

DLV-11

DLV-11E OFFLINE TEST  
CVDVADO

COPYRIGHT (c) 1977-84  
AH-B151D-MC  
FICHE 01 OF 01

FEB 1985  
digital  
Made In USA

This page contains a grid of 120 microfiche frames, arranged in 10 rows and 12 columns. Each frame contains a small, high-contrast image of a document page, likely a technical drawing or test report. The images are very small and difficult to read, but they appear to be organized in a systematic manner. The right side of the page is mostly blank, with a small grid of 12x10 frames in the bottom right corner.

.REM 0

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33

IDENTIFICATION  
-----

PRODUCT CODE: AC-B1500-MC  
PRODUCT TITLE: CVDVADO DLV11-E OFFLINE TEST  
PRODUCT DATE: AUGUST 1984  
AUTHOR: ODES CHOATE  
MAINTAINER: DIAGNOSTIC ENGINEERING GROUP

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

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

COPYRIGHT (C) 1977,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBUS  
DEC DECUS DECTAPE

34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70TABLE OF CONTENTS  
-----

1.0	GENERAL PROGRAM INFORMATION.
1.1	PROGRAM PURPOSE (ABSTRACT).
1.2	SYSTEM REQUIREMENTS.
1.3	RELATED DOCUMENTS AND STANDARDS.
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES.
1.5	ASSUMPTIONS.
2.0	OPERATING INSTRUCTIONS.
2.1	LOADING AND STARTING PROCEDURES.
2.2	SPECIAL ENVIRONMENTS.
2.3	OPERATIONAL SWITCH SETTINGS
2.4	PROGRAM OPTIONS.
2.5	EXECUTION TIMES.
3.0	ERROR INFORMATION.
3.1	ERROR REPORTING PROCEDURE.
3.2	ERROR HALTS.
4.0	PERFORMANCE AND PROGRESS REPORTS.
4.1	PERFORMANCE REPORTS.
5.0	DEVICE INFORMATION TABLES.
6.0	SUMMARY OF TESTS AND SPECIAL SUBROUTINES

71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
1171.0 GENERAL PROGRAM INFORMATION.  
-----

## 1.1 PROGRAM PURPOSE (ABSTRACT).

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE DLV11-E SERIAL LINE INTERFACE. THE PROGRAM AS SET INITIALLY DEFAULTS TO ALL OPTIONS, EXCEPT PROGRAMMABLE BAUD RATE, ENABLED AND A WRAP CABLE CONNECTED. THE USER CAN SELECTIVELY ENABLE AND DISABLE TESTING OF THE OPTIONS BY ALTERING THE CONTENTS OF 'USER'. THE DIAGNOSTIC IS DESIGNED TO TEST AND DETECT FAULTS TO THE LOGIC LEVEL (NOT TO THE CHIP LEVEL). THIS TEST OPERATES ON UP TO SIXTEEN(16) IDENTICALLY CONFIGURED DLV11-E SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

175610 -FIRST SERIAL LINE ADDRESS OF 16 CONSECUTIVE SERIAL LINE DEVICES.

300 - VECTOR FOR FIRST OF 16 DEVICES.

THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 4K OF MEMORY AND A DLV11-E (LSI-BUS) MODULE. IT CAN RUN UNDER XXDP, APT, AND ACT MONITORS, AND ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER. A POWER FAILURE WILL CAUSE THE DIAGNOSTIC TO RESTART.

## 1.2 SYSTEM REQUIREMENTS.

## 1. HARDWARE REQUIREMENTS:

ANY PDP-11 FAMILY PROCESSOR  
4K MEMORY - MINIMUM  
M315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)  
MODEM CABLE - BC01V-X OR BC05C-X

## SOFTWARE REQUIREMENTS:

THIS DIAGNOSTIC IS DESIGNED TO RUN IN ANY OF THE FOLLOWING WAYS:  
STAND ALONE  
WITH APT MONITOR  
WITH ACT MONITOR  
WITH XXDP MONITOR (CHAINABLE)

118	1.3	RELATED DOCUMENTS AND STANDARDS.	
119		DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS	175-003-009-02
120		APT	MD-11-DZZMA
121		ACT	AUTOCAT-11-QZAUB
122		SYSMAC	MD-11-DZQAC
123			
124			
125			
126	1.4	DIAGNOSTIC HIERARCHY PREREQUISITES.	
127		NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT	
128		THE PROCESSOR, MEMORY, AND BUS ARE ASSUMED TO BE FULLY	
129		OPERATIONAL.	
130			
131			
132	1.5	ASSUMPTIONS.	
133			
134			
135		THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED	
136		LOCATION '#USMR' AND '#DEVM' TO THE PROPER VALUES.	
137		THE (H) JUMPER MUST BE REMOVED FROM ALL DLV11-E'S UNDER TEST.	
138			
139			
140	2.0	OPERATING INSTRUCTIONS.	
141		-----	
142			
143			
144	2.1	LOADING AND STARTING PROCEDURES.	
145			
146		USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED	
147		MEDIA.	
148			
149		THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING ADDRESS. 200 FOR	
150		START AND RESTART.	
151			
152		THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING	
153		SWITCH 8 IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL)	
154		IN THE LOWER BYTE. (NOTE: ALL TESTS PREVIOUS TO THE SELECTED	
155		ONE ARE EXECUTED WITHOUT ITERATIONS.)	
156			
157			
158	2.2	SPECIAL ENVIRONMENTS.	
159			
160		THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING	
161		UDER APT,ACT,XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE	
162		PROCEDURES MANUAL AND SYSMAC PACKAGE.	
163			

2.3 OPERATIONAL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G <+G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: ' SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
  - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
  - B) IF A CONTROL U <+U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.
  - C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT.)

DYNAMIC SWITCH REGISTER

-----

- BIT 15 - HALT ON ERROR  
 14 - LOOP ON TEST  
 13 - INHIBIT ERROR TYPEOUTS  
 12 - (UNUSED)  
 11 - INHIBIT ITERATIONS  
 10 - BELL ON ERROR  
 9 - LOOP ON ERROR  
 8 - LOOP ON TEST IN SWR<7:0>  
 7:0 - TEST NUMBER TO LOOP ON (USED WITH BIT 8)

164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217

218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266

## 2.4 PROGRAM OPTIONS.

THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE DLV11-E'S. IT  
 REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '#BASE') AND  
 ITS INTERRUPT VECTOR (STORED AT '#VECT1'); AND WILL BE ABLE  
 TO ADDRESS ANY DLV11-E STARTING AT THE SPECIFIED BASE ADDRESS  
 UP TO 16 CONSECUTIVE DEVICES.

EXAMPLES:      #BASE: 175610  
                   #VECT1: 300

THE PROGRAM WILL BE ABLE TO TEST ANY DLV11-E WITHIN THE  
 ADDRESS RANGE 175610 --> 176000

#BASE AND #VECT1 DEFAULT TO 175610 AND 300 RESPECTIVELY.  
 THE PROGRAM ASSOCIATES UNIT NUMBERS AS FOLLOWS: (NUMBERS IN  
 PARENTHESIS ARE OCTAL)

UNIT#0 -- BASE ADDRESS STORED AT '#BASE'  
           ASSOCIATED BASE VECTOR STORED AT '#VECT1'  
 UNIT#1 -- BASE ADDRESS + (10)  
           BASE VECTOR + (10)

⋮  
 UP TO

UNIT#15 -- BASE ADDRESS + (170)  
           BASE VECTOR + (170)

LOCATION '#DEVM' IS USED AS A BIT MAP TO INDICATE WHICH UNIT  
 NUMBERS ARE PRESENT AND WILL BE TESTED.

BIT 15	BIT 1	BIT 0
!UNIT!	!UNIT!	UNIT!
! 15 !	! 01 !	! 00 !

A BIT MAP CAN BE ENTERED AT '#DEVM' PRIOR TO STARTING THE  
 PROGRAM.

### EXAMPLE:

#BASE: 175610  
 #VECTOR: 300  
 #DEVM: 13

THE PROGRAM WILL TEST-

UNIT#0	175610	300
UNIT#1	175620	310
UNIT#3	175640	330

267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322

OPTIONS

-----  
LOCATION #USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION. THE DEFAULT VALUE OF #USWR IS AS FOLLOWS:

BIT POSITION	DEFINITION	DEFAULT VALUE
-----	-----	-----
0-3	#OF DATA BITS	10(8) = 8
4	PARITY ENABLED	0 = NO
5	EVEN ODD PARITY	0 = ODD
6	COMMON SPEED	1 = YES
7	PROGRAMMABLE BAUD RATE	0 = NO
8-11	BAUD RATE OFFSET (SEE FOLLOWING NOTE)	05(8) = 110 BAUD
12	BREAK GENERATION ENABLED	1 = YES
13	CABLE TERMINATED (H315)	1 = YES
14	(-FR) AND (-FD) JUMPERS IN	1 = YES
15	(NOT DEFINED)	

NOTE

THIS DIAGNOSTIC DOES NOT TEST THE PARITY LOGIC.

WHEN THE PROGRAMMABLE BAUD RATE OPTION IS ENABLED THE PROGRAMMABLE BAUD RATE TEST WILL EXIT WITH THE BAUD RATE SET TO THE SELECTED VALUE. TO CHANGE THE DEFAULT VALUE OF 110 BAUD REPLACE BITS <11:8> WITH THE OFFSET INDICATED IN THE TABLE AT THE END OF THE PBR TEST.

