ;NO DZV11'S SELECTED!!
(2)  010736  004737  006612                  JSR     PC,ABORT        ;CHECK IF WE ARE UNDER UFD?
(2)  010742  000000                          HALT                    ;STOP THE SHOW.
(2)  010744  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
(2)  010746  013737  004726  001354  1$:     MOV     @MXCNT,#TIMES   ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
(2)  010754  033737  001412  001406          BIT     RUN,DZVACTV     ;IS THIS ONE "ACTIVE"
(2)  010762  001017                          BNE     2$              ;BR IF GOOD ONE FOUND.
(2)  010764  006137  001412                  ROL     RUN             ;UPDATE POINTER
(2)  010770  005537  001412                  ADC     RUN             ;CATCH CARRY FROM RUN
(2)  010774  062737  000012  001420          ADD     #12,ACTIVE      ;UPDATE ADDRESS POINTER.
(2)  011002  022737  001740  001420          CMP     #DZV.END,ACTIVE ;HAVE WE PASSED THE END OF THE MAP?
(2)  011010  001356                          BNE     1$              ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
(2)  011012  012737  001500  001420          MOV     #DZV.MAP,ACTIVE ;RESET ADDRESS POINTER.
(2)  011020  000752                          BR      1$              ;KEEP LOOKING FOR ACTIVE DZV11
(2)  011022  006137  001412          2$:     ROL     RUN             ;UPDATE POINTER.
(2)  011026  005537  001412                  ADC     RUN             ;CATCH CARRY.
(2)  011032  013700  001420                  MOV     ACTIVE,R0       ;GET ADDRESS POINTER.
(2)  011036  062737  000012  001420          ADD     #12,ACTIVE      ;UPDATE.
(2)  011044  022737  001740  001420          CMP     #DZV.END,ACTIVE
(2)                                                                  ;ALL DONE?
(2)  011052  001003                          BNE     3$              ;BR IF NO.
(2)  011054  012737  001500  001420          MOV     #DZV.MAP,ACTIVE ;RESTORE POINTER.
(2)  011062  012037  001174          3$:     MOV     (R0)+,#BASE     ;LOAD SYSTEM CTRL. REG
(2)  011066  012037  002040                  MOV     (R0)+,DZVRIV    ;LOAD VECTOR
(2)  011072  012037  001366                  MOV     (R0)+,LINE      ;SET UP DZV LINES ACTIVE
(2)  011076  012037  001370                  MOV     (R0)+,PAR       ;SET UP PARAMETERIZATION
(2)  011102  012037  001372                  MOV     (R0)+,MODE      ;SET UP MAINTENANCE MODE
(2)  011106  105037  001424                  CLRB    MNTFLG  ;RESET MAINT. FLAG IF
(2)  011112  005737  001372                  TST     MODE            ;RUNNING TESTS
(2)  011116  001003                          BNE     9$              ;IN
(2)  011120  112737  000010  001424          MOVB    #MAINT,MNTFLG   ;INTERNAL MAINT. MODE
(2)  011126  004737  011274          9$:     JSR     PC,DZVLEV       ;SET UP
(2)  011132  005737  000042                  TST     @#42            ;ARE WE UNDER MONITOR CONTROL?
(2)  011136  001051                          BNE     7$              ;IF YES, SKIP THIS SETUP
(2)  011140  032777  000002  170136          BIT     #SW01,@SWR      ;IF SW01=1, GET STARTING TEST #
(2)  011146  001445                          BEQ     7$              ;BR IF NO TEST IS TO BE INPUTTED
(2)  011150  104402  001357          4$:     TYPE    ,#CRLF
(3)  011154  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
(3)  011156  010223                          MTSTN                   ;POINTER TO MESSAGE TO BE PRINTED
(3)  011160  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
(3)  011162  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(3)  011164  001000                          1000                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
(3)  011166  001246                          #TSTNM                  ;POINTER TO MAP LOCATION TO BE FILLED
(3)  011170     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
(3)  011171     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
```

```
(2) 011172 012700 012274          MOV   #TST1,R0
(2) 011176 022710 000004    5$:   CMP   #4,(R0)
(2) 011202 001020                 BNE   6$
(2) 011204 022760 012737 000002   CMP   #12737,2(R0)
(2) 011212 001014                 BNE   6$
(2) 011214 023760 001246 000004   CMP   $TSTNM,4(R0)    ;IS THIS THE TEST ?
(2) 011222 001010                 BNE   6$             ;IF NOT, DON'T PROCESS NUMBER
(2) 011224 010037 001252          MOV   R0,$LPADR      ;SAVE PC
(2) 011230 062737 000002 001252   ADD   #2,$LPADR      ;POP OVER PREVIOUS SCOPE
(2) 011236 104402 001357          TYPE  ,$CRLF
(2) 011242 000412                 BR    8$
(2) 011244 005720           6$:   TST   (R0)+
(2) 011246 020027 014426          CMP   R0,#TLAST+10
(2) 011252 001351                 BNE   5$
(2) 011254 104402 001356          TYPE  ,$QUES
(2) 011260 000733                 BR    4$
(2) 011262 012737 012274 001252 7$:   MOV   #TST1,$LPADR    ;PREPARE TEST ADDRESS
(2) 011270                       8$:
(2) 011270 000177 167756 RESTART: JMP   @$LPADR        ;GO START TESTING.***WARNING!****
(2)                                                    ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!
(2)
(2)                              ;THIS UTILITY SETS UP CSR'S,SETS UP VECTORS.
(2) 011274 013700 002040 DZVLEV: MOV   DZVRIV,R0      ;PLACE THE BASE VECTOR ADDRESS IN R0
(2) 011300 062700 000002          ADD   #2,R0          ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
(2) 011304 010037 002042          MOV   R0,DZVRIS      ;STORE IT HERE
(2) 011310 062700 000002          ADD   #2,R0          ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
(2) 011314 010037 002044          MOV   R0,DZVTIV      ;STORE IT HERE
(2) 011320 062700 000002          ADD   #2,R0          ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
(2) 011324 010037 002046          MOV   R0,DZVTIS      ;STORE IT HERE
(2)
(2)                              ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZV11. $BASE IS THE BASE ADDRESS
(2)                              ;OF THE DEVICE
(2) 011330 013700 001174          MOV   $BASE,R0       ;COPY THE ADDRESS BEING LOADED
(2) 011334 010037 002010          MOV   R0,DZVCSR      ;XXX0
(2) 011340 005200                 INC   R0
(2) 011342 010037 002012          MOV   R0,HDZVCSR     ;XXX1
(2) 011346 005200                 INC   R0
(2) 011350 010037 002014          MOV   R0,DZVRBUF     ;XXX2
(2) 011354 010037 002020          MOV   R0,DZVLPR      ;XXX2
(2) 011360 005200                 INC   R0
(2) 011362 010037 002016          MOV   R0,HDZVRBUF    ;XXX3
(2) 011366 010037 002022          MOV   R0,HDZVLPR     ;XXX3
(2) 011372 005200                 INC   R0
(2) 011374 010037 002024          MOV   R0,DZVTCR      ;XXX4
(2) 011400 005200                 INC   R0
(2) 011402 010037 002026          MOV   R0,HDZVTCR     ;XXX5
(2) 011406 005200                 INC   R0
(2) 011410 010037 002030          MOV   R0,DZVMSR      ;XXX6
(2) 011414 010037 002034          MOV   R0,DZVTDR      ;XXX6
(2) 011420 005200                 INC   R0
(2) 011422 010037 002032          MOV   R0,HDZVMSR     ;XXX7
(2) 011426 010037 002036          MOV   R0,HDZVTDR     ;XXX7
(2) 011432 000207                 RTS   PC
```

```
(2)                                     ;CONVERT DECIMAL ASCII STRING TO OCTAL
(2)  011434  000002                     .PARMD: RTI             ; DECIMAL PARAMETERS UNUSED.        ;;GPA
(2)                                     .REM &                  ; DELETED TO CONSERVE SPACE...      ;;GPA
(2)                                                             ;...AND REMAIN UNDER 4KW SIZE.      ;;GPA
(2)                                     .PARMD: MOV     (SP),R5
(2)                                             MOV     (R5)+,6$
(2)                                             MOV     (R5)+,7$
(2)                                             MOV     (R5)+,8$
(2)                                             MOVB    (R5)+,9$
(2)                                             MOVB    (R5)+,10$
(2)                                             MOV     R5,(SP)
(2)                                     2$:     CLR     R5
(2)                                             MOV     #INBUF,R4
(2)                                             CMPB    #15,(R4)
(2)                                             BEQ     3$
(2)                                     1$:     CMPB    (R4),#'0
(2)                                             BLT     3$
(2)                                             CMPB    (R4),#'9
(2)                                             BGT     3$
(2)                                             BICB    #'0,(R4)
(2)                                             CLR     R2
(2)                                             BISB    (R4)+,R2
(2)                                             ADD     R2,R5
(2)                                             CMPB    #15,(R4)
(2)                                             BEQ     4$
(2)                                             ASL     R5      ;X2
(2)                                             MOV     R5,R2   ;SAVE X2
(2)                                             ASL     R5      ;X4
(2)                                             ASL     R5      ;X8
(2)                                             ADD     R2,R5   ;TIMES 10
(2)                                             BR      1$
(2)                                     3$:     INSTER
(2)                                             BR      2$
(2)
(2)                                             ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
(2)
(2)                                     4$:     CMP     R5,7$
(2)                                             BHI     3$
(2)                                             CMP     R5,6$
(2)                                             BLO     3$
(2)                                             BITB    9$,R5
(2)                                             BNE     3$
(2)
(2)                                             ;STORE NUMBER AT SPECIFIED ADDRESS
(2)
(2)                                             MOV     8$,R4
(2)                                     5$:     MOV     R5,(R4)+
(2)                                             ADD     #2,R5
(2)                                             DECB    10$
(2)                                             BNE     5$
(2)                                             RTI
(2)                                     6$:     0
(2)                                     7$:     0
(2)                                     8$:     0
(2)                                     9$:     .BYTE 0
(2)                                     10$:    .BYTE 0
```

```
(2)                                     ; END OF .PARMD DELETE RANGE            &                       ;;GPA
```

```
(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)
(2)  011436  012700  001500          SETAPT:    MOV     #DZV.MAP,R0     ;POINT TO THE DEVICE MAP TABLE
(2)  011442  013701  001174                     MOV     @BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
(2)  011446  013702  001170                     MOV     @VECT1,R2       ;BUILD DEVICE VECTORS IN R2
(2)  011452  042702  177007                     BIC     #^C<770>,R2     ;STRIP AWAY OTHER INFORMATION
(2)  011456  012704  001204                     MOV     #@DDW0,R4       ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
(2)  011462  013705  001176                     MOV     @DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
(2)  011466  105037  001414                     CLRB    DZVNUM          ;INITIALIZE THE NO. OF ACTIVE DEVICES
(2)  011472  005037  001410                     CLR     SAVACTV         ;CLEAR THE ACTIVE BIT MAP
(2)  011476  006005                  1$:        ROR     R5              ;GET A DEVICE SELECTION BIT
(2)  011500  103407                             BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
(2)  011502  001402                             BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
(2)  011504  005724                             TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
(2)  011506  062701  000010          2$:        ADD     #10,R1          ;SET UP THE NEXT ADDRESS
(2)  011512  062702  000010                     ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
(2)  011516  000767                             BR      1$              ;GO SEE IF MORE DEVICES REMAIN
(2)  011520  006137  001410          3$:        ROL     SAVACTV         ;SET BIT IN ACTIVE DEVICE MAP
(2)  011524  105237  001414                     INCB    DZVNUM          ;INCREMENT NO. OF ACTIVE DEVICES
(2)  011530  010120                             MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
(2)  011532  010220                             MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
(2)  011534  013720  001200                     MOV     @CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
(2)  011540  012420                             MOV     (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
(2)  011542  013720  001202                     MOV     @CDW2,(R0)+     ;LOAD DEFAULT TESTING MODE
(2)  011546  000757                             BR      2$              ;GO BUILD THE NEXT ADDRESS
(2)  011550  012710  177777          5$:        MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
(2)  011554  012737  001142  001304             MOV     #@SWREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
(2)  011562  000207                             RTS     PC              ;RETURN TO PRINT STATUS TABLE
(2)
(2)
(2)                                             ;*ROUTINE USED TO "AUTO SIZE" THE DZV11
(2)                                             ;*CSR AND VECTOR.
(2)                                             ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
(2)                                             ;*      ADDRESS RANGE (160000:167770)
(2)                                             ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
(2)                                             ;*      FLOATING VECTOR RANGE (300:770)
(2)                                             ;*
(2)
(2)  011564                          AUTO.SIZE:
(2)  011564  000005                             RESET                   ;INSURE A BUS INIT.
(2)  011566  105337  001422                     DECB    INIFLG          ;SHOW THAT I WAS HERE
(2)  011572  012702  001500          CSRMAP:    MOV     #DZV.MAP,R2     ;LOAD MAP POINTER.
(2)  011576  012703  001204                     MOV     #@DDW0,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
(2)  011602  005022                  1$:        CLR     (R2)+           ;ZERO ENTIRE MAP
(2)  011604  022702  001740                     CMP     #DZV.END,R2     ;ALL DONE?
(2)  011610  001374                             BNE     1$              ;BR IF NO
(2)  011612  105037  001414                     CLRB    DZVNUM          ;SET OCTAL NUMBER OF DZV11'S TO 0
(2)  011616  012702  001500                     MOV     #DZV.MAP,R2
(2)  011622  012701  160000                     MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
(2)  011626  012737  012076  000004             MOV     #6$,@4          ;SET FOR NON-EXISTENT DEVICE TIME OUT
(2)  011634  052711  000040          2$:        BIS     #BIT5,(R1)      ;TRY TO SET MASTER SCAN ENABLE
(2)  011640  052761  000017  000004             BIS     #17,4(R1)       ;TRY TO TRANSMIT ON ANY LINE
(2)  011646  005000                             CLR     R0              ;USE R0 AS A COUNTER
(2)  011650  005711                  7$:        TST     (R1)            ;HAS TRANSMITTER READY COME UP?
```