DLV11-E INDIVIDUAL TEST REQUIREMENTS

	TESTS NOT EXECUTED	IF BIT =
	-----	-----
APT TEST (BIT 1 OF #ENV)	T25, T37	1
PROGRAMMABLE BAUD RATE (BIT 7 OF #USWR)	T32	1
BREAK GENERATION CIRCUIT (BIT 12 OF #USWR)	T2, T40	0
CABLE TERMINATED (BIT 13 OF #USWR)	T15, T16, T17, T20, T21 T22, T23, PARTS OF T34, T36	1
(-FR) & (-FD) JUMPERS IN (BIT 14 OF #USWR)	T5, T6, T16, T17, T20, T22, T23	1

ALL OTHER TESTS WILL RUN REGARDLESS OF ANY BIT SETTINGS.



MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 10  
CVDVAD.P11      12-JUL-84 05:04

323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337

2.5      EXECUTION TIMES.

EXECUTION TIMES ARE FOR AN LSI-11 PROCESSOR WITH ALL OPTIONS  
ENABLED ON THE DLV11-E (EXCEPT FOR PROGRAMMABLE BAUD RATE), AT  
110 BAUD.

FIRST PASS-            90 SECONDS  
ADDITIONAL PASSES 95 SECONDS  
ADDITIONAL DEVICES 95 SECONDS

THE TEST TIME IS BAUD RATE DEPENDANT; HIGHER BAUD GIVES  
SHORTER PASS TIMES.

338 3.0 ERROR INFORMATION.  
 339 -----  
 340  
 341 3.1 ERROR REPORTING PROCEDURE.  
 342 -----  
 343  
 344 SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4-K OF MEMORY THE  
 345 ERROR TYPEOUT IS VERY BRIEF. THE FORMAT OF THE ERROR TYPEOUT  
 346 IS AS FOLLOWS:  
 347  
 348 TEST#\_\_\_\_\_,ERROR#\_\_\_\_\_,PC=\_\_\_\_\_,ADDRESS=\_\_\_\_\_,VECTOR=\_\_\_\_\_  
 349  
 350 WHERE ALL VALUES TYPED ARE OCTAL.  
 351 THE ADDRESS AND VECTOR REFER TO THE FAILING DLV11-E.  
 352 FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED.  
 353 BITS 15,13,10 AND 9 OF THE SWITCH REGISTER CONTROL THE  
 354 SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.  
 355  
 356  
 357 BIT 15 - CAUSES THE PROGRAM TO HALT IN THE ERROR  
 358 ROUTINE. CONTINUEING THE PROGRAM CAUSES IT TO  
 359 PROCEED.  
 360  
 361 BIT 13 - DISABLES THE PRINTING OF THE ERROR MESSAGE.  
 362  
 363 BIT 10 - CAUSES THE BELL TO RING ON ERROR.  
 364  
 365 BIT 9 - CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING  
 366 OF TEST TO ERROR.  
 367  
 368 THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.  
 369  
 370 3.2 ERROR HALTS.  
 371 -----  
 372  
 373 THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE, AND  
 374 IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER IS A ONE  
 375 WHEN AN ERROR OCCURS.  
 376  
 377 4.0 PERFORMANCE AND PROGRESS REPORTS.  
 378 -----  
 379  
 380 4.1 PERFORMANCE REPORTS.  
 381  
 382 AS EACH DEVICE COMPLETES ONE PASS OF THE DIAGNOSTIC THE  
 383 FOLLOWING WILL BE TYPED:  
 384 CSR:\_\_\_\_\_,VECTOR:\_\_\_\_\_,ERRORS:\_\_\_\_\_  
 385

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 12  
CVDVAD.P11      12-JUL-84 05:04

386                    WHERE. 'CSR:-----' IS THE DEVICE CSR UNDER TEST  
387                    'VECTOR:--' IS THE ASSOCIATED VECTOR  
388                    AND 'ERRORS:--' IS THE TOTAL NUMBER OF ERRORS ON THIS DEVICE  
389                    ON THIS PASS.

390  
391  
392                    NOTE  
393                    THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.  
394  
395                    AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS  
396                    TYPED:            "ENDPASS@-----"  
397

5.0 DEVICE INFORMATION TABLES.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCSR	DATA	RING	CLR	CAR	RCVR	REC			RCVR	RCVR	DATA		SEC	REQ	DTR	
	INT		SEND	DET	ACT	REC			DONE	IE	IE		XMIT	SEND		
RBUF	ERRO	OR	FR	P									RECEIVED DATA BUFFER			
	R	ERR	ERR	ERR												
TCSR	PROGRAMMABLE BAUD			PBR					XMIT	XMIT				MAIN		BREA
		RATE	SELECT	ENAB					RDY	IE				T		K
TBUF													TRANSMITTER DATA BUFFER			

NOTE

BLANK BOXES INDICATE UNUSED AND RESERVED BIT POSITIONS. SEE THE LISTING FOR AN EXPLANATION OF THE BITS.

398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425

426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.  
-----

TEST 1 ADDRESSABILITY  
-----

THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN THE  
HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E  
RESPONDS TO THAT ADDRESS SPACE.

THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR0 SET, CLEAR, RESET  
-----

TEST 3 MAINT - TCSR2 SET, CLEAR, RESET  
-----

TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET  
-----

TEST 5 DTR - RCSR1 SET, CLEAR  
-----

NOTE

RESET DOES NOT CLEAR THIS BIT. WE CANNOT TEST  
FOR AN INITIAL CONDITION AS THIS BIT IS  
UNDEFINED UPON POWER UP AND INIT DOESN'T  
AFFECT IT.

TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET  
-----

THIS TEST ASSUMES THAT JUMPER FR IS IN.

472	TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET
473	-----
474	
475	
476	TEST 10 DATAIE - RCSR5 SET, CLEAR, RESET
477	-----
478	
479	
480	TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET
481	-----
482	

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 16  
 CVDVAD.P11 12-JUL-84 05:04

483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530

THE FOLLOWING 4 TESTS VERIFY THAT RESET (INIT) INITIALIZES  
 READ ONLY BITS.

TEST 12 RCVRDONE - RCSR 7 - IS CLEARED BY INIT  
 .... ..

TEST 13 XMITRDY - TCSR 7 - IS SET BY INIT  
 .... ..

TEST 14 DATAINT - RCSR 15 - IS CLEARED BY INIT.  
 .... ..

TEST 15 RCVRACT - RCSR 11 - 15 CLEARED BY INIT  
 .... ..

THE FOLLOWING 4 TESTS VERIFY THAT THE EIA SIGNALS CAN BE  
 TRANSMITTED AND RECEIVED THROUGH THE CABLE.

TEST 16 CARDET SETS AND CLEARS AS DTR SETS AND CLEARS  
 .... ..

TEST 17 CLRSEND SETS AND CLEARS AS DTR SETS AND CLEARS  
 .... ..

TEST 20 RING SETS AND CLEARS AS REQSEND SETS AND CLEARS  
 .... ..

TEST 21 SECREC SETS AND CLEARS AS SECXMIT SETS AND CLEARS  
 .... ..

TEST 22 DATAINT (RCSR-15) SETS WHEN DTR CHANGES STATE AND THAT  
 .... ..  
 DATAINT IS CLEARED AFTER READING RCSR

NOTE

DTR IS TIED TO BOTH CARDET AND CLRSEND BY THE  
 M315.

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 17  
CVDVAD.P11      12-JUL-84 05:04

531  
532  
533  
534  
535  
536  
537  
538  
539  
540

TEST 23 DATAINT SETS WHEN RING SETS AND THAT DATAINT  
---- ..  
DOES NOT SET WHEN RING CLEARS

TEST 24 DATAINT SETS WHEN SECREC CHANGES STATE  
---- ..



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 18  
CVDVAD.P11 12-JUL-84 05:04

541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592

TEST 25 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED

-----

WITH A CHARACTER AND THAT IT SETS WITHIN A  
REASONABLE AMOUNT OF TIME.

TEST 26 OUTPUTTING A CHR FROM TBUF (WITH MAINT SET)

-----

RESULTS IN RCVRDONE SETTING WITHIN A  
REASONABLE AMOUNT OF TIME AND THAT RESET  
CLEARS THE BIT.

TEST 27 RCVRDONE IS CLEARED BY READING RBUF

-----

TEST 30 RCVRACT - RCSR 11 - SETS WHEN A START BIT IS

-----

RECEIVED AND CLEARS WHEN RCVRDONE - RCSR 7 -  
SETS

TEST 31 OVERRUN BIT - RBUF 14

-----

TEST 32 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS

-----

AVAILABLE A COMPARISON WILL BE MADE TO SEE IF  
NEW TIME IS LESS THAN PREVIOUS.

TEST 33 TRANSMITTER INTERRUPT LOGIC TEST

-----

LOGICALLY THIS IS 4 SEPARATE TESTS  
A) DOES TRANSMITTER INTERRUPT LOGIC WORK  
B) AT PRIORITY OF 0  
C) AND ONLY ONCE  
D) BUT NOT WITH INTERRUPT ENABLE CLEAR

TEST 34 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL

-----

OF THE RECEIVER SIDE OF THE INTERRUPT LOGIC, BOTH  
DATASET AND CHARACTER MODES.