```
(2) 011652 100403                          BMI   8$              ;IF SO, GO GET A FINAL CHECK
(2) 011654 005300                          DEC   R0              ;REDUCE COUNT. TIME UP?
(2) 011656 001374                          BNE   7$              ;IF NOT, KEEP WAITING
(2) 011660 000437                          BR    3$              ;ASSUME IT'S NOT A DZV11
(2) 011662 032761 000017 000004  8$:       BIT   #17,4(R1)       ;ARE ANY TCR BITS  STILL SET? THEY SHOULD BE
(2) 011670 001433                          BEQ   3$              ;IF IT'S NOT, ASSUME IT'S NOT A DZV11
(2) 011672 032711 000040                    BIT   #BIT5,(R1)      ;IS MASTER SCAN ENABLE STILL SET?
(2) 011676 001430                          BEQ   3$              ;IF NOT, ASSUME IT'S NOT A DZV11
(2) 011700 052711 000020                    BIS   #20,(R1)        ;SET DEVICE CLEAR
(2) 011704 000240                          NOP
(2) 011706 032711 000040                    BIT   #40,(R1)        ;DID SCANNER CLEAR
(2) 011712 001022                          BNE   3$              ;IF NOT ASSUME IT IS NOT DZV
(2) 011714 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) 011720 010122                          MOV   R1,(R2)+        ;STORE CSR IN CORE TABLE.
(2) 011722 005722                          TST   (R2)+           ;POP OVER VECTOR STORE AREA
(2) 011724 012722 000017                    MOV   #17,(R2)+       ;SET THE DEFAULT LINE SELECTION PARAMETER
(2) 011730 012712 017470                    MOV   #17470,(R2)     ;SET THE DEFAULT PARAMETERS
(2) 011734 012223                          MOV   (R2)+,(R3)+     ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
(2) 011736 005022                          CLR   (R2)+           ;SET THE DEFAULT MODE OF OPERATION
(2) 011740 012712 177777                    MOV   #-1,(R2)        ;TERMINATE LIST
(2) 011744 105237 001414                    INCB  DZVNUM          ;UPDATE DEVICE COUNTER
(2) 011750 122737 000020 001414             CMPB  #20,DZVNUM      ;ARE MAX. NO. OF DEV FOUND?
(2) 011756 001405                          BEQ   100$            ;YES DON'T LOOK FOR ANY MORE.
(2) 011760 062701 000010           3$:      ADD   #10,R1          ;UPDATE CSR POINTER ADDRESS
(2) 011764 022701 164000                    CMP   #164000,R1
(2) 011770 001321                          BNE   2$              ;BR IF MORE ADDRESS TO CHECK.
(2) 011772                         100$:
(2) 011772 105737 001414                    TSTB  DZVNUM          ;WERE ANY DZV11'S FOUND AT ALL?
(2) 011776 001430                          BEQ   5$              ;ERROR AUTO SIZER FOUND NO DZV11'S IN THIS SYS.
(2) 012000 113701 001414                    MOVB  DZVNUM,R1
(2) 012004 012737 000001 001410             MOV   #1,SAVACTV      ;CREATE A BIT MAP OF
(2) 012012 005301                  4$:      DEC   R1              ;THE DEVICES IN THE SYSTEM
(2) 012014 001404                          BEQ   98$
(2) 012016 000261                          SEC
(2) 012020 006137 001410                    ROL   SAVACTV
(2) 012024 000772                          BR    4$
(2) 012026 013737 001500 001174  98$:       MOV   DZCR0,#BASE     ;POINT TO THE ADDRESS OF FIRST DEVICE
(2) 012034 013737 001510 001202             MOV   MANT0,#CDW2     ;INDICATE TO ETABLE WHAT MODE IS BEING USED
(2) 012042 012737 000006 000004  99$:       MOV   #6,B#4          ;RESTORE TRAP VECTOR
(2) 012050 013737 001410 001176             MOV   SAVACTV,#DEVM   ;SAVE ACTIVE REGISTER
(2) 012056 000412                          BR    VECMAP          ;GO FIND THE VECTOR NOW.
(2) 012060 104402 010043           5$:      TYPE  ,MERR2          ;NOTIFY OPR THAT NO DZV11'S FOUND.
(2) 012064 005000                          CLR   R0              ;MAKE DATA DISPLAY ZERO
(2) 012066 004737 006612                    JSR   PC,ABORT        ;CHECK IF WE ARE UNDER UFD ?
(2) 012072 000000                          HALT                  ;STOP THE SHOW
(2) 012074 000776                          BR    .-2             ;DISABLE CONT. SW.
(2) 012076 012716 011760           6$:      MOV   #3$,(SP)        ;ENTERED BY NON-EXISTENT TIME-OUT
(2) 012102 000002                          RTI                   ;RETURN TO MAINSTREAM
(2)
(2) 012104 012737 000200 000022  VECMAP: MOV   #MASK,B#22        ;SET IOT TRAP PRIORITY
(2) 012112 012737 012226 000020             MOV   #4$,B#20        ;SET IOT TRAP VECTOR
(2) 012120 012702 001500                    MOV   #DZV.MAP,R2     ;SET SOFTWARE POINTER
(2) 012124 012700 000300                    MOV   #300,R0         ;FLOATING VECTORS START HERE.
(2) 012130 012701 000302                    MOV   #302,R1         ;PC OF IOT INSTR.
(2) 012134 010120                  1$:      MOV   R1,(R0)+        ;START FILLING VECTOR AREA
```

```
(2)  012136  012721  000004                    MOV     #4,(R1)+        ;WITH .+2; IOT
(2)  012142  022021                            CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
(2)  012144  020127  001000                    CMP     R1,#1000        ;HAS THE VECTOR AREA BEEN EXCEEDED?
(2)  012150  101771                            BLOS    1$              ;BR IF MORE TO FILL
(2)  012152  013704  001410                    MOV     SAVACTV,R4      ;STORE TEMPORARILY
(2)  012156  006004                    2$      ROR     R4              ;BRING OUT A BIT
(2)  012160  103036                            BCC     5$              ;BR IF ALL DONE
(2)  012162  106427  000000                    MTPS    #0              ;ZERO CPU PRIO
(2)  012166  012772  040040  000000            MOV     #BIT14+BIT5,@(R2)       ;SET TIE AND MAS SCAN
(2)  012174  011201                            MOV     (R2),R1         ;GET CSR
(2)  012176  112761  000017  000004            MOVB    #17,4(R1)       ;SET THE TCR BITS FOR ALL LINES
(2)                                                                    ;ATTEMPT TO FORCE AN INTERRUPT
(2)  012204  005200                            INC     R0              ;STALL
(2)  012206  001376                            BNE     .-2             ;       FOR TIME TO INTERRUPT
(2)  012210  012762  000300  000002            MOV     #300,2(R2)      ;NO INTERRUPT ASSUME 300 AND FIX DZV11 LATER
(2)  012216  000005                            RESET                   ;INIT
(2)  012220  062702  000012            3$:     ADD     #12,R2          ;POP SOFTWARE POINTER
(2)  012224  000754                            BR      2$              ;KEEP GOING
(2)  012226  011662  000002            4$:     MOV     (SP),2(R2)      ;GET VECTOR ADDRESS
(2)  012232  162762  000010  000002            SUB     #10,2(R2)       ;POINT BACK TO THE CORRECT VECTOR
(2)  012240  042762  000007  000002            BIC     #7,2(R2)        ;CLEAR JUNK
(2)  012246  022626                            POP2SP                  ;POP IOT JUNK OFF STACK
(2)  012250  012716  012220                    MOV     #3$,(SP)        ;SET FOR RETURN
(2)  012254  000002                            RTI
(2)  012256  013737  001502  001170    5$:     MOV     DZVC0,@#VECT1   ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
(2)  012264  012737  004464  000020            MOV     #.SCOPE.,IOTVEC ;RESTORE THE SCOPE TRAP
(2)  012272  000207                            RTS     PC              ;ALL DONE WITH "AUTO SIZING"
(2)
```

```
                                                ;************************ TEST 1 ******************************
                                                ;*THIS TEST VERIFIES OVERRUN AND SILO ALARM
                                                ;*ONE LINE AT A TIME  - BASED UPON VALID LINES
                                                ;*AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
                                                ;*TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
                                                ;*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
                                                ;*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
                                                ;*CHAR PULLED OUT OF THE SILO.
                                                ;*ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.
                                                ;*USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
                                                ;*ON DZV LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                                ;*USED TO SCOPE SILO ALARM PULSES, ETC.
                                        ;;*   TEST 1
                                        ;;*****************************************************
  012274  000004                        TST1:   SCOPE
  012276  012737  000001  001246                MOV     #1,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
  012304  012737  012736  001362                MOV     #TST2,NEXT      ;POINT TO THE START OF THE NEXT TEST
  012312  012737  012652  001364                MOV     #18$,LOCK       ;SET FOR LOOP
  012320  104417                                DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
  012322  104421                                LPRSET                  ;LOAD LINE PARAMETERS
  012324  005037  001374                        CLR     SAVLIN          ;INIT LINE INDICATOR
  012330  104422                                BUFSET                  ;ZERO DATA BUFFER
  012332  012702  000001                        MOV     #1,R2           ;LINE POINTER
  012336  052777  010040  167444                BIS     #MSENAB!SILOEN,@DZVCSR  ;START SCANNER & SET SILO ENABLE
  012344  030237  001366                3$:     BIT     R2,LINE         ;VALID LINE?
  012350  001533                                BEQ     21$             ;IF NOT GO TO NEXT LINE
  012352  013700  001374                        MOV     SAVLIN,R0       ;MAKE OFFSET
  012356  006300                                ASL     R0              ;MAKE POWER OF TWO
  012360  010277  167440                        MOV     R2,@DZVTCR      ;SET TCR BIT
  012364  105777  167420                4$:     TSTB    @DZVCSR         ;REC DONE = 1 ?
  012370  100001                                BPL     .+4
  012372  104020                                ERROR+  20              ;REC DONE SHOULD NOT = 1
  012374  005003                                CLR     R3              ;SET CHARACTER COUNT
  012376  005004                        5$:     CLR     R4
  012400  005777  167404                6$:     TST     @DZVCSR         ;IS TRDY SET?
  012404  100404                                BMI     7$              ;IF YES, LOAD CHAR.
  012406  104414                                DELAY                   ;WAIT FOR TRDY TO SET
  012410  005204                                INC     R4              ;INC DELAY COUNTER
  012412  001372                                BNE     6$
  012414  104003                                ERROR+  3               ;*TRDY FAILED TO SET
  012416  116077  001426  167410        7$:     MOVB    TDO(R0),@DZVTDR ;LOAD A CHARACTER
  012424  005260  001426                        INC     TDO(R0)         ;SET UP NEXT CHARACTER
  012430  020327  000017                        CMP     R3,#15.         ;16 CHARACTERS ?
  012434  103006                                BHIS    8$
  012436  032777  020000  167344                BIT     #SILOAL,@DZVCSR ;SILO ALARM = 0 ?
  012444  001413                                BEQ     10$             ;YES
  012446  104013                                ERROR+  13              ;*SILO ALARM SHOULD NOT = 1
                                                                        ;UNTIL 16. DATA CHARACTERS
  012450  000411                                BR      10$
  012452  005004                        8$:     CLR     R4
  012454  032777  020000  167326        9$:     BIT     #SILOAL,@DZVCSR
  012462  001004                                BNE     10$
  012464  104414                                DELAY
  012466  005204                                INC     R4
  012470  001371                                BNE     9$
```

```
 8514  012472  104014                          ERROR*  14              ;*SILO ALARM FAILED TO SET!
 8515                                                                  ;SILO ALARM SHOULD =1 AFTER 16.
 8516                                                                  ;DATA CHARACTERS
 8517  012474  005203                    10$:   INC     R3              ;INC CHAR COUNT
 8518  012476  022703  000102                   CMP     #66.,R3         ;FINISHED SENDING CHARACTERS ?
 8519  012502  001335                          BNE     5$              ;NO
 8520  012504  005004                          CLR     R4
 8521  012506  104414                          DELAY
 8522  012510  105204                          INCB    R4
 8523  012512  001375                          BNE     .-4
 8524                                          ;NOW LETS READ THE SILO
 8525  012514  013705  001374                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
 8526  012520  005737  001372                  TST     MODE            ;IS THIS TEST IN STAGGERED MODE?
  (1)  012524  100006                          BPL     13$             ;IF NOT, SKIP STAGGERED SETUP
  (1)
  (1)                                         ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
  (1)
  (1)  012526  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
  (1)  012530  103402                          BCS     11$             ;IF IT IS SET, GO CLEAR IT
  (1)  012532  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
  (1)  012534  000401                          BR      12$             ;SKIP THE CLEARING
  (1)  012536  000241                    11$:  CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
  (1)  012540  006105                    12$:  ROL     R5              ;GET THE NEW BIT BACK INTO R5
 8527  012542  000305                    13$:  SWAB    R5              ;PUT IN UPPER BYTE
 8528  012544  052705  100000                  BIS     #DVALID,R5      ;ADD DATA VALID
 8529  012550  017704  167240            14$:  MOV     @#DZVRBUF,R4    ;ACTUAL
 8530  012554  020405                          CMP     R4,R5           ;ACTUAL VS. EXPECTED
 8531  012556  001401                          BEQ     15$             ;YES
 8532  012560  104006                          ERROR*  6               ;*DATA/CONTENTS DID NOT COMPARE
 8533  012562  032777  020000  167220    15$:  BIT     #SILOAL,@#DZVCSR ;SILO ALARM= 0 ?
 8534  012570  001401                          BEQ     16$             ;YES
 8535  012572  104016                          ERROR*  16              ;READING DZVRBUF DID NOT CLEAR SILO ALARM
 8536  012574  005205                    16$:  INC     R5              ;UP CHARACTER
 8537  012576  120527  000077                  CMPB    R5,#63.         ;LAST SILO CHAR ?....64TH CHAR
 8538  012602  101762                          BLOS    14$
 8539  012604  005205                          INC     R5              ;ADD 1 MORE FOR THE CLOBBERED CHAR
 8540  012606  052705  040000                  BIS     #OVRRUN,R5      ;ADD OVERRUN TO EXPECTED
 8541  012612  120527  000101                  CMPB    R5,#65.         ;LAST CHARACTER ?
 8542  012616  001754                          BEQ     14$
 8543  012620  017704  167170                  MOV     @#DZVRBUF,R4    ;FOR GOOD MEASURE
 8544  012624  005704                          TST     R4              ;DATA VALID SHOULD = 0
 8545  012626  100001                          BPL     17$             ;YES
 8546  012630  104017                          ERROR*  17              ;DATA VALID SHOULD = 0
 8547  012632  040277  167166            17$:  BIC     R2,@#DZVTCR     ;CLR TCR BIT
 8548  012636  104401                          SCOP1                   ;LOOP?
 8549  012640  005237  001374            21$:  INC     SAVLIN          ;INC EXPECTED LINE
 8550  012644  104420                          SHIFT                   ;NEXT LINE
 8551  012646  000137  012344                  JMP     3$              ;YES
 8552
 8553                                         ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
 8554                                         ;ON DZV LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
 8555                                         ;USED TO SCOPE SILO ALARM PULSES, ETC.
 8556
 8557  012652  052777  010040  167130    18$:  BIS     #MSENAB!SILOEN,@#DZVCSR  ;SETUP DEVICE
 8558  012660  012777  012726  167156          MOV     #20$,@#DZVTIV   ;SETUP TRANSMITTER VECTOR
 8559  012666  012701  000024                  MOV     #20.,R1         ;TEMPORARY COUNT OF CHARACTER BURST
```