TEST 35 TEST ACTUAL DATA TRANSFERED NON-INTERRUPT

-----

MAINTENANCE BIT SET

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 19  
 CVDVAD.P11 12-JUL-84 05:04

593 TEST 36 TEST DATA THROUGH CABLE  
 594 ---- --  
 595  
 596  
 597 TEST 37 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE  
 598 ---- --  
 599 MODE.  
 600  
 601  
 602 TEST 40 TEST BREAK GENERATION LOGIC TRANSMIT KNOWN CHAR  
 603 ---- --  
 604 WITH BREAK SET AND COMPARE RECEIVED WITH 0.  
 605  
 606  
 607 TEST 41 NOT A TEST - SEND BACK TO LOOP  
 608 ---- --  
 609

## NOTE

FOR ALL OF THE FOLLOWING ROUTINES THE USE  
 OF (R5) IS PART OF THE LINKAGE MECHANISM  
 BETWEEN THE CALLER AND THE CALLED.

## ROUTINE:TIMER

-----

THIS ROUTINE IS USED TO TEST THE STATUS OF  
 ANY BIT IN ANY REGISTER.

## INPUTS:

HOWLONG THE MAXIMUM AMOUNT OF TIME TO  
 SPEND IN THIS ROUTINE.  
 WHICHBIT A MASK WITH THE BIT(S) SET THAT  
 ARE TO BE CHECKED  
 REG A POINTER TO THE REGISTER TO BE  
 CHECKED  
 SFTCLR THE DESIRED RESULTS -- EITHER SET  
 OR CLEAR

## OUTPUT:

THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT  
 IS TESTED BY THE IF.ERROR STATEMENT.

631  
 632  
 633  
 634  
 635

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 20  
CVDVAD.P11 12-JUL-84 05:04

636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677

## ROUTINE:DATLNG

-----  
THIS ROUTINE SETS UP A MASK FOR DATA, WITH -  
INPUT: NOTHING IS PASSED TO THIS ROUTINE BUT GLOBAL  
INFORMATION IS ASSUMED TO EXIST:  
\$USWR-- THE WORD FOR SOFTWARE PARAMETERS  
DATA-- A MASK FOR THE LOCATION OF THE OCTAL  
NUMBER OF DATA BITS  
OUTPUT-----  
MASK-- A MASK OF BINARY ZEROS RIGHT-JUSTIFIED  
THE NUMBER OF WHICH IS DEFINED IN \$USWR WORD.

## ROUTINE:WAIT

-----  
THIS ROUTINE IS USED TO DELAY EXECUTION OF THE  
MAIN PROGRAM FOR A SPECIFIED AMOUNT OF TIME.  
THIS IS ACCOMPLISHED BY INCREMENTING A  
REGISTER UP TO A LIMIT. THE INNER LOOP IS SET  
TO APPROXIMATE 1 MILLI SEC.

## SERVICE ROUTINE: INTSRV

-----  
THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT  
  
'INTFLAG' EACH TIME IT IS CALLED. IT ASSUMES  
THAT THE MAIN CALLING ROUTINE WILL KNOW WHAT  
TO LOOK FOR.

## ROUTINE:CYCLE

-----  
THIS ROUTINE CAUSES ADRS TO POINT TO THE  
ADDRESS OF DLV11-E UNDER TEST. ADRS +2 TO  
POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.  
IT KEEPS TRACK OF THE CURRENT DEVICE AND BIT  
MASKS.

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 21  
 CVDVAD.P11 12-JUL-84 05:04

```

678      @
679      .TITLE MAINDEC-ZZ-CVDVA-D
680      ;*COPYRIGHT (C) 1977
681      ;*DIGITAL EQUIPMENT CORP.
682      ;*MAYNARD, MASS. 01754
683      ;*
684      ;*PROGRAM BY ODES CHOATE
685      ;*
686      ;*THIS PROGRAM WAS ASSEMBLED USING THE POP-11 MAINDEC SYSMAC
687      ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
688      ;*
689      .SBTTL OPERATIONAL SWITCH SETTINGS
690      ;*
691      ;*      SWITCH                      USE
692      ;*      -----                      -----
693      ;*      15                      HALT ON ERROR
694      ;*      14                      LOOP ON TEST
695      ;*      13                      INHIBIT ERROR TYPEOUTS
696      ;*      11                      INHIBIT ITERATIONS
697      ;*      10                      BELL ON ERRGR
698      ;*      9                       LOOP ON ERROR
699      ;*      8                       LOOP ON TEST IN SWR<7:0>
700
701      .SBTTL BASIC DEFINITIONS
702
703      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
704      001100 STACK= 1100
705      .EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
706      .EQUIV IOT,SCOPE     ;;BASIC DEFINITION OF SCOPE CALL
707
708      ;*MISCELLANEOUS DEFINITIONS
709      000011 HT= 11        ;;CODE FOR HORIZONTAL TAB
710      000012 LF= 12        ;;CODE FOR LINE FEED
711      000015 CR= 15        ;;CODE FOR CARRIAGE RETURN
712      000200 CRLF= 200     ;;CODE FOR CARRIAGE RETURN-LINE FEED
713      177776 PS= 177776   ;;PROCESSOR STATUS WORD
714      .EQUIV PS,PSW
715      177774 STKLMT= 177774 ;;STACK LIMIT REGISTER
716      177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
717      177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
718      177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
719
720      ;*GENERAL PURPOSE REGISTER DEFINITIONS
721      000000 R0= #0        ;;GENERAL REGISTER
722      000001 R1= #1        ;;GENERAL REGISTER
723      000002 R2= #2        ;;GENERAL REGISTER
724      000003 R3= #3        ;;GENERAL REGISTER
725      000004 R4= #4        ;;GENERAL REGISTER
726      000005 R5= #5        ;;GENERAL REGISTER
727      000006 R6= #6        ;;GENERAL REGISTER
728      000007 R7= #7        ;;GENERAL REGISTER
729      000006 SP= #6        ;;STACK POINTER
730      000007 PC= #7        ;;PROGRAM COUNTER
731
732      ;*PRIORITY LEVEL DEFINITIONS
733      000000 PRO= 0        ;;PRIORITY LEVEL 0

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 22  
 CVDVAD.P11 12-JUL-84 05:04 BASIC DEFINITIONS

734	000040	PR1=	40	::	PRIORITY LEVEL 1
735	000100	PR2=	100	::	PRIORITY LEVEL 2
736	000140	PR3=	140	::	PRIORITY LEVEL 3
737	000200	PR4=	200	::	PRIORITY LEVEL 4
738	000240	PR5=	240	::	PRIORITY LEVEL 5
739	000300	PR6=	300	::	PRIORITY LEVEL 6
740	000340	PR7=	340	::	PRIORITY LEVEL 7

741  
 742 ;\* "SWITCH REGISTER" SWITCH DEFINITIONS

743	100000	SW15=	100000
744	040000	SW14=	40000
745	020000	SW13=	20000
746	010000	SW12=	10000
747	004000	SW11=	4000
748	002000	SW10=	2000
749	001000	SW09=	1000
750	000400	SW08=	400
751	000200	SW07=	200
752	000100	SW06=	100
753	000040	SW05=	40
754	000020	SW04=	20
755	000010	SW03=	10
756	000004	SW02=	4
757	000002	SW01=	2
758	000001	SW00=	1
759		.EQUIV	SW09,SW9
760		.EQUIV	SW08,SW8
761		.EQUIV	SW07,SW7
762		.EQUIV	SW06,SW6
763		.EQUIV	SW05,SW5
764		.EQUIV	SW04,SW4
765		.EQUIV	SW03,SW3
766		.EQUIV	SW02,SW2
767		.EQUIV	SW01,SW1
768		.EQUIV	SW00,SW0

769  
 770 ;\* DATA BIT DEFINITIONS (BIT00 TO BIT15)

771	100000	BIT15=	100000
772	040000	BIT14=	40000
773	020000	BIT13=	20000
774	010000	BIT12=	10000
775	004000	BIT11=	4000
776	002000	BIT10=	2000
777	001000	BIT09=	1000
778	000400	BIT08=	400
779	000200	BIT07=	200
780	000100	BIT06=	100
781	000040	BIT05=	40
782	000020	BIT04=	20
783	000010	BIT03=	10
784	000004	BIT02=	4
785	000002	BIT01=	2
786	000001	BIT00=	1
787		.EQUIV	BIT09,BIT9
788		.EQUIV	BIT08,BIT8
789		.EQUIV	BIT07,BIT7

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 23  
 CVDVAD.P11 12-JUL-84 05:04 BASIC DEFINITIONS