```
 8560  012672  050277  167126                        BIS     R2,@DZVTCR      ;ENABLE LINE
 8561  012676  052777  040000  167104                BIS     @TIE,@DZVCSR    ;ENABLE INTERRUPTS
 8562  012704  106427  000000                        MTPS    #0              ;LOWER PRIORITY
 8563  012710  000001                        19$:    WAIT                    ;ALLOW INTERRUPTS
 8564  012712  077102                                SOB     R1,19$          ;REDUCE COUNT. ALL CHARACTERS SENT?
 8565  012714  042777  050040  167066                BIC     @SILOEN!MSENAB!TIE,@DZVCSR ;RESET SILO COUNTER, CLEAR STROBE
 8566  012722  104401                                SCOP1                   ;LOOP AGAIN?
 8567  012724  000742                                BR      17$             ;IF NOT, RETURN TO WHERE YOU LEFT OFF
 8568  012726  112777  000252  167100        20$:    MOVB    #252,@DZVTDR    ;SEND A CHARACTER
 8569  012734  000002                                RTI                     ;ALLOW MORE CHARACTERS TO COME
 8570                                                ;********************** TEST 2 ****************************
 8571                                                ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
 8572                                                ;*RECEIVER INTERRUPTS AND THAT ON THE
 8573                                                ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
 8574                                                ;*INTERRUPT WITH "RIE" SET.
 8575                                                ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
 8576                                                ;*ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.
 8578                                        ;:*  TEST 2
  (5)                                       ;;**********************************************************************
  (4)   012736  000004                        TST2:   SCOPE
  (2)   012740  012737  000002  001246                MOV     #2,@TSTNM       ;LOAD THE NUMBER OF THIS TEST
  (2)   012746  012737  013236  001362                MOV     #TST3,NEXT      ;POINT TO THE START OF THE NEXT TEST
 8579   012754  012737  013000  001364                MOV     #3$,LOCK        ;SET FOR LOOP
 8580   012762  104417                                DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
 8581   012764  104421                                LPRSET                  ;LOAD LINE PARAMETERS
 8582   012766  005037  001374                        CLR     SAVLIN          ;INIT LINE INDICATOR
 8583   012772  104422                                @UFSET                  ;ZERO DATA BUFFER
 8584   012774  012702  000001                        MOV     #1,R2           ;LINE POINTER
 8585   013000  012777  013210  167032  3$:           MOV     #11$,@DZVRIV    ;SET FOR UNEXPECTED INTER.
 8586   013006  012777  000200  167026                MOV     #MASK,@DZVRIS   ;SET PRIO.
 8587   013014  052777  010140  166766                BIS     #MSENAB!SILOEN!RIE,@DZVCSR
 8588                                                                        ;START SCANNER & SET SILO ENABLE
 8589   013022  030237  001366                        BIT     R2,LINE         ;VALID LINE?
 8590   013026  001477                                BEQ     18$             ;IF NOT GO TO NEXT LINE
 8591   013030  005777  166760                        TST     @DZVRBUF        ;EMPTY THE SILO
 8592   013034  100775                                BMI     .-4             ;BR IF DATA VALID IS SET!
 8593   013036  106427  000000                        MTPS    #0              ;SET PROCESSOR PRIORITY TO 0
 8594   013042  013700  001374                        MOV     SAVLIN,R0       ;MAKE OFFSET
 8595   013046  006300                                ASL     R0              ;MAKE POWER OF TWO
 8596   013050  010277  166750                        MOV     R2,@DZVTCR      ;SET TCR BIT
 8597   013054  005004                        5$:     CLR     R4
 8598   013056  005777  166726                6$:     TST     @DZVCSR
 8599   013062  100404                                BMI     7$
 8600   013064  104414                                DELAY
 8601   013066  005204                                INC     R4
 8602   013070  001372                                BNE     6$
 8603   013072  104003                                ERROR*  3               ;*TRDY FAILED TO SET
 8604   013074  116077  001426  166732  7$:           MOVB    TDO(R0),@DZVTDR ;LOAD A CHARACTER
 8605   013102  005260  001426                        INC     TDO(R0)         ;SET UP NEXT CHARACTER
 8606   013106  022760  000017  001426                CMP     #15.,TDO(R0)    ;15 CHARS YET?
 8607   013114  001406                                BEQ     8$
 8608   013116  032777  020000  166664                BIT     #SILOAL,@DZVCSR ;SILO ALARM = 0 ?
 8609   013124  001401                                BEQ     .+4             ;YES
 8610   013126  104013                                ERROR*  13              ;*SILO ALARM SHOULD NOT = 1
 8611                                                                        ;UNTIL 16. DATA CHARACTERS
 8612   013130  000752                                BR      6$
```

```
 8613  013132  012777  013216  166700  8$:     MOV     #12$,@DZVRIV    ;SET NEW VECTOR
 8614  013140  005777  166644                  TST     @DZVCSR         ;READY FOR 16TH CHAR
 8615  013144  100375                          BPL     .-4
 8616  013146  016077  001426  166660          MOV     TDO(R0),@DZVTDR ;LOAD THE 16TH CHAR.
 8617  013154  005004                          CLR     R4
 8618  013156  032777  020000  166624  9$:     BIT     #SILOAL,@DZVCSR
 8619  013164  001005                          BNE     10$
 8620  013166  104414                          DELAY
 8621  013170  005204                          INC     R4
 8622  013172  001371                          BNE     9$
 8623  013174  104014                          ERROR*  14              ;*SILO ALARM FAILED TO SET!
 8624  013176  000410                          BR      17$             ;SILO ALARM SHOULD =1 AFTER 16.
 8625                                                                  ;DATA CHARACTERS
 8626  013200  000240                  10$:    NOP                     ;STALL
 8627  013202  000240                          NOP
 8628  013204  104027                          ERROR*  27              ;SILO ALARM NOT INTERRUPTING.
 8629  013206  000404                          BR      17$             ;CONTINUE TEST.
 8630  013210  022626                  11$:    POP2SP                  ;FAKE RTI
 8631  013212  104012                          ERROR*  12              ;RX SHOULD NOT INTERRUPT
 8632  013214  000401                          BR      17$             ;CONTINUE
 8633  013216  022626                  12$:    POP2SP                  ;GOOD INTERRUPT TO HERE.
 8634  013220  040277  166600          17$:    BIC     R2,@DZVTCR      ;CLR TCR BIT
 8635  013224  104401                          SCOP1                   ;LOOP?
 8636  013226  005237  001374          18$:    INC     SAVLIN          ;INC EXPECTED LINE
 8637  013232  104420                          SHIFT                   ;NEXT LINE
 8638  013234  000661                          BR      3$              ;YES
```

```
                                                ;*********************** TEST 3 ****************************
                                                ;*THIS TEST RUNS ALL LINES FULL BORE
                                                ;*BASED UPON QUALIFIED LINES
                                                ;*..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
                                                ;*TRANSMITTER
                                        ;;*  TEST 3
                                        ;;*************************************************************
013236  000004                          TST3:   SCOPE
013240  012737  000003  001246                  MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013246  012737  014000  001362                  MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
013254  104417                                  DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
013256  013737  001366  013776                  MOV     LINE,RXTCR      ;SET IMAGE OF TCR BITS
013264  013737  001366  013516                  MOV     LINE,TXTCR      ;SET IMAGE OF TCR BITS
013272  104421                                  LPRSET                  ;LOAD LINE PARAMETERS
013274  104422                                  BUFSET                  ;ZERO DATA BUFFER
013276  012777  013520  166534                  MOV     #RXSVC,@DZVRIV  ;SET UP REC INTR VECTOR
013304  012777  000200  166530                  MOV     #MASK,@DZVRIS   ;STATUS
013312  012777  013404  166524                  MOV     #TXSVC,@DZVTIV  ;SET UP TRANS INTR VECTOR
013320  012777  000200  166520                  MOV     #MASK,@DZVTIS   ;STATUS
013326  052777  040140  166454                  BIS     #MSENAB!RIE!TIE,@DZVCSR ;SET MASTER SCAN ENABLE
013334  113777  001366  166462                  MOVB    LINE,@DZVTCR    ;SET TCR BITS
013342  106427  000000                          MTPS    #CLEAR          ;ALLOW INTERRUPTS

013346  005037  013402                  SNAP:   CLR     4$              ;CLEAR DELAY COUNTER
013352  104414                          2$:     DELAY                   ;WAIT FOR RECEIVERS TO FINISH
013354  105737  013776                          TSTB    RXTCR           ;WAIT FOR ALL RECIEVERS TO FINISH
013360  001002                                  BNE     3$              ;
013362  000137  013724                          JMP     OUT
013366  005237  013402                  3$:     INC     4$              ;INCREMENT DELAY COUNTER
013372  001367                                  BNE     2$              ;DELAY FINISHED?
013374  104007                                  ERROR+  7               ;*TRANSMITTER FAILED TO INTERRUPT
013376  104011                                  ERROR+  11              ;*RECEIVER FAILED TO INTERRUPT
013400  104400                                  ADVANCE                 ;LEAVE THIS TEST
013402  000000                          4$:     0

                                        ;TRANS INTR SVC ROUTINE
013404  117703  166402                  TXSVC:  MOVB    @DZVCSR,R3      ;FIND LINE NO.
013410  042703  177774                          BIC     #^C<3>,R3       ;ISOLATE LINE NO.
013414  010304                                  MOV     R3,R4           ;SAVE LINE NO.
013416  010337  001374                          MOV     R3,SAVLIN       ;SAVE LINE NO.
013422  005777  166362                          TST     @DZVCSR         ;TRANS READY SET ?
013426  100401                                  BMI     .+4
013430  104003                                  ERROR+  3               ;*TRANSMITTER FAILED
013432  012702  000001                          MOV     #1,R2           ;SET UP POSITION POINTER
013436  105303                          3$:     DECB    R3              ;IS IT THIS LINE ?
013440  100402                                  BMI     4$              ;YES
013442  006302                                  ASL     R2              ;UP THE LINE #
013444  000774                                  BR      3$              ;GO 'ROUND AGAIN
013446  030237  001366                  4$:     BIT     R2,LINE         ;VALID LINE?
013452  001001                                  BNE     .+4             ;YES
013454  104015                                  ERROR+  15              ;NO,INVALID LINE!!!!
013456  030237  013516                          BIT     R2,TXTCR        ;DATA FINISHED?
013462  001003                                  BNE     6$              ;IF NOT SEND CHAR.
013464  040277  166334                          BIC     R2,@DZVTCR      ;CLEAR TCR BIT
```

```
 8693  013470  000411                          BR      5$              ;RETURN
 8694  013472  006304                  6$:     ASL     R4              ;MAKE POWER OF 2
 8695  013474  116477  001426  166332          MOVB    TDO(R4),@DZVTDR ;LOAD CHARACTER
 8696  013502  105264  001426                  INCB    TDO(R4)         ;SET UP NEXT CHARACTER
 8697  013506  001002                          BNE     5$              ;LAST CHARACTER ?
 8698  013510  040237  013516                  BIC     R2,TXTCR        ;INDICAT LINE FINISHED
 8699  013514  000002                  5$:     RTI
 8700
 8701  013516  000000                  TXTCR:  0
 8702
 8703                                  ;REC INTR SVC ROUTINE
 8704  013520  105777  166264          RXSVC:  TSTB    @DZVCSR         ;REC DONE ?
 8705  013524  100401                          BMI     .+4             ;YES
 8706  013526  104004                          ERROR+  4               ;FALSE INTERRUPT
 8707  013530  032777  020000  166252          BIT     @SILOAL,@DZVCSR ;SILO ALARM?
 8708  013536  001401                          BEQ     .+4             ;NO
 8709  013540  104013                          ERROR+  13              ;SILO ALARM SHOULD NOT =1
 8710  013542  017704  166246                  MOV     @DZVRBUF,R4     ;SAVE IT
 8711  013546  010403                          MOV     R4,R3
 8712  013550  000303                          SWAB    R3
 8713  013552  042703  177774                  BIC     #^C<3>,R3       ;STRIP JUNK
 8714  013556  010337  001374                  MOV     R3,SAVLIN       ;SAVE LINE NUMBER
 8715  013562  005704                          TST     R4              ;DATA VALID?
 8716  013564  100401                          BMI     4$              ;IF YES SKIP ERROR PRINTOUT
 8717  013566  104023                          ERROR+  23              ;YOU LOSE  ...DATA VALID WAS'NT SET
 8718  013570  032704  040000          4$:     BIT     #OVRRUN,R4      ;TEST FOR OVERRUN
 8719  013574  001401                          BEQ     1$              ;IF NO OVERRUN SKIP ERROR
 8720  013576  104024                          ERROR+  24              ;DATA OVERRUN
 8721  013600  032704  020000          1$:     BIT     #FRMERR,R4      ;DATA FRAMING ERROR
 8722  013604  001401                          BEQ     2$              ;IF NO FRAMING ERROR CONTINUE
 8723  013606  104025                          ERROR+  25              ;FRAMING ERROR
 8724  013610  032704  010000          2$:     BIT     #PARER,R4       ;TEST FOR PARITY ERROR
 8725  013614  001401                          BEQ     3$              ;BRANCH IF NO ERROR
 8726  013616  104026                          ERROR+  26              ;TYPE OUT PARITY ERROR
 8727  013620  012702  000001          3$:     MOV     #1,R2           ;SET UP POSITION POINTER
 8728  013624  105303                  5$:     DECB    R3
 8729  013626  100402                          BMI     6$
 8730  013630  006302                          ASL     R2              ;RE POSITION POINTER
 8731  013632  000774                          BR      5$              ;GO 'ROUND AGAIN
 8732  013634  030237  001366          6$:     BIT     R2,LINE         ;LINE VALID ?
 8733  013640  001001                          BNE     .+4             ;YES
 8734  013642  104015                          ERROR+  15              ;INVALID LINE #
 8735  013644  013703  001374                  MOV     SAVLIN,R3       ;GET THE LINE NUMBER AGAIN
 8736  013650  006303                          ASL     R3              ;USE R3 AS A POINTER IN THE DATA TABLE
 8737  013652  126304  001436                  CMPB    TRO(R3),R4      ;DOES THE DATA CHARACTER COMPARE ?
 8738  013656  001410                          BEQ     7$              ;YES
 8739  013660  013705  001374                  MOV     SAVLIN,R5       ;MOVE LINE NO INTO EXPECTED
 8740  013664  000305                          SWAB    R5              ;ADJUST TO HIGH BYTE
 8741  013666  052705  100000                  BIS     #DVALID,R5      ;SET DVALID IN EXPECTED
 8742  013672  056305  001436                  BIS     TRO(R3),R5      ;SET DATA IN EXPECTED
 8743  013676  104005                          ERROR+  5               ;=NO. DATA DOES NOT COMPARE
 8744  013700  005263  001436          7$:     INC     TRO(R3)         ;SET UP FOR NEXT CHARACTER
 8745  013704  105763  001436                  TSTB    TRO(R3)         ;ALL CHARS DONE?
 8746  013710  001002                          BNE     .+6
 8747  013712  040237  013776                  BIC     R2,RXTCR        ;ZERO LINE DONE INDICATOR
 8748  013716  012716  013346                  MOV     #SNAP,(SP)      ;RESET THE BACKGROUND TIMING LOOP
```