```

790      .EQUIV BIT06,BIT6
791      .EQUIV BIT05,BIT5
792      .EQUIV BIT04,BIT4
793      .EQUIV BIT03,BIT3
794      .EQUIV BIT02,BIT2
795      .EQUIV BIT01,BIT1
796      .EQUIV BIT00,BIT0
797
798      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
799      000004  ERRVEC= 4          ;; TIME OUT AND OTHER ERRORS
800      000010  RESVEC= 10       ;; RESERVED AND ILLEGAL INSTRUCTIONS
801      000014  TBITVEC=14      ;; "T" BIT
802      000014  TRTVEC= 14      ;; TRACE TRAP
803      000014  BPTVEC= 14      ;; BREAKPOINT TRAP (BPT)
804      000020  IOTVEC= 20      ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
805      000024  PMRVEC= 24      ;; POWER FAIL
806      000030  EMTVEC= 30      ;; EMULATOR TRAP (EMT) **ERROR**
807      000034  TRAPVEC=34     ;; "TRAP" TRAP
808      000060  TKVEC= 60       ;; TTY KEYBOARD VECTOR
809      000064  TPVEC= 64       ;; TTY PRINTER VECTOR
810      000240  PIRQVEC=240     ;; PROGRAM INTERRUPT REQUEST VECTOR
811
812      ILLMEM= 4
813      ADRS= P1
814      GOOD= R2
815      BAD= R3
816      REGISTER=R1
817      BIT= R2
818      FUNCT= R3
819      LEAD= R2
820      FOLLOW= R4
821      DLADDR= 175610
822
823      ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
824      177777  SET= -1
825      000000  CLR= 0
826
827      ;*****
828      ; RCSR REGISTER BIT NAMES
829      ;*****
830      100000  DATAINT= BIT15   ; DATASET INTERRUPT
831      040000  RING= BIT14      ; RINGING SIGNAL INDICATOR
832      020000  CLRSEND= BIT13   ; CLEAR TO SEND FROM DATASET
833      010000  CARDET= BIT12    ; CARRIER DETECT
834      004000  RCVRACT= BIT11   ; RECEIVER ACTIVE INDICATOR
835      002000  SECREC= BIT10    ; SECONDARY RECEIVE
836      ; UNUSED BIT09
837      ; UNUSED BIT08
838      000200  RCVRDONE= BIT07   ; RECEIVER DONE
839      000100  RCVRIE= BIT06    ; RECEIVER INTERRUPT ENABLE
840      000040  DATAIE= BIT05   ; DATASET INTERRUPT ENABLE
841      ; UNUSED BIT04
842      000010  SECXMIT= BIT03    ; SECONDARY TRANSMIT DATA
843      000004  REQSEND= BIT02    ; REQUEST TO SEND
844      000002  DTR= BIT01      ; DATA TERMINAL READY
845      ; UNUSED BIT00

```

```

846
847 ;:*****
848 ; RBUF REGISTER BIT NAMES
849 ;:*****
850 100000 ERROR= BIT15 ; ERROR INDICATOR
851 040000 ORERR= BIT14 ; OVERRUN ERROR
852 020000 FRERR= BIT13 ; FRAMING ERROR
853 010000 PERR= BIT12 ; PARITY ERROR
854 ; UNUSED BIT11
855 ; UNUSED BIT10
856 ; UNUSED BIT09
857 ; UNUSED BIT08
858 000200 RDATA7= BIT07 ; \
859 000100 RDATA6= BIT06 ; |
860 000040 RDATA5= BIT05 ; |
861 000020 RDATA4= BIT04 ; |
862 000010 RDATA3= BIT03 ; |
863 000004 RDATA2= BIT02 ; |
864 000002 RDATA1= BIT01 ; |
865 000001 RDATA0= BIT00 ; /
866
867 ;:*****
868 ; TCSR REGISTER BIT NAMES
869 ;:*****
870 100000 PBAUD3= BIT15 ; \
871 040000 PBAUD2= BIT14 ; | PROGRAMMABLE BAUD
872 020000 PBAUD1= BIT13 ; | RATE BITS
873 010000 PBAUD0= BIT12 ; /
874 004000 PBAUDSET= BIT11 ; ENABLE SETTING OF
875 ; PROGRAMMABLE BAUDE RATE
876 ; UNUSED BIT10
877 ; UNUSED BIT09
878 ; UNUSED BIT08
879 000200 XMITRDY= BIT07 ; TRANSMITTER READY
880 000100 XMITIE= BIT06 ; TRANSMITTER INTERRUPT ENABLE
881 ; UNUSED BIT05
882 ; UNUSED BIT04
883 ; UNUSED BIT03
884 000004 MAINT= BIT02 ; MAINTENANCE SET BIT
885 ; UNUSED BIT01
886 000001 BREAK= BIT00 ; SEND BREAK (CONTINUOUS SPACE)
887
888
889 ;:*****
890 ; TBUF REGISTER BIT NAMES
891 ;:*****
892 ; UNUSED BIT15

```

```

893 ; UNUSED BIT14
894 ; UNUSED BIT13
895 ; UNUSED BIT12
896 ; UNUSED BIT11
897 ; UNUSED BIT10
898 ; UNUSED BIT09
899 ; UNUSED BIT08
900 000200 TDATA7= BIT07 ; \
901 000100 TDATA6= BIT06 ; !
902 000040 TDATA5= BIT05 ; !
903 000020 TDATA4= BIT04 ; \ TRANSMITTER DATA BUFFER
904 000010 TDATA3= BIT03 ; /
905 000004 TDATA2= BIT02 ; !
906 000002 TDATA1= BIT01 ; !
907 000001 TDATA0= BIT00 ; /

```

```

;*****
; FLAG BITS TO BE USE OR CLEARED IN $USWR.

```

```

913 000017 DATA = 17
914 000020 PARITY = 20
915 000040 EVENODD = 40
916 000100 COMSPD = 100
917 000200 PBR = 200
918
919 ; BAUDE MUST BE ON THE UPPER
920 ; BYTE BOUNDRY OF $USWR.--4 BITS
921 007400 BAUD = 7400
922 010000 BRK = 10000
923 020000 CABLE = 20000
924 040000 FRFD = 40000

```

```

;*****
.SBTTL TRAP CATCHER

```

```

928 000000 .=0
929 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
930 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
931 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
932 000174 .=174
933 000174 000000 DISPRG: .WORD 0 ;:SOFTWARE DISPLAY REGISTER
934 000176 000000 SMREG: .WORD 0 ;:SOFTWARE SWITCH REGISTER
935
936 000200 000137 001336 .SBTTL STARTING ADDRESS(ES)
937 JMP @START ;:JUMP TO STARTING ADDRESS OF PROGRAM
938 .SBTTL ACT11 HOOKS

```

```

;*****
;HOOKS REQUIRED BY ACT11

```

```

941 000204 $SVPC=. ;SAVE PC
942 000046 .=46
943 000046 012452 $ENDAD ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
944 000052 .=52
945 000052 000000 .WORD 0 ;:2)SET LOC.52 TO ZERO
946 000204 .=$SVPC ;: RESTORE PC
947 001000 .=1000
948 .SBTTL APT PARAMETER BLOCK

```



MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 26  
 CVDVAD.P11      12-JUL-84 05:04      APT PARAMETER BLOCK

```

949
950      ;:*****
951      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
952      ;:*****
953      001000      . $X=      ;:SAVE CURRENT LOCATION
954      000024      =24      ;:SET POWER FAIL TO POINT TO START OF PROGRAM
955      000024      000200      200      ;:FOR APT START UP
956      000044      =44      ;:POINT TO APT INDIRECT ADDRESS PNTR.
957      000044      001000      $APTHDR ;:POINT TO APT HEADER BLOCK
958      001000      =.$X      ;:RESET LOCATION COUNTER
959      ;:*****
960      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
961      ;INTERFACE SPEC.
962
963      001000      $APTHD:
964      001000      000000      $HIBTS: .WORD 0      ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
965      001002      001174      $MBADR: .WORD $MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)
966      001004      000005      $TSTM: .WORD 5      ;:RUN TIM OF LONGEST TEST
967      001006      000055      $PASTM: .WORD 45. ;:RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
968      001010      000036      $UNITM: .WORD 30. ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
969      001012      000030      .WORD $ETEND-$MAIL/2 ;:LENGTH MAILBOX-ETABLE(WORDS)

```