```
 8749  013722  000002                          RTI
 8750
 8751
 8752                                          ;FINISH UP ROUTINE
 8753  013724  106427  000200          OUT:    MTPS    #MASK           ;STOP ALL INTERRUPTS
 8754  013730  104413                          DEVICE.CLR              ;CLEAR ALL INTERRUPTS AWAY
 8755  013732  005003                          CLR     R3
 8756  013734  005037  001374                  CLR     SAVLIN
 8757  013740  012702  000001                  MOV     #1,R2
 8758  013744  030237  001366          1$:     BIT     R2,LINE         ;VALID LINE ?
 8759  013750  001405                          BEQ     2$              ;NO
 8760  013752  022763  000400  001436          CMP     #400,TRO(R3)    ;RECEIVED A BINARY COUNT PATTERN ?
 8761  013760  001401                          BEQ     .+4             ;YES
 8762  013762  104030                          ERROR*  30              ;THE LINE FAILED TO RECEIVE A FULL
 8763                                                                  ;BINARY COUNT PATTERN
 8764  013764  005237  001374          2$:     INC     SAVLIN          ;SET UP FOR NEXT LINE
 8765  013770  005723                          TST     (R3)+           ;ADD 2
 8766  013772  104420                          SHIFT                   ;SET UP NEXT LINE POINTER
 8767  013774  000763                          BR      1$              ;FINISHED ?
 8768  013776  000000          RXTCR:  0                               ;RX IMAGE OF TCR BITS
 8769
 8770
 8771                                          ;**********************  TEST 4  ******************************
 8772                                          ;*DZV11 RELATIVE TIMING TEST.
 8773                                          ;*EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
 8774                                          ;*AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
 8775                                          ;*WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
 8776                                          ;*DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
 8777                                          ;*THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
 8778                                          ;* AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
 8779                                          ;*PARAMETERS ARE:
 8780                                          ;*  EIGHT BITS/PER/CHAR - TWO STOP BITS AT
 8781                                          ;*       50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
 8782                                          ;*       2400, 3600, 4800, 7200, 9600 BAUD.
 8783                                          ;*  19.2 K BAUD - TWO STOP BITS AT
 8784                                          ;*       SEVEN, SIX, FIVE BITS/PER/CHAR.
 8785                                          ;*AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
 8786                                          ;*THE NEXT SELECTED LINE IS THEN TESTED.
 8787                                          ;*WHEN RUNNING UNDER THE APT MANUFACTURING SYSTEM
 8788                                          ;*THIS TEST IS ONLY RUN THE FIRST PASS
 8790                                  ;;*  TEST 4
  (5)                                  ;;**************************************************************
  (4)  014000  000004                  TST4:   SCOPE
  (2)  014002  012737  000004  001246          MOV     #4,@TSTNM       ;LOAD THE NUMBER OF THIS TEST
  (2)  014010  012737  014416  001362          MOV     #TST5,NEXT      ;POINT TO THE START OF THE NEXT TEST
  (1)  014016  012737  014136  001364          MOV     #3$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
 8791  014024  132737  000001  001140          BITB    #1,@ENV         ;RUNNING UNDER APT?
 8792  014032  001405                          BEQ     10$             ;IF NOT CONTINUE WITH TEST
 8793  014034  005737  001126                  TST     @PASS           ;IF YES IS THIS FIRST PASS
 8794  014040  001402                          BEQ     10$             ;IF NOT 1ST PASS SKIP TEST
 8795  014042  000177  165314                  JMP     @NEXT
 8796  014046  012737  000002  001354  10$:    MOV     #2,@TIMES       ;SET UP FOR 2 ITERATIONS
 8797  014054  005037  015420                  CLR     OFFSET          ;RESET THIS VARIABLE
 8798  014060  005037  001374                  CLR     SAVLIN          ;RESET LINE NUMBER INDICATOR
 8799  014064  005037  001376                  CLR     XMTLIN          ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
 8800  014070  012702  000001                  MOV     #1,R2           ;USE R2 AS A BIT POINTER
```

```
014074 012703 010070                  MOV     #RCVON!S50!EIGHT!TWOSTOP,R3 ;BUILD TEMPORARY PARAMETERS
014100 030237 001366          1$:     BIT     R2,LINE             ;IS THIS LINE ACTIVE?
014104 001014                         BNE     3$                  ;IF SO, GO GET STARTED
014106 012703 010070          2$:     MOV     #RCVON!S50!EIGHT!TWOSTOP,R3 ;LOAD PARAMETERS TEMPORARILY
014112 005237 001376                  INC     XMTLIN              ;POINT TO THE NEXT LINE TO TRANSMIT
014116 042703 000007                  BIC     #7,R3               ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
014122 053703 001376                  BIS     XMTLIN,R3           ;ADD DESIRED LINE NUMBER
014126 005037 015420                  CLR     OFFSET
014132 104420                         SHIFT                       ;POINT TO THE NEXT LINE
014134 000761                         BR      1$                  ;PROCESS THE NEXT LINE
014136                        3$:
014136 104417                         DCLASM                      ;CLEAR DEVICE AND SET MAINT BIT IF 1 MODE
014140 042703 010000                  BIC     #RCVON,R3           ;ZERO PARAMTERS FOR TX LINE
014144 010377 165650                  MOV     R3,@DZVLPR          ;LOAD PARAMTERS FOR TX
014150 005737 001372                  TST     MODE                ;STAGGERED?
014154 100007                         BPL     100$                ;BR IF NO
014156 000241                         CLC                         ;SET UP LINE
014160 006003                         ROR     R3                  ;
014162 103002                         BCC     98$                 ;BR IF LINE WAS EVEN
014164 000241                         CLC                         ;PREPARE TO MKE LINE EVEN
014166 000401                         BR      99$                 ;CONTINUE
014170 000261                 98$:    SEC                         ;PREPARE TO MAKE LINE ODD
014172 006103                 99$:    ROL     R3                  ;SET ALTERED LINE
014174 052703 010000          100$:   BIS     #RCVON,R3           ;SET RX ON
014200 010377 165614                  MOV     R3,@DZVLPR          ;LOAD RX PARAMETERS
014204 010337 001374                  MOV     R3,SAVLIN           ;SET FOR RECEIV. LINE
014210 042737 177774 001374           BIC     #^C<3>,SAVLIN       ;ISOLATE LINE NO.
014216 042703 000003                  BIC     #3,R3               ;CLEAR OLD LINE #
014222 053703 001376                  BIS     XMTLIN,R3           ;SET LINE UP AGAIN
014226 010337 001402                  MOV     R3,REGIST           ;SAVE PARAMETERS FOR PRINTOUT
014232 104422                         BUFSET                      ;ZERO DATA BUFFER
014234 005037 001342                  CLR     $TMP0               ;USE $TMP0 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
014240 005037 001344                  CLR     $TMP1               ;INITIALIZE THE TIMER
014244 005037 001350                  CLR     $TMP3               ;INITIALIZE THESE BITS ALSO
014250 012737 000020 001400           MOV     #20,XMTCNT          ;SET HOW MANY CHARACTERS TO TRANSMIT
014256 012777 015046 165560           MOV     #XMTSRV,@DZVTIV
014264 012777 015216 165546           MOV     #RXISR1,@DZVRIV
014272 012777 000200 165542           MOV     #MASK,@DZVRIS
014300 012777 000200 165540           MOV     #MASK,@DZVTIS
014306 110277 165512                  MOVB    R2,@DZVTCR          ;START THE VALID LINE
014312 052777 040140 165470           BIS     #TIE!RIE!MSENAB,@DZVCSR
014320 106427 000000                  MTPS    #0                  ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
014324 032777 000100 165456   4$:     BIT     #RIE,@DZVCSR        ;IS ROUTINE DONE?
014332 001407                         BEQ     5$                  ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
014334 005237 001344                  INC     $TMP1               ;INCREMENT TIMER
014340 001371                         BNE     4$                  ;WHEN IT OVERFLOWS
014342 005237 001350                  INC     $TMP3               ;CATCH CARRY
014346 001366                         BNE     4$                  ;CONTINUE TEST
014350 104011                         ERROR+  11                  ;INTERRUPTS NOT FINISHED
014352 004737 007322          5$:     JSR     PC,SERV.G           ;<^G>?
014356 104401                         SCOP1                       ;LOOP?
014360 062737 000002 015420           ADD     #2,OFFSET
014366 022703 017400                  CMP     #17400,R3
014372 003006                         BGT     6$
014374 032703 000030                  BIT     #BIT4+BIT3,R3       ;IS  CHARACTER SIZE DONE?
014400 001642                         BEQ     2$
```

```
 8856  014402  162703  000010                    SUB     #BIT3,R3
 8857  014406  000653                            BR      3$
 8858  014410  062703  000400          6$:       ADD     #400,R3
 8859  014414  000650                            BR      3$
 8860                                            ;********************** TEST 5 ******************************
  (1)                                            ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
  (1)                                            ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
  (1)                                            ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
  (1)                                            ;*YOU ARE IN "STAGGERED" MODE.
  (1)                                            ;*40(8) CHARS ARE USED FOR THIS TEST.
  (1)                                            ;*ALL SELECTED LINES WILL BE ENABLED AT THE SAME TIME.
  (1)                                            ;*THIS TEST FIRST CHECKS EVEN PARITY FOR ODD LINES AND
  (1)                                            ;*ODD PARITY FOR EVEN LINES, THEN IT CHECKS THE REVERSE.
  (3)                                    ;:*  TEST 5
  (6)                                    ;;*****************************************************************
  (5)  014416  000004                    TST5:   SCOPE
  (3)  014420  012737  000005  001246            MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
  (2)  014426  012737  004250  001362            MOV     #$EOP,NEXT      ;POINT TO THE END-OF-PASS HANDLER
  (1)  014434  005737  001372                    TST     MODE            ;IS THIS STAGGERED MODE?
  (1)  014440  100131                            BPL     17$             ;IF NOT, DON'T DO THIS TEST
  (1)  014442  105037  001425                    CLRB    DONFLG          ;SET UP FOR FIRST TEST PASS
  (1)  014446  104413                    14$:    DEVICE.CLR              ;SET DCLR IN CSR
  (1)  014450  013701  001370                    MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
  (1)  014454  042701  000200                    BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
  (1)  014460  052701  000100                    BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
  (1)  014464  012702  000001                    MOV     #1,R2           ;USE R2 AS A LINE POINTER
  (1)  014470  030237  001366            1$:     BIT     R2,LINE         ;IS THIS A VALID LINE?
  (1)  014474  001420                            BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
  (1)  014476  105737  001425                    TSTB    DONFLG          ;FIRST PASS THROUGH TEST?
  (1)  014502  001004                            BNE     15$             ;IF NO BRANCH
  (1)  014504  032701  000001                    BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
  (1)  014510  001006                            BNE     2$              ;IF IT'S ODD, USE EVEN PARITY
  (1)  014512  000403                            BR      16$             ;IF EVEN SET FOR ODD PARITY
  (1)  014514  032701  000001            15$:    BIT     #BIT0,R1        ;IF THE LINE  IS EVEN SET FOR EVEN PAR.
  (1)  014520  001402                            BEQ     2$              ;GO LOAD PARAMETER
  (1)  014522  052701  000200            16$:    BIS     #ODDPAR,R1      ;IF IT'S ODD, USE ODD PARITY
  (1)  014526  010177  165266            2$:     MOV     R1,@DZVLPR      ;LOAD THE LINE PARAMETER REGISTER
  (1)  014532  042701  000200                    BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
  (1)  014536  005201                    3$:     INC     R1              ;POINT TO THE NEXT LINE
  (1)  014540  006302                            ASL     R2              ;
  (1)  014542  032702  000020                    BIT     #BIT4,R2        ;ALL LINES DONE?
  (1)  014546  001750                            BEQ     1$              ;IF NOT, GO CHECK THE NEXT LINE
  (1)  014550  005037  001374                    CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
  (1)  014554  005037  001342                    CLR     $TMPO           ;USE $TMPO TO COUNT TOTAL NUMBER OF TRANSMISSIONS
  (1)  014560  005003                            CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
  (1)  014562  012737  000040  001400            MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
  (1)  014570  104422                            BUFSET                  ;ZERO BUFFER AREA
  (2)  014572  012777  015046  165244            MOV     #XMTSRV,@DZVTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
  (2)  014600  012777  014726  165232            MOV     #9$,@DZVRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
  (2)  014606  012777  000200  165226            MOV     #MASK,@DZVRIS   ;SET THE INTERRUPT VECTOR STATUS
  (2)  014614  012777  000200  165224            MOV     #MASK,@DZVTIS   ;SET TRANSMITTER INTERRUPT PRIORITY
  (2)  014622  052777  040140  165160            BIS     #RIE!TIE!MSENAB,@DZVCSR ;ENABLE THE DEVICE
  (1)  014630  113777  001366  165166            MOVB    LINE,@DZVTCR    ;ENABLE ALL SELECTED LINES
  (1)  014636  106427  000000                    MTPS    #0              ;ALLOW INTERRUPTS
  (1)  014642  005037  014720            4$:     CLR     7$
  (1)  014646  005037  014722                    CLR     8$
```

```
(1)  014652  032777  000100  165130  5$:     BIT     @RIE,@DZVCSR    ;WHEN RX DONE; RIE WILL =0
(1)  014660  001407                          BEQ     6$              ;BR IF ALL DONE
(1)  014662  005237  014720                  INC     7$
(1)  014666  001371                          BNE     5$
(1)  014670  105237  014722                  INCB    8$
(1)  014674  100366                          BPL     5$
(1)  014676  104011                          ERROR*  11              ;*RX FAILED TO FINISH (INTERRUPT)
(1)  014700  106427  000200          6$:     MTPS    @MASK           ;SHUT OFF INTERRUPTS
(1)  014704  105737  001425                  TSTB    DONFLG          ;IS THIS SECOND TEST PASS
(1)  014710  001005                          BNE     17$             ;IF SO GET OUT
(1)  014712  105237  001425                  INCB    DONFLG          ;INDICATE FIRST TEST PASS DONE
(1)  014716  000653                          BR      14$             ;START OVER
(1)  014720  000000                  7$:     0
(1)  014722  000000                  8$:     0
(1)  014724  104400                  17$:    ADVANCE
(1)
(1)                                          ;RECEIVER SERVICE ROUTINE
(1)
(1)  014726  01. 24  165062          9$:     MOV     @DZVRBUF,R4     ;GET THE CHARACTER
(1)  014732  010401                          MOV     R4,R1           ;COPY THE RECEIVED INFORMATION
(1)  014734  000301                          SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
(1)  014736  042701  177774                  BIC     #^C<3>,R1       ;ISOLATE THE LINE NUMBER
(1)  014742  010137  001374                  MOV     R1,SAVLIN       ;SET LINE INDIC. TO RECEIVING LINE
(1)  014746  005704                          TST     R4              ;IS DATA VALID SET?
(1)  014750  100401                          BMI     10$             ;IF YES DON'T PRINT ERROR
(1)  014752  104023                          ERROR*  23              ;DATA VALID NOT SET
(1)  014754  010105                  10$:    MOV     R1,R5           ;BUILD LINE NO. FOR
(1)  014756  000305                          SWAB    R5              ;EXPECTED DATA IN RECEIVER BUFFER
(1)  014760  006301                          ASL     R1              ;ADJUST R1 FOR OFFSET
(1)  014762  156105  001436                  BISB    TRO(R1),R5      ;LOAD CHARACTER IN EXPECTED
(1)  014766  052705  110000                  BIS     @DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
(1)  014772  020405                          CMP     R4,R5           ;DOES RECEIVED=EXPECTED
(1)  014774  001401                          BEQ     12$             ;IF YES DON'T PRINT ERROR
(1)  014776  104006                          ERROR*  6               ;*ERROR- DID NOT GET CORRECT INFORMATION
(1)  015000  005261  001436          12$:    INC     TRO(R1)         ;SET UP THE NEXT CHARACTER
(1)  015004  005203                          INC     R3              ;ADD TO THE TOTAL RECEIVED COUNT
(1)  015006  032777  040000  164774          BIT     #TIE,@DZVCSR    ;ARE TRANSMISSIONS DONE?
(1)  015014  001011                          BNE     13$             ;IF NO, GO RECEIVE SOME MORE
(1)  015016  023703  001342                  CMP     $TMPO,R3        ;ARE ALL CHARACTERS RECEIVED?
(1)  015022  001006                          BNE     13$             ;IF NO, GO RECEIVE SOME MORE
(1)  015024  042777  000100  164756          BIC     @RIE,@DZVCSR    ;DISABLE RECEIVER INTERRUPTS
(1)  015032  012716  014700                  MOV     #6$,(SP)        ;CRUNCH THE STACK
(1)  015036  000002                          RTI                     ;RETURN AND FINISH
(1)  015040  012716  014642          13$:    MOV     #4$,(SP)        ;CRUNCH THE STACK
(1)  015044  000002                          RTI                     ;GO BACK TO RECEIVER WAIT LOOP
```

```
8862
8863                                    ;TRANSMITTER INTERRUPT SERVICE
8864                                    ;----------------------------
8865
8866  015046  117701  164740          XMTSRV: MOVB    @#DZVCSR,R1     ;GET THE LINE NUMBER.
8867  015052  042701  177774                  BIC     #^C<3>,R1       ;CLEAR JUNK
8868  015056  013705  001374                  MOV     SAVLIN,R5       ;SAVE REC. LINE NO.
8869  015062  010137  001374                  MOV     R1,SAVLIN       ;LOAD TRANS LINE NO FOR ERROR PRINTOUT
8870  015066  006301                          ASL     R1              ;ADJUST R1 FOR OFFSET
8871  015070  023761  001400  001426          CMP     XMTCNT,TDO(R1)  ;HAVE ALL CHAR. BEEN SENT
8872  015076  003414                          BLE     6$              ;IF YES GO CLEAR TCR
8873  015100  005777  164704                  TST     @DZVCSR         ;TRDY SET?
8874  015104  100401                          BMI     2$              ;IF YES GO LOAD CHAR.
8875  015106  104003                          ERROR*  3               ;*TRANSMITTER NOT READY- FALSE INTERRUPT
8876  015110  116177  001426  164716  2$:     MOVB    TDO(R1),@DZVTDR ;LOAD THE CURRENT CHARACTER FOR THIS LINE
8877  015116  005261  001426                  INC     TDO(R1)         ;SET UP NEXT CHARACTER FOR THIS LINE
8878  015122  005237  001342                  INC     $TMP0           ;UP THE NUMBER OF TRANSMISSIONS
8879  015126  000415                          BR      7$              ;GO RETURN
8880  015130  012700  000001          6$:     MOV     #1,R0           ;SET UP A DESELECTION POINTER
8881  015134  006201                          ASR     R1              ;GET LINE NO. AGAIN
8882  015136  005301                  12$:    DEC     R1              ;REDUCE THE COUNT. WAS THIS THE LINE?
8883  015140  100402                          BMI     3$              ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
8884  015142  006300                          ASL     R0              ;MOVE THE POINTER TO THE NEXT LINE
8885  015144  000774                          BR      12$             ;GO CHECK THE NEXT LINE
8886  015146  140077  164652          3$:     BICB    R0,@DZVTCR      ;DISABLE THE LINE POINTED TO BY R0
8887  015152  001003                          BNE     7$              ;IF MORE LINES ARE ACTIVE , GO CONTINUE TRANSMIT
8888  015154  042777  040000  164626          BIC     #TIE,@DZVCSR    ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
8889  015162  010537  001374          7$:     MOV     R5,SAVLIN       ;RESTORE RECEIV. LINE
8890  015166  000002                          RTI                     ;RETURN TO THE TIMING LOOP
8891
8892                                    ; RELATIVE TIME BUILDING ROUTINE
8893                                    ; -----------------------------
8894
8895  015170  012737  000004  001346  BUILD:  MOV     #4,$TMP2        ;ROTATE 4 BITS BACK INTO $TMP1
8896  015176  006037  001350          1$:     ROR     $TMP3           ;GET THE BITS FROM $TMP3, THE HIGH BYTE
8897  015202  006037  001344                  ROR     $TMP1           ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
8898  015206  005337  001346                  DEC     $TMP2           ;INTO $TMP1 USING THE CARRY BIT WITH
8899                                                                  ;ROTATE INSTRUCTIONS
8900  015212  001371                          BNE     1$              ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
8901  015214  000207                          RTS     PC              ;RETURN TO CALLING TEST
8902
```

```
8904                                            ;RECEIVER SERVICE ROUTINE
8905
8906  015216  105777  164566          RXISR1: TSTB    @DZVCSR         ;IS THE RECEIVER REALLY READY?
8907  015222  100401                          BMI     1$              ;IF SO, GO SERVICE IT
8908  015224  104004                          ERROR*  4               ;*ERROR- RECEIVER DONE FLAG ISN'T SET
8909  015226  017704  164562          1$:     MOV     @DZVRBUF,R4     ;SAVE THE RECEIVER INFORMATION
8910  015232  100401                          BMI     2$              ;IF IT WAS VALID, GO PROCESS IT
8911  015234  104023                          ERROR*  23              ;ERROR- DATA VALID WASN'T SET
8912  015236  032704  040000          2$:     BIT     #OVRRUN,R4      ;OVERRUN ERROR FLAG SET?
8913  015242  001401                          BEQ     6$              ;IF NOT DON'T TYPE ERROR
8914  015244  104024                          ERROR*  24              ;OVERRUN ERROR
8915  015246  032704  020000          6$:     BIT     #FRAMERR,R4     ;FRAMING ERROR FLAG SET?
8916  015252  001401                          BEQ     9$              ;IF NOT DON'T TYPE ERROR
8917  015254  104025                          ERROR*  25              ;FRAMING ERROR
8918  015256  032704  010000          9$:     BIT     #PARER,R4       ;PARITY ERROR FLAG SET?
8919  015262  001401                          BEQ     3$              ;IF NOT, GO CONTINUE PROCESSING
8920  015264  104026                          ERROR*  26              ;ERROR- RECEIVER ERROR FLAG SET
8921  015266  013701  001374          3$:     MOV     SAVLIN,R1       ;CALCULATE THE DATA OFFSET
8922  015272  006301                          ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
8923  015274  120461  001436                  CMPB    R4,TRO(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
8924  015300  001407                          BEQ     4$              ;IF SO,GO CONTINUE PROCESSING
8925  015302  116105  001436                  MOVB    TRO(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
8926  015306  042705  177400                  BIC     #^C<377>,R5     ;ELIMINATE PROPAGATED SIGN
8927  015312  042704  177400                  BIC     #^C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
8928  015316  104005                          ERROR*  5               ;*DATA ERROR
8929  015320  005261  001436          4$:     INC     TRO(R1)         ;SET UP THE NEXT EXPECTED CHARACTER
8930  015324  122761  000020  001436          CMPB    #20,TRO(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
8931  015332  001031                          BNE     8$              ;IF NOT RETURN
8932  015334  126137  001436  001342          CMPB    TRO(R1),$TMP0   ;ALL CHARAC. RECEIVED?
8933  015342  001025                          BNE     8$              ;IF SO,GO DETERMINE THE TIMING
8934  015344  004737  015170                  JSR     PC,BUILD        ;GET THE RELATIVE TIME (SIGNIFICANT BITS)
8935  015350  013700  015420                  MOV     OFFSET,R0       ;GET POINTER
8936  015354  013760  001344  002050          MOV     $TMP1,TMTBL(R0) ; SAVE THIS TEST'S TIME
8937  015362  005737  015420                  TST     OFFSET          ;FIRST TEST?
8938  015366  001410                          BEQ     7$              ;IF NOT, GO CHECK THE TIME
8939  015370  005740                          TST     -(R0)           ;POINT TO THE PREVIOUS  TIME TAKEN
8940  015372  026037  002050  001344          CMP     TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
8941  015400  101003                          BHI     7$              ;IF SO, GO TO THE NEXT TEST
8942  015402  016005  002050                  MOV     TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
8943  015406  104021                          ERROR*  21              ;TIMING ERROR
8944  015410  042777  000140  164372  7$:     BIC     #RIE!MSENAB,@DZVCSR ;DISABLE THE DEVICE
8945  015416  000002                  8$:     RTI                     ;RETURN TO THE PROGRAM
8946  015420  000000                  OFFSET: 0
```

```

                                              ;ERROR TABLE
015422  000000                        .ERRTAB:       0          ;ERROR 0
015424  000000                                       0
015426  000000                                       0

015430  015650                                       EM1        ;ERROR
015432  016774                                       DH1
015434  017174                                       DT1

015436  015723                                       EM2        ;ERROR 2
015440  017020                                       DH2
015442  017206                                       DT2

015444  015751                                       EM3        ;ERROR 3
015446  017053                                       DH3
015450  017224                                       DT3

015452  016010                                       EM4        ;ERROR 4
015454  017053                                       DH3
015456  017224                                       DT3

015460  016037                                       EM5        ;ERROR 5
015462  017065                                       DH4
015464  017232                                       DT4

015466  016066                                       EM6        ;ERROR 6
015470  017065                                       DH4
015472  017232                                       DT4

015474  000000                                       0
015476  000000                                       0
015500  000000                                       0

015502  000000                                       0
015504  000000                                       0
015506  000000                                       0

015510  016125                                       EM11       ;ERROR 11
015512  017053                                       DH3
015514  017224                                       DT3

015516  000000                                       0
015520  000000                                       0
015522  000000                                       0

015524  016163                                       EM13       ;ERROR 13
015526  017053                                       DH3
015530  017224                                       DT3

015532  016214                                       EM14       ;ERROR 14
015534  017053                                       DH3
015536  017224                                       DT3

015540  016246                                       EM15       ;ERROR 15
015542  000000                                       0
```

```
9004  015544  000000                        0
9005
9006  015546  016310                        EM16
9007  015550  017053                        DH3
9008  015552  017224                        DT3
9009
9010  015554  016362                        EM17    ;ERROR 17
9011  015556  017053                        DH3
9012  015560  017224                        DT3
9013
9014  015562  016420                        EM20
9015  015564  017053                        DH3
9016  015566  017224                        DT3
9017
9018  015570  016461                        EM21    ;ERROR 21
9019  015572  017114                        DH5
9020  015574  017250                        DT5
9021
9022  015576  000000                        0
9023  015600  000000                        0
9024  015602  000000                        0
9025
9026  015604  016511                        EM23    ;ERROR 23
9027  015606  017053                        DH3
9028  015610  017224                        DT3
9029
9030  015612  016541                        EM24
9031  015614  017053                        DH3
9032  015616  017224                        DT3
9033
9034  015620  016567                        EM25
9035  015622  017053                        DH3
9036  015624  017224                        DT3
9037
9038  015626  016617                        EM26
9039  015630  017053                        DH3
9040  015632  017224                        DT3
9041
9042  015634  016646                        EM27
9043  015636  017053                        DH3
9044  015640  017224                        DT3
9045
9046  015642  016714                        EM30
9047  015644  017053                        DH3
9048  015646  017224                        DT3
```