```

970      .SBTTL  COMMON TAGS
971
972      ;;*****
973      ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
974      ;*USED IN THE PROGRAM.
975
976      001100      .=1100
977      001100      $CMTAG:                ;;START OF COMMON TAGS
978      001100      000000      .WORD      0
979      001102      000      $TSTNM: .BYTE  0      ;;CONTAINS THE TEST NUMBER
980      001103      000      $ERFLG: .BYTE  0      ;;CONTAINS ERROR FLAG
981      001104      000000      $ICNT:  .WORD  0      ;;CONTAINS SUBTEST ITERATION COUNT
982      001106      000000      $LPADR: .WORD  0      ;;CONTAINS SCOPE LOOP ADDRESS
983      001110      000000      $LPERR: .WORD  0      ;;CONTAINS SCOPE RETURN FOR ERRORS
984      001112      000000      $ERTTL: .WORD  0      ;;CONTAINS TOTAL ERRORS DETECTED
985      001114      000      $ITEMB: .BYTE  0      ;;CONTAINS ITEM CONTROL BYTE
986      001115      001      $ERMAX: .BYTE  1      ;;CONTAINS MAX. ERRORS PER TEST
987      001116      000000      $ERRPC: .WORD  0      ;;CONTAINS PC OF LAST ERROR INSTRUCTION
988      001120      000000      $GOADR: .WORD  0      ;;CONTAINS ADDRESS OF 'GOOD' DATA
989      001122      000000      $BDADR: .WORD  0      ;;CONTAINS ADDRESS OF 'BAD' DATA
990      001124      000000      $GDDAT: .WORD  0      ;;CONTAINS 'GOOD' DATA
991      001126      000000      $BDDAT: .WORD  0      ;;CONTAINS 'BAD' DATA
992      001130      000000      .WORD  0      ;;RESERVED--NOT TO BE USED
993      001132      000000      .WORD  0
994      001134      000      $AUTOB: .BYTE  0      ;;AUTOMATIC MODE INDICATOR
995      001135      000      $INTAG: .BYTE  0      ;;INTERRUPT MODE INDICATOR
996      001136      000000      .WORD  0
997      001140      177570      $SMR:   .WORD  DSWR      ;;ADDRESS OF SWITCH REGISTER
998      001142      177570      $DISPLAY: .WORD  DDISP      ;;ADDRESS OF DISPLAY REGISTER
999      001144      177560      $TKS:   177560      ;;TTY KBD STATUS
1000     001146      177562      $TKB:   177562      ;;TTY KBD BUFFER
1001     001150      177564      $TPS:   177564      ;;TTY PRINTER STATUS REG. ADDRESS
1002     001152      177566      $TPB:   177566      ;;TTY PRINTER BUFFER REG. ADDRESS
1003     001154      000      $NULL: .BYTE  0      ;;CONTAINS NULL CHARACTER FOR FILLS
1004     001155      002      $FILLS: .BYTE  2      ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
1005     001156      012      $FILLC: .BYTE  12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
1006     001157      000      $TPFLG: .BYTE  0      ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
1007     001160      000000      $TIMES: 0      ;;MAX. NUMBER OF ITERATIONS
1008     001162      000000      $ESCAPE:0      ;;ESCAPE ON ERROR ADDRESS
1009     001164      177607      000377 $BELL:  .ASCIZ <207><377><377> ;;CODE FOR BELL
1010     001170      077      $QUES:  .ASCII  /?/      ;;QUESTION MARK
1011     001171      015      $CRLF:  .ASCII  <15>      ;;CARRIAGE RETURN
1012     001172      000012      $LF:   .ASCIZ  <12>      ;;LINE FEED
1013
1014      .SBTTL  APT MAILBOX-ETABLE
1015
1016      ;;*****
1017      .EVEN
1018     001174      $MAIL:                ;;APT MAILBOX
1019     001174      000000      $MSGTY: .WORD  MSGTY      ;;MESSAGE TYPE CODE
1020     001176      000000      $FATAL: .WORD  AFATAL     ;;FATAL ERROR NUMBER
1021     001200      000000      $TESTN: .WORD  ATESTN    ;;TEST NUMBER
1022     001202      000000      $PASS:  .WORD  APASS      ;;PASS COUNT
1023     001204      000000      $DEVCT: .WORD  ADEVCT    ;;DEVICE COUNT
1024     001206      000000      $UNIT:  .WORD  AUNIT      ;;I/O UNIT NUMBER
1025     001210      000000      $MSGAD: .WORD  MSGAD      ;;MESSAGE ADDRESS

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 28  
 CVDVAD.P11 12-JUL-84 05:04 APT MAILBOX-ETABLE

1026	001212	000000	\$MSGLG: .WORD	AMSGLG	::MESSAGE LENGTH
1027	001214		\$ETABLE:		::APT ENVIRONMENT TABLE
1028	001214	000	\$ENV: .BYTE	AENV	::ENVIRONMENT BYTE
1029	001215	000	\$ENVM: .BYTE	AENVM	::ENVIRONMENT MODE BITS
1030	001216	000000	\$SWREG: .WORD	ASWREG	::APT SWITCH REGISTER
1031	001220	071110	\$USWR: .WORD	AUSWR	::USER SWITCHES
1032	001222	000000	\$CPUOP: .WORD	ACPUOP	::CPU TYPE,OPTIONS
1033			;*:		BITS 15-11=CPU TYPE
1034			;*:		11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1035			;*:		11/70=06,PDQ=07,Q=10
1036			;*:		BIT 10=REAL TIME CLOCK
1037			;*:		BIT 9=FLOATING POINT PROCESSOR
1038			;*:		BIT 8=MEMORY MANAGEMENT
1039	001224	000	\$MAMS1: .BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
1040	001225	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE,BLK#1
1041			;*:		MEM.TYPE BYTE -- (HIGH BYTE)
1042			;*:		900 NSEC CORE=001
1043			;*:		300 NSEC BIPOLAR=002
1044			;*:		500 NSEC MOS=003
1045	001226	000000	\$MADR1: .WORD	AMADR1	::HIGH ADDRESS,BLK#1
1046			;*:		MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1047	001230	000	\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
1048	001231	000	\$MTYP2: .BYTE	AMTYP2	::MEM.TYPE,BLK#2
1049	001232	000000	\$MADR2: .WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
1050	001234	000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
1051	001235	000	\$MTYP3: .BYTE	AMTYP3	::MEM.TYPE,BLK#3
1052	001236	000000	\$MADR3: .WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
1053	001240	000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
1054	001241	000	\$MTYP4: .BYTE	AMTYP4	::MEM.TYPE,BLK#4
1055	001242	000000	\$MADR4: .WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
1056	001244	000300	\$VECT1: .WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
1057	001246	000000	\$VECT2: .WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
1058	001250	175610	\$BASE: .WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
1059	001252	000001	\$DEVM: .WORD	ADEVM	::DEVICE MAP
1060	001254		\$ETEND:		
1061			.MEXIT		

```

1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076 001254
1077
1078 001254 175610
1079 001256 000300
1080 001260 175610
1081 001262 175612
1082 001264 175614
1083 001266 175615
1084 001270 175616
1085 001272 000000
1086 001274 000020
1087 001334 000000
1088 001336
1089
1090
1091 001336 012706 001100
1092 001342 005026
1093 001344 022706 001140
1094 001350 001374
1095 001352 012706 001100
1096
1097 001356 012737 014374 000020
1098 001364 012737 000340 000022
1099 001372 012737 014174 000030
1100 001400 012737 000340 000032
1101 001406 012737 015326 000034
1102 001414 012737 000340 000036
1103 001422 012737 012506 000024
1104 001430 012737 000340 000026
1105 001436 016767 010756 010746
1106 001444 005067 177510
1107 001450 005067 177506
1108 001454 112767 000001 177433
1109 001462 012767 001462 177416
1110 001470 012767 001470 177412
1111
1112
1113 001476 013746 000004
1114 001502 012737 001536 000004
1115 001510 012767 177570 177422
1116 001516 012767 177570 177416
1117 001524 022777 177777 177406

```

.SBTTL ERROR POINTER TABLE

```

; * 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:

```

```

; * EM ; ; POINTS TO THE ERROR MESSAGE
; * DM ; ; POINTS TO THE DATA HEADER
; * DT ; ; POINTS TO THE DATA
; * DF ; ; POINTS TO THE DATA FORMAT

```

#ERRTB:

```

; ; GLOBAL DATA
; ; DLADD: DLADDR
; ; DLVEC: 300
; ; RCSR: DLADDR * 0
; ; RBUF: DLADDR * 2
; ; TCSR: DLADDR * 4
; ; TCSRHI: DLADDR * 5
; ; TBUF: DLADDR * 6
; ; I: 0
; ; .BLKW 20 ; FOR R5 STACK
; ; RSSTACK: .WORD 0

```

START:

```

.SBTTL INITIALIZE THE COMMON TAGS
; ; CLEAR THE COMMON TAGS (#CHTAG) AREA
MOV #CHTAG,R6 ; ; FIRST LOCATION TO BE CLEARED
CLR (R6) ; ; CLEAR MEMORY LOCATION
CMP #SWR,R6 ; ; DONE?
BNE -6 ; ; LOOP BACK IF NO
MOV #STACK,SP ; ; SETUP THE STACK POINTER
; ; INITIALIZE A FEW VECTORS
MOV #SCOPE,#IOTVEC ; ; IOT VECTOR FOR SCOPE ROUTINE
MOV #340,#IOTVEC+2 ; ; LEVEL 7
MOV #ERROR,#EMTVEC ; ; EMT VECTOR FOR ERROR ROUTINE
MOV #340,#EMTVEC+2 ; ; LEVEL 7
MOV #TRAP,#TRAPVEC ; ; TRAP VECTOR FOR TRAP CALLS
MOV #340,#TRAPVEC+2 ; ; LEVEL 7
MOV #PWRDN,#PWRVEC ; ; POWER FAILURE VECTOR
MOV #340,#PWRVEC+2 ; ; LEVEL 7
MOV #ENDCT,#EOPCT ; ; SETUP END-OF-PROGRAM COUNTER
CLR #TIMES ; ; INITIALIZE NUMBER OF ITERATIONS
CLR #ESCAPE ; ; CLEAR THE ESCAPE ON ERROR ADDRESS
MOVB #1,#ERMAX ; ; ALLOW ONE ERROR PER TEST
MOV #.,#LPADR ; ; INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV #.,#LPERR ; ; SETUP THE ERROR LOOP ADDRESS
; ; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
; ; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV #ERRVEC,-(SP) ; ; SAVE ERROR VECTOR
MOV #64,#ERRVEC ; ; SET UP ERROR VECTOR
MOV #DSWR,SWR ; ; SETUP FOR A HARDWARE SWICH REGISTER
MOV #DDISP,DISPLAY ; ; AND A HARDWARE DISPLAY REGISTER
CMP #-1,BSWR ; ; TRY TO REFERENCE HARDWARE SWR

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 30  
 CVDVAD.P11 12-JUL-84 05:04 INITIALIZE THE COMMON TAGS

```

1118 001532 001012          BNE      66$      ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1119                                ;;AND THE HARDWARE SWR IS NOT = -1
1120 001534 000403          BR       65$      ;;BRANCH IF NO TIMEOUT
1121 001536 012716 001544    64$:  MOV     #65$,(SP) ;;SET UP FOR TRAP RETURN
1122 001542 000002          RTI
1123 001544 012767 000176 177366 65$:  MOV     @SWREG,SWR ;;POINT TO SOFTWARE SWR
1124 001552 012767 000174 177362    MOV     @DISPREG,DISPLAY
1125 001560 012637 000004    66$:  MOV     (SP)+,@ERRVEC ;;RESTORE ERROR VECTOR
1126
1127 001564 005067 177412          CLR     @PASS      ;;CLEAR PASS COUNT
1128 001570 132767 000200 177417    BITB   @APTSIZE,@ENVH ;;TEST USER SIZE UNDER APT
1129 001576 001403          BEQ     67$      ;;YES,USE NON-APT SWITCH
1130 001600 012767 001216 177332    MOV     @SWREG,SWR ;;NO,USE APT SWITCH REGISTER
1131 001606
1132                                67$:
1133                                .SBTTL  TYPE PROGRAM NAME
1134                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1134 001606 005227 177777          INC     @-1      ;;FIRST TIME?
1135 001612 001051          BNE     68$      ;;BRANCH IF NO
1136 001614 022737 012452 000042    CMP     @ENDAD,@42 ;;ACT-11?
1137 001622 001445          BEQ     68$      ;;BRANCH IF YES
1138 001624 104401 001672          TYPE   ,69$    ;;TYPE ASCIZ STRING
1139                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1140 001630 005737 000042          TST     @42      ;;ARE WE RUNNING UNDER XXDP/ACT?
1141 001634 001012          BNE     70$      ;;BRANCH IF YES
1142 001636 126727 177352 000001    CMPB   @ENV,@1   ;;ARE WE RUNNING UNDER APT?
1143 001644 001406          BEQ     70$      ;;BRANCH IF YES
1144 001646 026727 177266 000176    CMP     SWR,@SWREG ;;SOFTWARE SWITCH REG SELECTED?
1145 001654 001005          BNE     71$      ;;BRANCH IF NO
1146 001656 104406          GTSWR      ;;GET SOFT-SWR SETTINGS
1147 001660 000403          BR      71$
1148 001662 112767 000001 177244    70$:  MOVB   @1,@AUTOB ;;SET AUTO-MODE INDICATOR
1149 001670          71$:
1150 001670 000422          BR      68$    ;;GET OVER THE ASCIZ
1151                                ;;69$: .ASCIZ <CRLF>*ZZ-CVDVA-D DLV11-E OFFLINE TEST*<CRLF>
1152                                68$:
1153                                LET INITFLAG := #1
1154 001736 012767 000001 010410    MOV     @1,INITFLAG
1155 001744          LOOP:
1156 001744          CALL  CYCLE      ; NO ARGUMENTS--ADRS -> NEXT ADDRESS
1157 001744 004767 010254          JSR    PC,CYCLE
1158
1159                                ;
1160                                ADDR*2 -> NEXT VECTOR
1160 001750          ;GET UNIT ADDRESS
1161 001750 012167 177300          MOV     (ADRS)+,DLADD
1162                                DLADD := (ADRS)+
1163                                ;GET UNIT VECTOR
1163 001754          LET     DLVEC := (ADRS)
1164 001754 011167 177276          MOV     (ADRS),DLVEC
1165 001760          LET     ADRS := DLADD
1166 001760 016701 177270          MOV     DLADD,ADRS
1167                                ;RCSR = DLADD * 0
1168 001764          LET     RCSR := DLADD
1169 001764 016767 177264 177266    MOV     DLADD,RCSR
1170 001772          LET     RBUF := DLADD * #2
1171 001772 016767 177256 177262    MOV     DLADD,RBUF
1172 002000 062767 000002 177254    ADD     @2,RBUF
1173 002006          LET     TCSR := DLADD * #4

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 31  
 CVDVAD.P11 12-JUL-84 05:04 GET VALUE FOR SOFTWARE SWITCH REGISTER

1174	002006	016767	177242	177250	MOV	DLADD,TCSR		
1175	002014	062767	000004	177242	ADD	#4,TCSR		
1176	002022						LET	TCSRHI := DLADD + #5
1177	002022	016767	177226	177236	MOV	DLADD,TCSRHI		
1178	002030	062767	000005	177230	ADD	#5,TCSRHI		
1179	002036						LET	TBUF := DLADD + #6
1180	002036	016767	177212	177224	MOV	DLADD,TBUF		
1181	002044	062767	000006	177216	ADD	#6,TBUF		
1182	002052						LET	R5 := #RSSTACK
1183	002052	012705	001334		MOV	#RSSTACK,R5		
1184								::BRESET
1185	002056	000005			RESET			

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 32  
 CVDVAD.P11 12-JUL-84 05:04 T1 ADDRESSABILITY

```

1186
1187
1188
1189
1190
1191
1192 002060 000004
1193 002062 012767 000002 177070
1194 002070 012767 000001 177102
1195 002076
1196 002076 016701 177152
1197
1198 002102
1199 002102 010146
1200 002104 012701 000004
1201 002110 012721 012032
1202 002114 012711 000340
1203 002120 012601
1204 002122
1205 002122 005067 177144
1206 002126
1207 002126
1208 002126
1209 002126 012767 002134 176754
1210
1211 002134
1212 002134 005067 007700
1213
1214
1215 002140 005711
1216 002142
1217 002142 005767 007672
1218 002146 001401
1219
1220 002150
1221 002150 104001
1222 002152
1223 002152
1224 002152
1225 002152
1226 002152 062767 000002 177112
1227 002160
1228 002160 016701 177070
1229 002164 066701 177102
1230 002170
1231 002170 026727 177076 000010
1232 002176 001353
1233 002200
1234 002200 010146
1235 002202 010246
1236 002204 012701 000004
1237 002210 010102
1238 002212 062702 000002
1239 002216 010221
1240 002220 005011
1241 002222 012602

;*****
;*TEST 1 ADDRESSABILITY
;* THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
;* THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
;* TO THAT ADDRESS SPACE
;*****
TST1: SCOPE
MOV #2, #TIMES ;DO 2 ITERATIONS
MOV #1, #TESTN ;SET TEST NUMBER IN APT MAIL BOX
LET ADRS := DLADD
MOV DLADD, ADRS
SETVEC ; SET UP INTERRUPT
; ILLMEM, #INTSRV, #PR7
MOV R1, -(SP)
MOV #ILLMEM, R1
MOV #INTSRV, (R1)+
MOV #PR7, (R1)+
MOV (SP)+, R1
LET I := #0
REPEAT
BGNSUB
; CLEAR FLAG
LET INTFLAG := #0
; READ FLAG
IF INTFLAG NE #0 THEN
; FATAL ERROR
ERRDF 1, .NODL
ENDIF
ENDSUB
LET I := I + #2
LET ADRS := DLADD + I
UNTIL I EQ #8.
CLRVEC ILLMEM
; PUSH R1 ON STACK
; PUSH R2 ON STACK
MOV R1, -(SP)
MOV R2, -(SP)
MOV #ILLMEM, R1
MOV R1, R2
ADD #2, R2
MOV R2, (R1)+
CLR (R1)
MOV (SP)+, R2 ; POP STACK INTO R2
    
```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 33  
CVDVAD.P11 12-JUL-84 05:04 T1 ADDRESSABILITY

```

1242 002224 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
1243                                     ;;END OF TEST
1244 002226                                     ENDTST
1245      ;;*****
1246      ;;* THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
1247      ;;*****
1248
1249

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 34  
 CVDVAD.P11 12-JUL-84 05:04 T2 BREAK - TCSRO SET, CLEAR, RESET