```
9050
9051                                                ;ERROR MESSAGES
9055  015650 047200 020117 052502 EM1:    .ASCIZ  <200>/NO BUS REPLY RESPONSE FROM DZV11 REGISTER/
9056  015723    200 042522 044507 EM2:    .ASCIZ  <200>?REGISTER R/W FAILURE?
9057  015751    200 051124 047101 EM3:    .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
9058  016010 051200 041505 044505 EM4:    .ASCIZ  <200>/RECEIVER DONE NOT SET/
9059  016037    200 040504 040524 EM5:    .ASCIZ  <200>/DATA COMPARISON ERROR/
9060  016066 042200 053132 030461 EM6:    .ASCIZ  <200>/DZV11 «RECEIVER BUFFER» ERROR/
9061  016125    200 042522 042503 EM11:   .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
9062  016163    200 044523 047514 EM13:   .ASCIZ  <200>/SILO ALARM SET TOO SOON/
9063  016214 051600 046111 020117 EM14:   .ASCIZ  <200>/SILO ALARM FAILED TO SET/
9064  016246 040600 052103 047511 EM15:   .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
9065  016310 051200 040505 044504 EM16:   .ASCIZ  <200>/READING DZVRBUF DID NOT CLEAR SILO ALARM/
9066  016362 042200 052101 020101 EM17:   .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
9067  016420 051200 041505 044505 EM20:   .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
9068  016461    200 042522 040514 EM21:   .ASCIZ  <200>/RELATIVE TIMING ERROR./
9069  016511    200 040504 040524 EM23:   .ASCIZ  <200>/DATA VALID IS NOT SET!/
9070  016541    200 040504 040524 EM24:   .ASCIZ  <200>/DATA OVERRUN IS SET!/
9071  016567    200 051106 046501 EM25:   .ASCIZ  <200>/FRAMING ERROR OCCURRED/
9072  016617    200 040520 044522 EM26:   .ASCIZ  <200>/PARITY ERROR OCCURRED/
9073  016646 051600 046111 020117 EM27:   .ASCIZ  <200>/SILO ALARM FAILED TO CAUSE INTERRUPT/
9074  016714 046200 047111 020105 EM30:   .ASCIZ  <200>/LINE DID NOT RECEIVE FULL BINARY COUNT PATTERN/
9075
9076  016774 052200 040522 020120 DH1:    .ASCIZ  <200>/TRAP PC  DZV11 REG/
9077  017020 042600 050130 041505 DH2:    .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
9078  017053    200 044514 042516 DH3:    .ASCIZ  <200>/LINE NO./
9079  017065    200 054105 042520 DH4:    .ASCIZ  <200>/EXPECTED  FOUND  LINE/
9080  017114 052200 020130 044514 DH5:    .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/
9081
9082        017174                  .EVEN
9086                                        ;DATA TABLES FOR ERROR MESSAGES
9087  017174 000002                 DT1:    2
9088  017176    006     003                 .BYTE   6,3
9089  017200 001330                         $REG1
9090  017202    006     001                 .BYTE   6,1
9091  017204 001326                         $REG0
9092
9093  017206 000003                 DT2:    3
9094  017210    006     004                 .BYTE   6,4
9095  017212 001340                         $REG5
9096  017214    006     001                 .BYTE   6,1
9097  017216 001336                         $REG4
9098  017220    006     001                 .BYTE   6,1
9099  017222 001326                         $REG0
910C
9101  017224 000001                 DT3:    1
9102  017226    003     001                 .BYTE   3,1
9103  017230 001374                         SAVLIN
9104
9105  017232 000003                 DT4:    3
9106  017234    006     004                 .BYTE   6,4
9107  017236 001340                         $REG5
9108  017240    006     001                 .BYTE   6,1
9109  017242 001336                         $REG4
9110  017244    003     001                 .BYTE   3,1
9111  017246 001374                         SAVLIN
```

```
9112
9113  017250  000004                DT5:    4
9114  017252      003       005             .BYTE   3,5
9115  017254  001374                        SAVLIN
9116  017256      006       011             .BYTE   6,9.
9117  017260  001340                        $REGS
9118  017262      006       007             .BYTE   6,7
9119  017264  001344                        $TMP1
9120  017266      006       001             .BYTE   6,1
9121  017270  001402                        REGIST
9128                                        ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
9129                                        ;-------------------------------------- ----- -
9130
9131  017272  002450                DLYTBL: 2450                    ;TIME FOR   50 BAUD
9132  017274  001560                        1560                    ;TIME FOR   75 BAUD
9133  017276  001120                        1120                    ;TIME FOR  110 BAUD
9134  017300  000750                         750                    ;TIME FOR  134 BAUD
9135  017302  000660                         660                    ;TIME FOR  150 BAUD
9136  017304  000330                         330                    ;TIME FOR  300 BAUD
9137  017306  000150                         150                    ;TIME FOR  600 BAUD
9138  017310  000060                          60                    ;TIME FOR 1200 BAUD
9139  017312  000040                          40                    ;TIME FOR 1800 BAUD
9140  017314  000030                          30                    ;TIME FOR 2000 BAUD
9141  017316  000020                          20                    ;TIME FOR 2400 BAUD
9142  017320  000010                          10                    ;TIME FOR 3600 BAUD
9143  017322  000001                           1                    ;TIME FOR 4800 BAUD
9144  017324  000001                           1                    ;TIME FOR 7200 BAUD
9145  017326  000001                           1                    ;TIME FOR 9600 BAUD
9146  017330  000001                           1                    ;TIME OF DELAY FOR 19200 BAUD
9147
9148                                        ;DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
9149                                        ;FOR ALL TESTS TO FUNCTION CORRECTLY ON A LSI11.
```

```
 9151
 9152                                   .SBTTL  FALCON (KXT-11) UPGRADE ROUTINES.                       ;;GPA
 9153                                   ;
 9154                                   ; THE FOLLOWING ROUTINES HAVE BEEN ADDED TO ALLOW DIAGNOSTIC(S)
 9155                                   ; TO RUN ON A FALCON (KXT-11) BASED SYSTEM.
 9156                                   ; TO DETERMINE WHETHER WE'RE A FALCON OR NOT, WE'LL SIZE THE 1ST 3/4 OF
 9157                                   ; THE I/O PAGE (28K TO 31K).  FALCON HAS 2KW LOCAL RAM AT 28K(+4) TO 30K
 9158                                   ; AND A MACRO-ODT AT 30K TO 31K.  CONSEQUENTLY, ALL I/O DEVICES MUST
 9159                                   ; BE PLACED BETWEEN 174000 AND 177776.  ADDITIONALLY, WE'LL STRAP THE
 9160                                   ; EMT AND TRAP SERVICE LEVEL TO PRI6, AND SET THE HALT VECTOR SO THAT
 9161                                   ; WE CAN STOP THE SUCKER !!
 9162                                   ;
 9163                                   ; TO MINIMIZE THE IMPACT OF THESE CHANGES ON FINAL PROGRAM SIZE, THE
 9164                                   ; BULK OF THIS CODE IS PLACED IN THE FLOATING VECTOR SPACE (400-776).
 9165                                   ; IF THE CPU AT HAND IS A FALCON (KXT11), IT STAYS THERE (NO HARM DONE).
 9166                                   ; OTHERWISE, THE AREA IS RESTORED TO ITS ORIGINAL "TRAP-CATCHER" STATE.
 9167                                   ;
 9168  017332  005227  177777           FALCON: INC     #-1             ; ONCE-ONLY !!!                 ;;GPA
 9169  017336  001002                           BNE     1$                                              ;;GPA
 9170  017340  004737  000400                   CALL    KXTCHK          ; EXECUTE FALCON CHECK          ;;GPA
 9171  017344  005727                   1$:     TST     (PC)+           ; TEST FALCON FLAG...           ;;GPA
 9172  017346  000000                   KXTFLAG: 0                      ;...NZ = FALCON...              ;;GPA
 9173  017350  000207                           RETURN                  ;...AND RETURN TO CALLER...     ;;GPA
 9174
 9175          017352                           #SVPC= .                                                ;;GPA
 9176          000400                           .= 400                  ; RESTORE FROM 374:376 AT END   ;;GPA
 9177  000400  005037  017346           KXTCHK: CLR     KXTFLAG         ; ASSUME NOT FALCON.            ;;GPA
 9178  000404  013746  000004                   MOV     @#4,-(SP)       ; SAVE ERROR VECTOR.            ;;GPA
 9179  000410  012737  000504  000004           MOV     #2$,@#4         ; SET A TRAP CATCHER.           ;;GPA
 9180  000416  012700  160010                   MOV     #160010,R0      ; FALCON RAM STARTS AT 28K+4.   ;;GPA
 9181  000422  005720                   1$:     TST     (R0)+           ;                               ;;GPA
 9182  000424  000240                           240                                                     ;;GPA
 9183  000426  020027  174000                   CMP     R0,#174000      ; SIZE TO 31K.                  ;;GPA
 9184  000432  103773                           BLO     1$                                              ;;GPA
 9185  000434  010037  017346                   MOV     R0,KXTFLAG      ; MUST BE FALCON, SET THE FLAG  ;;GPA
 9186  000440  012700  000040                   MOV     #40,R0          ; GET PRI1 BIT...               ;;GPA
 9187  000444  040037  000006                   BIC     R0,@#6          ;...AND LOWER BUS-ERROR...      ;;GPA
 9188  000450  040037  000016                   BIC     R0,@#16         ;...BPT...                      ;;GPA
 9189  000454  040037  000022                   BIC     R0,@#22         ;...IOT...                      ;;GPA
 9190  000460  040037  000032                   BIC     R0,@#32         ;...EMT...                      ;;GPA
 9191  000464  040037  000036                   BIC     R0,@#36         ;...AND TRAP SERVICE TO PRI6    ;;GPA
 9192  000470  012737  170000  000140           MOV     #170000,@#140   ; ENABLE "BREAK" HALT.          ;;GPA
 9193  000476  012637  000004                   MOV     (SP)+,@#4       ; RESTORE ERROR VECTOR...       ;;GPA
 9194  000502  000207                           RETURN                  ;...AND RETURN.                 ;;GPA
 9195
 9196  000504  012716  000512           2$:     MOV     #3$,(SP)        ; TRAP -- NOT A FALCON...       ;;GPA
 9197  000510  000002                           RTI                     ;...CONTINUE.                   ;;GPA
 9198  000512  012637  000004           3$:     MOV     (SP)+,@#4       ; RESET ERROR VECTOR            ;;GPA
 9199  000516  012700  000402                   MOV     #402,R0         ; SET-UP TO RESTORE FLOATING... ;;GPA
 9200  000522  013701  000376                   MOV     @#376,R1        ;...VECTORS (400 - 776).        ;;GPA
 9201  000526  010602                           MOV     SP,R2           ; SAVE STACK POINTER IN R2      ;;GPA
 9202  000530  012704  000570                   MOV     #6$,R4
 9203  000534  014446                   4$:     MOV     -(R4),-(SP)     ; PUSH THE RESTORE CODE...      ;;GPA
 9204  000536  020427  000546                   CMP     R4,#5$          ;...ONTO THE STACK.             ;;GPA
 9205  000542  101374                           BHI     4$                                              ;;GPA
 9206  000544  010607                           MOV     SP,PC           ; AND EXECUTE IT.               ;;GPA
```

```
                                        FALCON (KXT-11) UPGRADE ROUTINES.                       ;;GPA

                                        ;
                                        ; THIS CODE IS RELOCATED TO AND EXECUTED IN THE STACK AREA.
                                        ;
000546  010060  177776          5$:     MOV     R0,-2(R0)       ; RESTORE .+2...                ;;GPA
000552  010110                          MOV     R1,(R0)         ;...HALT (OR IOT).              ;;GPA
000554  022020                          CMP     (R0)+,(R0)+                                     ;;GPA
000556  020027  000776                  CMP     R0,#776                                         ;;GPA
000562  101771                          BLOS    5$              ; LOOP 'TIL DONE                ;;GPA
000564  010206                          MOV     R2,SP           ; THEN RESTORE SP...            ;;GPA
000566  000207                          RETURN                  ;...AND RETURN TO CALLER        ;;GPA
000570                          6$:                                                             ;;GPA
                                        ;
                                        ; IF FALCON, THIS AREA IS FREE FOR ANY PROGRAM UNIQUE
                                        ; CHANGES OR DATA STRUCTURES.
                                        ; BE SURE IT DOESN'T GET SCREWED UP !!
                                        ;
                                        ; INIT $BASE AND $VECT1 AND TWEAK THE "$GETPAR" CALLING
                                        ; SEQUENCE TO ACCEPT THE VALID FALCON RANGE.
                                        ;
000570  023727  001174  160010  FALCINI: CMP     $BASE,@#BASE    ; IS $BASE VIRGIN ??            ;;GPA
000576  001003                          BNE     1$              ; SKIP NEXT IF NOT              ;;GPA
000600  012737  174040  001174          MOV     #174040,$BASE   ; YES, SET ENGINEERING DEFAULT  ;;GPA
000606  023727  001170  000300  1$:     CMP     $VECT1,@#VECT1  ; IS $VECT1 VIRGIN ??           ;;GPA
000614  001003                          BNE     2$              ; SKIP NEXT IF NOT              ;;GPA
000616  012737  000370  001170          MOV     #370,$VECT1     ; YES, SET ENGINEERING DEFAULT  ;;GPA
000624  012737  000670  002464  2$:     MOV     #3$,GETCSR+2    ; SUBSTITUE CSR TEXT...         ;;GPA
000632  012737  174000  002470          MOV     #174000,GETCSR+6                                ;;GPA
000640  012737  177770  002472          MOV     #177770,GETCSR+10 ;...AND VALID RANGE.          ;;GPA
000646  012737  000732  002510          MOV     #4$,GETVEC+2    ; SUBSTITUTE VECTOR TEXT...     ;;GPA
000654  005037  002514                  CLR     GETVEC+6                                        ;;GPA
000660  012737  000370  002516          MOV     #370,GETVEC+10  ;...AND VALID RANGE.            ;;GPA
000666  000207                          RETURN                  ; RETURN TO CALLER.             ;;GPA

000670  030600  052123  041440  3$:     .ASCIZ  <200>'1ST CSR ADDRESS (174000:177770) '         ;;GPA
000676  051123  040440  042104
000704  042522  051523  024040
000712  033461  030064  030060
000720  030472  033467  033467
000726  024460  000040

000732  030600  052123  053040  4$:     .ASCIZ  <200>'1ST VECTOR ADDRESS (000:370)     '        ;;GPA
000740  041505  047524  020122
000746  042101  051104  051505
000754  020123  030050  030060
000762  031472  030067  020051
000770  020040  000040

                                        .EVEN                                                   ;;GPA

                                        $FREE=  <1000-.>/2              ; FREE WORDS LEFT.      ;;GPA

                                        .=$SVPC                                                 ;;GPA
017352                          CORMAX:
                                .END
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
AUTO.S  011564          8456    8457#
AVECT1= 000300          8348#   8456    9230
AVECT2= 000000          8456
BINARD  006402          8457#
BIT0    = 000001        8456#   8860
BIT00   = 000001        8456#
BIT01   = 000002        8456#
BIT02   = 000004        8456#
BIT03   = 000010        8456#
BIT04   = 000020        8456#
BIT05   = 000040        8456#
BIT06   = 000100        8456#
BIT07   = 000200        8456#
BIT08   = 000400        8456#
BIT09   = 001000        8456#
BIT1    = 000002        8456#
BIT10   = 002000        8456#
BIT11   = 004000        8456#   8457
BIT12   = 010000        8456#
BIT13   = 020000        8456#
BIT14   = 040000        8456#   8457
BIT15   = 100000        8456#
BIT2    = 000004        8456#
BIT3    = 000010        8456#   8854    8856
BIT4    = 000020        8456#   8457    8854    8860
BIT5    = 000040        8456#   8457
BIT6    = 000100        8456#
BIT7    = 000200        8456#   8457
BIT8    = 000400        8456#
BIT9    = 001000        8456#
BPTVEC= 000014          8456#
BRK0    = 000400        8456#
BRK1    = 001000        8456#
BRK2    = 002000        8456#
BRK3    = 004000        8456#
BRW     004724          8456    8457#
BUFSET= 104422          8456#   8480    8583    8652    8830    8860
BUILD   015170          8895#   8934
CHRONT  006400          8457#*
CLEAR   = 000000        8456#   8659*
CNVRT   = 104412        8456#   8457
CONVRT= 104411          8456#   8457
CORMAX  017352          9251#   9252
CO0     = 000400        8456#
CO1     = 001000        8456#
CO2     = 002000        8456#
CO3     = 004000        8456#
CR      = 000015        8456#   8457
CRLF    = 000200        8456#   8457
CSRMAP  011572          8457#
CYCLE   010724          8456    8457#
DATABP  007152          8457#*
DATAHD  007140          8457#*
DCLASM= 104417          8456#   8477    8580    8648    8811
DCLR    = 000020        8456#   8457
DDISP   = 177570        8456#
```

```
DELAY - 104414          8456#   8495    8511    8521    8600    8620    8663
DEVADR  006124          8457#
DEVICE- 104413          8456#   8457    8754    8860
DH1     016774          8955    9076#
DH2     017020          8959    9077#
DH3     017053          8963    8967    8987    8995    8999    9007    9011    9015    9027    9031    9035    9039    9043
                        9047    9078#
DH4     017065          8971    8975    9079#
DH5     017114          9019    9080#
DISPLA  001306          8456#   8457#
DISPRE  000174          8456#
DLYCNT  006476          8456#   8457#
DLYTBL  017272          8456    9131#
DONFLG  001425          8456#   8860#
DSMR  - 177570          8456#
DTR0  - 000400          8456#
DTR1  - 001000          8456#
DTR2  - 002000          8456#
DTR3  - 004000          8456#
DT1     017174          8956    9087#
DT2     017206          8960    9093#
DT3     017224          8964    8968    8988    8996    9000    9008    9012    9016    9028    9032    9036    9040    9044
                        9048    9101#
DT4     017232          8972    8976    9105#
DT5     017250          9020    9113#
DVALID- 100000          8456#   8528    8741    8860
DZCR0   001500          8456#   8457
DZCR1   001512          8456#
DZCR10  001620          8456#
DZCR11  001632          8456#
DZCR12  001644          8456#
DZCR13  001656          8456#
DZCR14  001670          8456#
DZCR15  001702          8456#
DZCR16  001714          8456#
DZCR17  001726          8456#
DZCR2   001524          8456#
DZCR3   001536          8456#
DZCR4   001550          8456#
DZCR5   001562          8456#
DZCR6   001574          8456#
DZCR7   001606          8456#
DZVAC1  001406          8456#   8457
DZVCSR  002010          8456#   8457#   8482#   8488    8493    8503    8509    8533    8557#   8561#   8565#   8587#   8598
                        8608    8614    8618    8657#   8679    8704    8707    8840#   8842    8860#   8873    8888#   8906
                        8944#
DZVC0   001502          8456#   8457
DZVC1   001514          8456#
DZVC10  001622          8456#
DZVC11  001634          8456#
DZVC12  001646          8456#
DZVC13  001660          8456#
DZVC14  001672          8456#
DZVC15  001704          8456#
DZVC16  001716          8456#
DZVC17  001730          8456#
```

CVDZB D MACr11 30A(1052) 21-AUG-84 08:31 PAGE 31-3
CVDZBO.P11 21 AUG-84 08:28 CROSS REFERENCE TABLE - USER SYMBOLS

```
DZVC2   001526          8456#
DZVC3   001540          8456#
DZVC4   001552          8456#
DZVC5   001564          8456#
DZVC6   001576          8456#
DZVC7   001610          8456#
DZVLEV  011274          8457#
DZVLPR  002020          8456#   8457*   8813*   8824*   8860*
DZVMSR  002030          8456#   8457*
DZVMUM  001414          8456#*  8457*
DZVRBU  002014          8456#   8457*   8529    8543    8591    8710    8860    8909
DZVRIS  002042          8456#   8457*   8586*   8654*   8837*   8860*
DZVRIV  002040          8456#   8457*   8585*   8613*   8653*   8836*   8860*
DZVTCR  002024          8456#   8457*   8487*   8547*   8560*   8596*   8634*   8658*   8692*   8839*   8860*   8886*
DZVTDR  002034          8456#   8457*   8499*   8568*   8604*   8616*   8695*   8876*
DZVTIS  002046          8456#   8457*   8656*   8838*   8860*
DZVTIV  002044          8456#   8457*   8558*   8655*   8835*   8860*
DZV.EN  001740          8456#   8457
DZV.MA  001500          8456#   8457
EIGHT   000030          8456#   8801    8804
EIGHTS* 000070          8456#
EMTSAV  002134          8456#
EMTVEC* 000030          8456#
EM1     015650          8954    9055#
EM11    016125          8986    9061#
EM13    016163          8994    9062#
EM14    016214          8998    9063#
EM15    016246          9002    9064#
EM16    016310          9006    9065#
EM17    016362          9010    9066#
EM2     015723          8958    9056#
EM20    016420          9014    9067#
EM21    016461          9018    9068#
EM23    016511          9026    9069#
EM24    016541          9030    9070#
EM25    016567          9034    9071#
EM26    016617          9038    9072#
EM27    016646          9042    9073#
EM3     015751          8962    9057#
EM30    016714          9046    9074#
EM4     016010          8966    9058#
EM5     016037          8970    9059#
EM6     016066          8974    9060#
ERRMSG  007126          8457#*
ERROR * 104000          8456#   8490    8498    8505    8514    8532    8535    8546    8603    8610    8623    8628    8631
                        8669    8670    8681    8689    8706    8709    8717    8720    8723    8726    8734    8743    8762
                        8848    8860    8875    8908    8911    8914    8917    8920    8928    8943
ERRVEC* 000004          8456#   8457*
ERTABO  007306          8457#
EVEPAR* 000000          8456#
EXITER  007236          8457#
FALCIN  000570          8456*   9227#
FALCON  017332          8456*   9168#
FIVE  * 000000          8456#
FIVES * 000040          8456#
FRMERR* 020000          8456#   8721    8915
```

```
GETCSR- 002462          8456#   9233#   9234#   9235#
GETSWR- 007350          8456#   8457#
GETVEC- 002506          8456#   9236#   9237#   9238#
HALTS   007156          8457#
HDRFLG  001423          8456#
HDZVCS  002012          8456#   8457#   8675    8866
HDZVLP  002022          8456#   8457#
HDZVWS  002032          8456#   8457#
HDZVWB  002016          8456#   8457#
HDZVTC  002026          8456#   8457#
HDZVTD  002036          8456#   8457#
HILIM   006122          8457#
HT    - 000011          8456#   8457
INBUF   010620          8457#
INTFLG  001422          8456#   8457#
INSTER- 104404          8456#   8457
INSTR - 104403          8456#   8457
INSTR2  005722          8457#
IOTVEC- 000020          8456#   8457#
KXTCHK  000400          9170#   9177#
KXTFLA  017346          8456    9172#   9177#   9185#
LF    - 000012          8456#   8457
LIMITS  006050          8457#
LINE    001366          8456#   8457#   8483    8589    8649    8650    8658    8687    8732    8758    8802    8860
LINE0   001504          8456#
LINE1   001516          8456#
LINE10  001624          8456#
LINE11  001636          8456#
LINE12  001650          8456#
LINE13  001662          8456#
LINE14  001674          8456#
LINE15  001706          8456#
LINE16  001720          8456#
LINE17  001732          8456#
LINE2   001530          8456#
LINE3   001542          8456#
LINE4   001554          8456#
LINE5   001566          8456#
LINE6   001600          8456#
LINE7   001612          8456#
LOBITS  006126          8457#
LOCK    001364          8456#   8457#   8476#   8579#   8790#
LOLIM   006120          8457#
LPRSET- 104421          8456#   8478    8581    8651
LP0   - 000000          8456#
LP1   - 000001          8456#
LP2   - 000002          8456#
LP3   - 000003          8456#
MAINT - 000010          8456#   8457
MANT0   001510          8456#   8457
MANT1   001522          8456#
MANT10  001630          8456#
MANT11  001642          8456#
MANT12  001654          8456#
MANT13  001666          8456#
MANT14  001700          8456#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
MANT15  001712           8456#
MANT16  001724           8456#
MANT17  001736           8456#
MANT2   001534           8456#
MANT3   001546           8456#
MANT4   001560           8456#
MANT5   001572           8456#
MANT6   001604           8456#
MANT7   001616           8456#
MASK  - 000200           8456#  8457    8586    8654    8656    8753#  8837    8838    8860#
MASTEK  010235           8457#
MBADLN  010350           8456    8457#
MCSRX   010165           8457#
MDATA   010662           8457#
MEPASS  010003           8457#
MERRPC  010315           8457#
MERRX   010212           8457#
MERR2   010043           8457#
MERR3   010112           8456    8457#
MLOCK   010136           8456    8457#
MNEW    010240           8456    8457#
MNTFLG  001424           8456#   8457#
MODE    001372           8456#   8457#  8526    8814    8860
MPASSX  010201           8457#
MPFAIL  007740           8457#
MR      010027           8456    8457#
MSENMB- 000040           8456#   8482    8557    8565    8587    8657    8840    8860    8944
MTITLE  001000           8456#
MTSTN   010223           8457#
MVECX   010173           8457#
NEXT    001362           8456#   8457    8475#  8578#  8647#  8790#  8795    8860#
NOLIST- ****** U         8452
ODDPAR- 000200           8456#   8860
OFFSET  015420           8797#   8808#  8851#  8935    8937    8946#
ONESTO- 000000           8456#
OUT     013724           8666    8753#
OVRRUN- 040000           8456#   8540    8718    8912
PAR     001370           8456#   8457#  8860
PARAM - 104405           8456#   8457
PARAM1  005770           8457#
PARER - 010000           8456#   8724    8860    8918
PARERR  006044           8457#
PARITY- 000100           8456#   8860
PARMD - 104415           8456#
PAR0    001506           8456#
PAR1    001520           8456#
PAR10   001626           8456#
PAR11   001640           8456#
PAR12   001652           8456#
PAR13   001664           8456#
PAR14   001676           8456#
PAR15   001710           8456#
PAR16   001722           8456#
PAR17   001734           8456#
PAR2    001532           8456#
PAR3    001544           8456#
```

```
PAR4    001556          8456#
PAR5    001570          8456#
PAR6    001602          8456#
PAR7    001614          8456#
PAUCH - 104416          8456#
PIRQ  - 177772          8456#
PIRQVE- 000240          8456#
POPRO - 012600          8456#
POP1SP- 005726          8456#
POP2SP- 022626          8456#   8457    8630    8633
PR0   - 000000          8456#
PR1   - 000040          8456#
PR2   - 000100          8456#
PR3   - 000140          8456#
PR4   - 000200          8456#
PR5   - 000240          8456#
PR6   - 000300          8456#
PR7   - 000340          8456#
PS    - 177776          8456#
PSW   - 177776          8456#
PUSHR0- 010046          8456#
PUSH1S- 005746          8456#
PUSH2S- 024646          8456#
PURVEC- 000024          8456#   8457#
RCVON - 010000          8456#   8801    8804    8812    8823
RDONE - 000200          8456#
REGIST  001402          8456#   8829#   9121
RESREG  007154          8457#
RESTAR  011270          8457#
RESVEC- 000010          8456#
RES05 - 104410          8456#   8457
RIE   - 000100          8456#   8587    8657    8840    8842    8860    8944
RING0 - 000001          8456#
RING1 - 000002          8456#
RING2 - 000004          8456#
RING3 - 000010          8456#
RL0   - 000000          8456#
RL1   - 000400          8456#
RL2   - 001000          8456#
RL3   - 001400          8456#
RUN     001412          8456#   8457#
RXISR1  015216          8836    8906#
RXSVC   013520          8653    8704#
RXTCR   013776          8649#   8664    8747#   8768#
SAVACT  001410          8456#   8457#
SAVLIN  001374          8456#   8479#   8485    8525    8549#   8582#   8594    8636#   8678#   8714#   8735    8739    8756#
                        8764#   8798#   8825#   8826#   8860#   8868    8869#   8889#   8921    9103    9111    9115
SAVN0   001416          8456#   8457
SAVNUM  001415          8456#   8457#
SAVPC   001404          8456#   8457#
SAV05 - 104407          8456#   8457
SAV30   002152          8456#   8457
SAV32   002154          8456#   8457
SCOPE - 000004          8456#   8457    8475    8578    8647    8790    8860
SCOP1 - 104401          8456#   8548    8566    8635    8850
SERV G  007322          8457#   8849
```

```
CVDZB-0 MACY11 30A(1052)  21-AUG-84  08:31  PAGE 31 7
CVDZBD.P11    21-AUG-84 08:28          CROSS REFERENCE TABLE    USER SYMBOLS

SETAPT  011436           8456#   8457#
SETFLG= 104406           8456#
SEVEN  = 000020          8456#
SEVENS= 000060           8456#
SHIFT  = 104420          8456#   8550    8637    8766    8809
SILONL = 020000          8456#   8503    8509    8533    8608    8618    8707
SILOEN= 010000           8456#   8482    8557    8565    8587
SIX    = 000010          8456#
SIXS   = 000050          8456#
SNAP     013346          8462#   8748
SPACNT  006401           8457#
STACK  = 001120          8456#   8457
STKLMT= 177774           8456#
STOP     001446          8456#   8457
SV05     006136          8457#
SWR      001304          8456#=  8457#
SWREG    000176          8456#
SW0    = 000001          8456#
SW00   = 000001          8456#
SW01   = 000002          8456#   8457
SW02   = 000004          8456#
SW03   = 000010          8456#
SW04   = 000020          8456#
SW05   = 000040          8456#
SW06   = 000100          8456#
SW07   = 000200          8456#
SW08   = 000400          8456#   8457
SW09   = 001000          8456#   8457
SW1    = 000002          8456#
SW10   = 002000          8456#   8457
SW11   = 004000          8456#
SW12   = 010000          8456#   8457
SW13   = 020000          8456#   8457
SW14   = 040000          8456#
SW15   = 100000          8456#
SW2    = 000004          8456#
SW3    = 000010          8456#
SW4    = 000020          8456#
SW5    = 000040          8456#
SW6    = 000100          8456#
SW7    = 000200          8456#
SW8    = 000400          8456#
SW9    = 001000          8456#
S110   = 001000          8456#
S1200  = 003400          8456#
S134   = 001400          8456#
S150   = 002000          8456#
S1800  = 004000          8456#
S19200= 007400           8456#
S2000  = 004400          8456#
S2400  = 005000          8456#
S300   = 002400          8456#
S3600  = 005400          8456#
S4800  = 006000          8456#
S50    = 000000          8456#   8801    8804
S600   = 003000          8456#
```