```

1250 ;*****
1251 ;*TEST 2 BREAK - TCSRO SET, CLEAR, RESET
1252 ;* NOTE: THE (H) JUMPER MUST BE REMOVED FOR THIS
1253 ;* TEST TO FUNCTION PROPERLY.
1254 ;*****
1255 002226 000004 TST2: SCOPE
1256 002230 012767 000010 176722 MOV #10,#TIMES ;:DO 10 ITERATIONS
1257 002236 012767 000002 176734 MOV #2,#TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1258
1259 002244 IF #BRK NOTSETIN #USWR THEN
1260 002244 032767 010000 176746 BIT #BRK,#USWR
1261 002252 001004 BNE #3
1262 002254 EXIT TST
1263 002254 012767 000001 176676 MOV #1,#TIMES
1264 002262 000452 BR TST3 ;:EXIT THIS TEST
1265 002264 ENDIF
1266 002264 ;3:
1267 ; SEE IF IT IS CLEAR
1268 002264 BGNSUB
1269 002264 012767 002272 176616 MOV #64#,#LPERR
1270
1271 002272 IF #BREAK SETIN @TCSR THEN
1272 002272 032777 000001 176764 BIT #BREAK,@TCSR
1273 002300 001401 BEQ #4
1274 ; BREAK DID NOT RESET IN TCSR
1275 002302 ERRHRD 2.,DIDNOT
1276 002302 104002 ERROR 2
1277 002304 ENDF
1278 002304 ;4:
1279 002304 ENDSUB
1280
1281 ; TRY TO SET BREAK BIT
1282 002304 BGNSUB
1283 002304 012767 002312 176576 MOV #64#,#LPERR
1284 002312 LET @TCSR := @TCSR SET.BY #BREAK
1285 002312 052777 000001 176744 BIS #BREAK,@TCSR
1286 ; STUCK TO 0
1287 002320 IF #BREAK NOTSETIN @TCSR THEN
1288 002320 032777 000001 176736 BIT #BREAK,@TCSR
1289 002326 001001 BNE #5
1290 ; BREAK DID NOT SET IN TCSR
1291 002330 ERRHRD 3.,DIDNOT
1292 002330 104003 ERROR 3
1293 002332 ENDF
1294 002332 ;5:
1295 002332 ENDSUB
1296
1297 ; TRY TO CLEAR A SET BIT
1298 002332 BGNSUB
1299 002332 012767 002340 176550 MOV #64#,#LPERR
1300
1301 LET @TCSR := @TCSR CLR.BY #BREAK
1302 002340 042777 000001 176716 BIC #BREAK,@TCSR
1303 ; SHOULD HAVE CLEARED
1304 002346 IF #BREAK SETIN @TCSR THEN
1305 002346 032777 000001 176710 BIT #BREAK,@TCSR

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 35  
CVDVAD.P11 12-JUL-84 05:04 T2 BREAK - TCSRO SET, CLEAR, RESET

```

1306 002354 001401          BEQ      #6
1307                                     ; BREAK DID NOT CLEAR IN TCSR
1308 002356                                     ERRHRD 4,,DIDNOT
1309 002356 104004          ERROR    4
1310 002360                                     ENDIF
1311 002360          $6:
1312 002360                                     ENDSUB
1313
1314                                     ; NOW SEE IF RESET CLEARS IT
1315 002360                                     BGNSUB
1316 002360 012767 002366 176522      MOV      #64#,#LPERR
1317
1318 002366                                     LET      @TCSR := @TCSR SET.BY #BREAK
1319 002366 052777 000001 176670      BIS      #BREAK,@TCSR
1320                                     ; ISSUE BUS RESET
1321 002374                                     BRESET
1322 002374 000005          RESET
1323 002376                                     IF      #BREAK SETIN @TCSR THEN
1324 002376 032777 000001 176660      BIT      #BREAK,@TCSR
1325 002404 001401          BEQ      #7
1326                                     ; BREAK DID NOT RESET IN TCSR
1327 002406                                     ERRHRD 5,,DIDNOT
1328 002406 104005          ERROR    5
1329 002410                                     ENDIF
1330 002410          $7:
1331 002410                                     ENDSUB
1332 002410                                     ENDTST
1333
1334
1335          ;;*****

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 36  
CVDVAD.P11 12-JUL-84 05:04 T3 MAINT - TCSR2 SET, CLEAR, RESET

```

1336 ;*****
1337 ;*TEST 3 MAINT - TCSR2 SET, CLEAR, RESET
1338 ;*****
1339 002410 000004 TST3: SCOPE
1340 002412 012767 000010 176540 MOV #10,#TIMES ;DO 10 ITERATIONS
1341 002420 012767 000003 176552 MOV #3,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1342
1343 ; SEE IF IT IS CLEAR
1344 002426 BGNSUB
1345 002426 012767 002434 176454 MOV #64#,#LPERR
1346
1347 IF #MAINT SETIN @TCSR THEN
1348 002434 032777 000004 176622 BIT #MAINT,@TCSR
1349 002442 001401 BEQ #10
1350 ; MAINT DID NOT RESET IN TCSR
1351 002444 ERRHRD 6,,DIDNOT
1352 002444 104006 ERROR 6
1353 002446 ENDF
1354 002446 $10:
1355 002446 ENDSUB
1356
1357 ; TRY TO SET MAINT BIT
1358 002446 BGNSUB
1359 002446 012767 002454 176434 MOV #64#,#LPERR
1360 002454 LET @TCSR := @TCSR SET.BY #MAINT
1361 002454 052777 000004 176602 BIS #MAINT,@TCSR
1362 ; STUCK TO 0
1363 002462 IF #MAINT NOTSETIN @TCSR THEN
1364 002462 032777 000004 176574 BIT #MAINT,@TCSR
1365 002470 001001 BNE #11
1366 ; MAINT DID NOT SET IN TCSR
1367 002472 ERRHRD 7,,DIDNOT
1368 002472 104007 ERROR 7
1369 002474 ENDF
1370 002474 $11:
1371 002474 ENDSUB
1372
1373 ; TRY TO CLEAR A SET BIT
1374 002474 BGNSUB
1375 002474 012767 002502 176406 MOV #64#,#LPERR
1376
1377 LET @TCSR := @TCSR CLR.BY #MAINT
1378 002502 032777 000004 176554 BIC #MAINT,@TCSR
1379 ; SHOULD HAVE CLEARED
1380 002510 IF #MAINT SETIN @TCSR THEN
1381 002510 032777 000004 176546 BIT #MAINT,@TCSR
1382 002516 001401 BEQ #12
1383 ; MAINT DID NOT CLEAR INTCSR
1384 002520 ERRHRD 10,,DIDNOT
1385 002520 104010 ERROR 10
1386 002522 ENDF
1387 002522 $12:
1388 002522 ENDSUB
1389
1390 ; NOW SEE IF RESET CLEARS IT
1391 002522 BGNSUB

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 37  
CVDVAD.P11 12-JUL-84 05:04 T3 MAINT - TCSR2 SET, CLEAR, RESET

```

1392 002522 012767 002530 176360      MOV      @64@, @LPERR
1393
1394 002530                                LET      @TCSR := @TCSR SET.BY @MAINT
1395 002530 052777 000004 176526      BIS      @MAINT, @TCSR
1396                                     ; ISSUE BUS RESET
1397 002536                                BRESET
1398 002536 000005      RESET
1399 002540                                IF      @MAINT SET IN @TCSR THEN
1400 002540 032777 000004 176516      BIT      @MAINT, @TCSR
1401 002546 001401      BEQ      @13
1402                                     ; MAINT DID NOT RESET IN TCSR
1403 002550                                ERRHRD 11,, DIDNOT
1404 002550 104011      ERROR 11
1405 002552                                ENDIF
1406 002552      @13:                                ENDSUB
1407 002552                                ENDTST
1408 002552
1409
1410
1411
1412      ;*****

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 38  
 CVDVAD.P11 12-JUL-84 05:04 T4 XMITIE - TCSR6 SET, CLEAR, RESET

```

1413 ;*****
1414 ;*TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
1415 ;*****
1416 002552 000004 TST4: SCOPE
1417 002554 012767 000010 176376 MOV #10,#TIMES ;DO 10 ITERATIONS
1418 002562 012767 000004 176410 MOV #4,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1419 ; USE PRIORITY OF 7
1420 002570 012746 000340 MOV #PR7,-(SP) ;PUT NEW PS ON STACK
1421 002574 012746 002602 MOV #64#,-(SP) ;PUT NEW PC ON STACK
1422 002600 000002 RTI ;POP NEW PC AND PS
1423 002602 64#:
1424
1425 ; SEE IF IT IS CLEAR
1426 002602 ; BGNSUB
1427 002602 012767 002610 176300 MOV #65#,#LPERR
1428
1429 002610 IF #XMITIE SETIN @TCSR THEN
1430 002610 032777 000100 176446 BIT #XMITIE,@TCSR
1431 002616 001401 BEQ #14
1432 ; XMITIE DID NOT RESET IN TCSR
1433 002620 ERRHRD 12,,DIDNOT
1434 002620 104012 ERROR 12
1435 002622 ENDIF
1436 002622 #14:
1437 002622 ENDSUB
1438
1439 ; TRY TO SET XMITIE BIT
1440 002622 ; BGNSUB
1441 002622 012767 002630 176260 MOV #64#,#LPERR
1442 002630 LET @TCSR := @TCSR SET.BY #XMITIE
1443 002630 052777 000100 176426 BIS #XMITIE,@TCSR
1444 ; STUCK TO 0
1445 002636 IF #XMITIE NOTSETIN @TCSR THEN
1446 002636 032777 000100 176420 BIT #XMITIE,@TCSR
1447 002644 001001 BNE #15
1448 ; XMIT DID NOT RESET IN TCSR
1449 002646 ERRHRD 13,,DIDNOT
1450 002646 104013 ERROR 13
1451 002650 ENDIF
1452 002650 #15:
1453 002650 ENDSUB
1454
1455 ; TRY TO CLEAR A SET BIT
1456 002650 ; BGNSUB
1457 002650 012767 002656 176232 MOV #64#,#LPERR
1458
1459 002656 LET @TCSR := @TCSR CLR.BY #XMITIE
1460 002656 042777 000100 176400 BIC #XMITIE,@TCSR
1461 ; SHOULD HAVE CLEARED
1462 002664 IF #XMITIE SETIN @TCSR THEN
1463 002664 032777 000100 176372 BIT #XMITIE,@TCSR
1464 002672 001401 BEQ #16
1465 ; XMIT DID NOT CLEAR IN TCSR
1466 002674 ERRHRD 14,,DIDNOT
1467 002674 104014 ERROR 14
1468 002676 ENDIF