```
S7200  - 006400          8456#
S75    - 000400          8456#
S9600  - 007000          8456#
TBITVE - 000014          8456#
TCR0   - 000001          8456#
TCR1   - 000002          8456#
TCR2   - 000004          8456#
TCR3   - 000010          8456#
TD0      001426          8456#  8457    8499    8500#   8604    8605#   8606    8616    8695    8696#   8871    8876    8877#
TD1      001430          8456#
TD2      001432          8456#
TD3      001434          8456#
TEIGHT   002106          8456#
TFIVE    002114          8456#
TIE    - 040000          8456#  8561    8565    8657    8840    8860    8888
TKVEC  - 000060          8456#
TLAST  - 014416          8457   9126#
TL0    - 000U00          8456#
TL1    - 000400          8456#
TL2    - 001000          8456#
TL3    - 001400          8456#
TMTBL    002050          8456#  8936#   8940    8942
TPVEC  - 000064          8456#
TRAPVE - 000034          8456#
TRDY   - 100000          8456#
TRTVEC - 000014          8456#
TRO      001436          8456#  8737    8742    8744#   8745    8760    8860#   8923    8925    8929#   8930    8932
TR1      001440          8456#
TR2      001442          8456#
TR3      001444          8456#
TSEVEN   002110          8456#
TSIX     002112          8456#
TST1     012274          8457   8475#
TST2     012736          8475   8578#
TST3     013236          8578   8647#
TST4     014000          8647   8790#
TST5     014416          8790   8860#   9126
TST6   - ****** U        8860
TTST     004476          8456#  8457#
TWOSTO - 000040          8456#  8801    8804
TXSVC    013404          8655   8675#
TXTCR    013516          8650#  8690    8698#   8701#
TYPDAT   007142          8457#
TYPE   - 104402          8456#  8457
TYPMSG   007032          8457#
T110     002054          8456#
T1200    002066          8456#
T134     002056          8456#
T150     002060          8456#
T1800    002070          8456#
T2000    002072          8456#
T2400    002074          8456#
T300     002062          8456#
T3600    002076          8456#
T4800    002100          8456#
T50      002050          8456#
```

```
CVDZB D MACY11 30A(1052)  21-AUG 84  08:31  PAGE 31 9
CVDZ30.P11    21-AUG-84 08:28          CROSS REFERENCE TABLE    USER SYMBOLS

T600     002064            8456#
T7200    002102            8456#
T75      002052            8456#
T9600    002104            8456#
UFD    - 000040            8457#
UFDSET - 000001            8456#
VECMAP   012104            8457#
VMKOR    002156            8456#
WRDCNT   006376            8457#*
WTBS.F   007130            8457#
XBX      006720            8457#
XCSR     004434            8457#
XERR     004456            8457#
XHEAD    010322            8456    8457#
XMTCNT   001400            8456#   8834*   8860*   8871
XMTLIN   001376            8456#   8799*   8805*   8807    8828
XMTSRV   015046            8835    8860    8866#
XPASS    004450            8457#
XSTATQ   010412            8456    8457#
XTSTN    007314            8457#
XVEC     004442            8457#
XX     - 160210            8456#
YY     - 000500            8456#
ZZ     - 000020            8456#
$APTHD   001446            8456#
$ASTAT - ****** U          8457
$ATYC    005362            8457#
$ATY1    005336            8457#
$ATY3    005344            8457#
$ATY4    005354            8457#
$AUTOB   001300            8456#
$BASE    001174            8456#*  8457*   9227    9229*
$BNADR   001266            8456#
$BDDAT   001272            8456#
$CDW1    001200            8456#   8457
$CDW2    001202            8456#   8457*
$CHARC   005332            8457#*
$CMTAG   001244            8456#
$CM1   - 000006            8456#
$CM2   - 000014            8456#
$CM3   - 000006            8456#
$CM4   - 000005            8456#
$CPUDP   001146            8456#
$CRLF    001357            8456#   8457
$DDW0    001204            8456#   8457
$DDW1    001206            8456#
$DDW10   001230            8456#
$DDW11   001232            8456#
$DDW12   001234            8456#
$DDW13   001236            8456#
$DDW14   001240            8456#
$DDW15   001242            8456#
$DDW2    001210            8456#
$DDW3    001212            8456#
$DDW4    001214            8456#
$DDW5    001216            8456#
```

```
$DDM6   001220          84560#
$DDM7   001222          84560#
$DDM8   001224          84560#
$DDM9   001226          84560#
$DEVCT  001130          84560#  8457#
$DEVM   001176          84560#  8457#
$DOAGN  004430          8457#
$E    - 000007          68240#  8475#   8578#   8647#   8790#   8860#
$ENDAD  004420          8456    8457#
$ENDCT  004404          8457#
$ENV    001140          84560#  8457    8791
$ENVM   001141          84560#  8457
$EOP    004250          8457#   8860
$EOPCT  004376          8457#
$ERFLG  001247          84560#  8457#
$ERMAX  001261          84560#  8457
$ERROR  006664          8456    8457#
$ERRPC  001262          84560#  8457#
$ERRTB  001362          84560#
$ERTTL  001256          84560#  8457#
$ETABL  001140          84560#
$ETEND  001244          84560#
$FATAL  001122          84560#  8457#
$FFLG   005602          8457#
$FILLC  001322          84560#  8457
$FILLS  001321          84560#  8457
$FLIP - 177777          5671#   8462#   8474#   8570#   8577#   8641#   8646#   8771#   8789#   8860#
$FREE - 000002          9245#   9246
$GDADR  001264          84560#
$GDDAT  001270          84560#
$GET42  004410          8457#
$HD   - 000001          8456
$HIBTS  001446          84560#
$ICNT   001250          84560#  8457#
$ILLUP  007732          8457#
$INTAG  001301          84560#
$ITEMB  001260          84560#  8457#
$LF     001360          84560#  8457
$LFLG   005601          8457#
$LPADR  001252          84560#  8457#
$LPERR  001254          84560#
$MADR1  001152          84560#
$MADR2  001156          84560#
$MADR3  001162          84560#
$MADR4  001166          84560#
$MAIL   001120          84560#  8457
$MAMS1  001150          84560#
$MAMS2  001154          84560#
$MAMS3  001160          84560#
$MAMS4  001164          84560#
$MBADR  001450          84560#
$MFLG   005600          8457#
$MSGAD  001134          84560#  8457#
$MSGLG  001136          84560#  8457#
$MSGTY  001120          84560#  8457#
$MTYP1  001151          84560#
```

```
$MTYP2  001155          84560
$MTYP3  001161          84560
$MTYP4  001165          84560
$MXCNT  004726          84570
$N    - 000005          68230#  8462    8475#   8570    8578#   8641    8647#   8771    8790#   8860#   9123#   9124
$NULL   001320          84560#  8457
$NWTST- 000000          8475#   8578#   8647#   8790#   8860#
$OVER   004664          84570
$PASS   001126          84560#  8457#   8793
$PASTN  001454          84560
$PWRAD  007726          84570
$PWRDN  007566          8456    8457#
$PWRMG  007722          84570
$PWRUP  007640          84570
$QUES   001356          84560#  8457
$REGAD  001324          84560
$REG0   001326          84560#  8457#   9091    9099
$REG1   001330          84560#  8457#   9089
$REG2   001332          84560#  8457#
$REG3   001334          84560#  8457#
$REG4   001336          84560#  8457#   9097    9109
$REG5   001340          84560#  8457#   9095    9107    9117
$RTNAD  004432          84570
$SAVR6  007736          84570#
$SCOPE  004464          84570
$SETUP- 000000          8457
$SVLAD  004646          84570
$SVPC   017352          84560#  9175#   9250
$SWR    164000          56780#  8456    8457    8475    8578    8647    8790    8860
$SWREG  001142          84560#  8457
$SWRMX- 000000          8457
$TESTN  001124          84560#  8457#
$TIMES  001354          84560#  8457#   8796#
$TKB    001312          84560#  8457
$TKS    001310          84560#  8457
$TMP0   001342          84560#  8831#   8860#   8878#   8932
$TMP1   001344          84560#  8457    8832#   8844#   8897#   8936    8940    9119
$TMP2   001346          84560#  8457    8895#   8898#
$TMP3   001350          84560#  8833#   8846#   8896#
$TMP4   001352          84560#
$TN   - 000006          84560#  8475#   8578#   8647#   8790#   8860#
$TPB    001316          84560#  8457#
$TPFLG  001323          84560#  8457
$TPS    001314          84560#  8457
$TSTN   001452          84560#
$TSTNM  001246          84560#  8457#   8475#   8578#   8647#   8790#   8860#
$TYPE   005002          84570
$TYPEC  005214          84570
$TYPEX  005334          84570
$UNIT   001132          84560#
$UNITM  001456          84560#
$USWR   001144          84560#
$VECT1  001170          84560#  8457#   9230    9232#
$VECT2  001172          84560#
$XOFF - 000023          8457
$XON  - 000021          8457
```

CVDZB-D MACY11 30A(1052) 21-AUG-84 08:31 PAGE 31-12
CVDZBD.P11 21 AUG-84 08:28 CROSS REFERENCE TABLE USER SYMBOLS

```
$XTSTR  004524            8457#
$Y    = 000023            8456#
$$GET4= 000000            8457#
$40CAT= ****** U          8457
      = 017352            8456#  8457#  8489   8523   8592   8609   8615   8680   8688   8705   8708   8733   8746
                          8761   9082#  9175   9176#  9245   9250#
.ADVAN  006500            8456   8457#
.BEGIN  004154            8456#
 BUFSE  006570            8456   8457#
 CNVRT  006226            8456   8457#
.CONVR  006222            8456   8457#
.DCLAS  006446            8456   8457#
.DELAY  006460            8456   8457#
.DEVIC  006426            8456   8457#
.ERRTA  015422            8457   8950#
.INSTE  005710            8456   8457#
 INSTR  005604            8456   8457#
.INST1  005624            8457#
.LPRSE  006530            8456   8457#
.MSG    005626            8457#*
 PARAM  005730            8456   8457#
 PARMD  011434            8456   8457#
.PAMCH  010544            8456   8457#
.RESOS  006170            8456   8457#
 SAVOS  006130            8456   8457#
.SCOPE  004464            8456   8457#
.SCOP1  004730            8456   8457#
.SETFL  010424            8456   8457#
 SHIFT  006512            8456   8457#
 START  002116            8456#
 TRPSR  006404            8456   8457#
 TRPTA  001742            8456#  8457
 TYPE   004754            8456   8457#
 $ASTA= ****** U          8457
 $X   = 001446            8456#
```

COMMEN	15580	84560							
ENDCOM	15700	84560							
ESCAPE	16840	84560							
GETPRI	13150	84560							
GETSWR	17530	84560							
MULT	45310	84560							
NEWTST	16160	84560	8475	8578	8647	8790	8860		
PASEND	68550	8457							
POP	21270	84560	8457						
PRGEND	79200	8457							
PRGFRT	70050	8456							
PUSH	21190	84560	8457						
REPORT	54770	56770	84560						
SAVEMT	11670	56750	8456						
SC	67940	8457							
SC1	68030	8457							
SETPRI	12840	84560							
SETUP	13380	84560							
SKIP	17170	84560							
SLASH	15110	84560							
STARS	14810	84560	8457	8475	8578	8647	8790	8860	
SWRSU	14510	84560							
TYPBIN	20640	84560							
TYPDEC	20340	84560							
TYPNAM	18060	84560							
TYPNUM	20010	84560							
TYPOCS	19540	84560							
TYPOCT	19170	84560							
TYPTXT	18720	84560							
$ABORT	28380	56760	8457						
$BUFFE	69340	8457							
$CYCLE	75980	8457							
$EOP	68750	8457							
$GETFL	67380	8456							
$GETPA	67280	8456	8457						
$HEADE	64870	8456							
$INTSE	69810	8860							
$JUNK	69670	8456							
$MRESE	67180	8811							
$MSG	69000	8457							
$PARTS	83550	8860							
$SCOPE	68120	8457							
$SETFL	67440	8457							
$STAG	69880	8526							
$STAGF	69750								
$TRPDE	65120	8456							
$TSTN	68260	8475	8578	8647	8790	8860			
$VARIA	65250	8456							
$XZ	79010	8462	8474	8570	8577	8641	8646	8771	8789	8860
$$CMRE	84560								
$$CMTM	84560								
$$ESCA	16970	84560							
$$NEWT	16520	84560	8475	8578	8647	8790	8860		
$$SKIP	17300	84560							
.EQUAT	1910	56760	8456						
.HEADE	670	56750	8456						

```
CVDZB D MACY11 30A(1052)  21 AUG 84  08 31  PAGE 32-1
CVDZBD P11    21-AUG 84 08:28          CROSS REFERENCE TABLE    MACRO NAMES

 KI11     3330
 SETUP   12130   5675#
 SWRMI     108#
 $ACT1   50904   5677#   8456
 $APTB   51334   5677#   84564
 $APTH   53884   5677#   8456
 $APTY   55604   5677#   8457
 $ASTA   54334
 $CATC    917#   5675#
 $CMTA   10264   8456#
 $DB2D   47264
 $DB2O   48474
 $DIV    46304
 $EOP    21854   5675#   8457
 $ERRO   26644   5676#
 $ERRT   29194
 $MULT   45684
 $POWE   42444   5676#   8457
 $RAND   43184
 $RDDE   39184
 $RDOC   38284
 $READ   34274
 $SAZ    49884
 $SAVE   39924
 $SB2D   48094
 $SB2O   49084
 $SCOP   24194   5676#   8457
 $SIZE   43704
 $SUPR   49454
 $TRAP   40934   5676#
 $TYPB   33214
 $TYPD   32454
 $TYPE   30054   5675#   8457
 $TYPO   31504
 @40CA    9554
 .1170    5114


  ABS.  017352     000


ERRORS DETECTED:  0

CVDZBD,CVDZBD/CRF=CVDZBD.MLB,CVDZBD.P11
RUN TIME: 14 17 1 SECONDS
RUN-TIME RATIO: 229/33=6.7
CORE USED:  49K  (97 PAGES)
```