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 40  
 CVDVAD.P11 12-JUL-84 05:04 T5 DTR - RCSR1 SET, CLEAR

```

1495
1496
1497
1498
1499
1500
1501
1502
1503 002726 000004
1504 002730 012767 000010 176222
1505 002736 012767 000005 176234
1506 002744
1507 002744 032767 040000 176246
1508 002752 001004
1509 002754
1510 002754 012767 000001 176176
1511 002762 000441
1512 002764
1513 002764
1514
1515 002764
1516 002764 012767 002772 176116
1517 002772
1518 002772 042777 000002 176260
1519
1520 003000
1521 003000 032777 000002 176252
1522 003006 001401
1523
1524 003010
1525 003010 104016
1526 003012
1527 003012
1528 003012
1529
1530
1531 003012
1532 003012 012767 003020 176070
1533
1534 003020
1535 003020 052777 000002 176232
1536 003026
1537 003026 032777 000002 176224
1538 003034 001001
1539
1540 003036
1541 003036 104017
1542 003040
1543 003040
1544 003040
1545
1546
1547 003040
1548 003040 012767 003046 176042
1549 003046
1550 003046 042777 000002 176204

```

```

*****
; *TEST 5      DTR - RCSR1  SET, CLEAR
; *           NOTE:  RESET DOES NOT CLEAR THIS BIT
; *           WE CANNOT TEST FOR AN INITIAL CONDITION
; *           AS THIS BIT IS UNDEFINED UPON POWER UP AND
; *           INIT DOESN'T AFFECT IT.
; *           THE (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK.
*****
TST5:  SCOPE
      MOV     #10, #TIMES      ;;DO 10 ITERATIONS
      MOV     #5, #TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                               IF #FRFD NOTSETIN #USWR THEN
      BIT     #FRFD, #USWR
      BNE     $20
                               EXIT TST
      MOV     #1, #TIMES
      BR     TST6              ;;EXIT THIS TEST
                               ENDIF
$20:
      ; TRY TO CLEAR DTR BIT
      BGNSUB
      MOV     #64, #LPERR
      LET     @RCSR := @RCSR CLR.BY #DTR
      BIC     #DTR, @RCSR
      ; STUCK TO 0
      IF     #DTR SETIN @RCSR THEN
      ; DTR DID NOT CLEAR IN RCSR
      ERRHRD 16,,DIDNOT
      ENDIF
$21:
      ENDSUB
      ; TRY TO SET DTR
      BGNSUB
      MOV     #64, #LPERR
      LET     @RCSR := @RCSR SET.BY #DTR
      BIS     #DTR, @RCSR
      IF     #DTR NOTSETIN @RCSR THEN
      ; DTR DID NOT SET IN RCSR
      ERRHRD 17,,DIDNOT
      ENDIF
$22:
      ENDSUB
      ; TRY TO CLEAR IT AGAIN
      BGNSUB
      MOV     #64, #LPERR
      LET     @RCSR := @RCSR CLR.BY #DTR
      BIC     #DTR, @RCSR

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 41  
CVDVAD.P11 12-JUL-84 05:04 T5 DTR - RCSR1 SET, CLEAR

1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565

003054  
003054 032777 000002 176176  
003062 001401  
003064  
003064 104020  
003066  
003066  
003066  
003066

BIT #DTR,@RCSR  
BEQ \$23  
  
ERROR 20

\$23:

IF ; SHOULD HAVE CLEARED IT  
; @DTR SETIN @RCSR THEN

; DTR DID NOT CLEAR IN RCSR  
ERRHRD 20.,.DIDNOT

ENDIF

ENDSUB  
ENDTST

::\*\*\*\*\*



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 42  
CVDVAD.P11 12-JUL-84 05:04 T6 REQSEND - RCSR2 SET, CLEAR, RESET

```

1566 ;*****
1567 ;*TEST 6          REQSEND - RCSR2          SET, CLEAR, RESET
1568 ;*              THIS TEST ASSUMES THAT JUMPER -(FR) IS IN
1569 ;*****
1570 003066 000004 TST6: SCOPE
1571 003070 012767 000010 176062 MOV #10,#TIMES ;DO 10 ITERATIONS
1572 003076 012767 000006 176074 MOV #6,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1573 003104 ; ; IF #FRFD NOTSETIN #USWR THEN
1574 003104 032767 040000 176106 BIT #FRFD,#USWR
1575 003112 001004 BNE $24
1576 003114 ; ; EXIT TST
1577 003114 012767 000001 176036 MOV #1,#TIMES
1578 003122 000452 BR TST7 ;EXIT THIS TEST
1579 003124 ; ; ENDF
1580 003124 $24:
1581 ; ;
1582 ; ; SEE IF IT IS CLEAR
1583 003124 ; ; BGNSUB
1584 003124 012767 003132 175756 MOV #64,#LPERR
1585 ; ;
1586 003132 ; ; IF #REQSEND SETIN #RCSR THEN
1587 003132 032777 000004 176120 BIT #REQSEND,#RCSR
1588 003140 001401 BEQ $25
1589 ; ;
1590 003142 ; ; ; REQSEND DID NOT RESET IN RCSR
1591 003142 104021 ERROR 21 ;ERRRD 21..DIDNOT
1592 003144 ; ; ENDF
1593 003144 $25:
1594 003144 ; ; ENDSUB
1595 ; ;
1596 ; ; TRY TO SET REQSEND BIT
1597 003144 ; ; BGNSUB
1598 003144 012767 003152 175736 MOV #64,#LPERR
1599 003152 ; ; LET #RCSR := #RCSR SET.BY #REQSEND
1600 003152 052777 000004 176100 BIS #REQSEND,#RCSR
1601 ; ;
1602 003160 ; ; IF #REQSEND NOTSETIN #RCSR THEN
1603 003160 032777 000004 176072 BIT #REQSEND,#RCSR
1604 003166 001001 BNE $26
1605 ; ;
1606 003170 ; ; ; REQSEND DID NOT SET IN RCSR
1607 003170 104022 ERROR 22 ;ERRRD 22..DIDNOT
1608 003172 ; ; ENDF
1609 003172 $26:
1610 003172 ; ; ENDSUB
1611 ; ;
1612 ; ; TRY TO CLEAR A SET BIT
1613 003172 ; ; BGNSUB
1614 003172 012767 003200 175710 MOV #64,#LPERR
1615 ; ;
1616 003200 ; ; LET #RCSR := #RCSR CLR.BY #REQSEND
1617 003200 042777 000004 176052 BIC #REQSEND,#RCSR
1618 ; ;
1619 003206 ; ; IF #REQSEND SETIN #RCSR THEN
1620 003206 032777 000004 176044 BIT #REQSEND,#RCSR
1621 003214 001401 BEQ $27

```

```

1622                                     ; REQSEND DID NOT CLEAR IN RCSR
1623 003216                                     ERRMRD 23,,DIDNOT
1624 003216 104023          ERROR 23
1625 003220                                     ENDIF
1626 003220          $27:
1627 003220                                     ENDSUB
1628
1629                                     ; NOW SEE IF RESET CLEARS IT
1630 003220                                     BGNSUB
1631 003220 012767 003226 175662      MOV      #64$, $LPERR
1632
1633 003226                                     LET      BRCSR := BRCSR SET.BY #REQSEND
1634 003226 052777 000004 176024      BIS      #REQSEND, BRCSR
1635                                     ; ISSUE BUS RESET
1636 003234                                     BRESET
1637 003234 000005      RESET
1638 003236                                     IF      #REQSEND SETIN BRCSR THEN
1639 003236 032777 000004 176014      BIT      #REQSEND, BRCSR
1640 003244 001401      BEQ      $30
1641                                     ; REQSEND DID NOT RESET IN RCSR
1642 003246                                     ERRMRD 24,,DIDNOT
1643 003246 104024          ERROR 24
1644 003250                                     ENDIF
1645 003250          $30:
1646 003250                                     ENDSUB
1647 003250      ENDTST
1648
1649
1650
1651      ; .....

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 44  
 CVDVAD.P11 12-JUL-84 05:04 T7 SECXMIT - RCSR3 SET, CLEAR, RESET

```

1652
1653
1654
1655 003250 000004
1656 003252 012767 000010 175700
1657 003260 012767 000007 175712
1658
1659 003266
1660 003266 012767 003274 175614
1661
1662 003274
1663 003274 032777 000010 175756
1664 003302 001401
1665
1666 003304
1667 003304 104025
1668 003306
1669 003306
1670 003306
1671
1672
1673 003306
1674 003306 012767 003314 175574
1675 003314
1676 003314 052777 000010 175736
1677
1678 003322
1679 003322 032777 000010 175730
1680 003330 001001
1681
1682 003332
1683 003332 104026
1684 003334
1685 003334
1686 003334
1687
1688
1689 003334
1690 003334 012767 003342 175546
1691
1692 003342
1693 003342 042777 000010 175710
1694
1695 003350
1696 003350 032777 000010 175702
1697 003356 001401
1698
1699 003360
1700 003360 104027
1701 003362
1702 003362
1703 003362
1704
1705 003362
1706 003362 012767 003370 175520
1707

```

```

;*****
;TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET
;*****
TST7: SCOPE
MOV #10,#TIMES ;DO 10 ITERATIONS
MOV #7,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
; SEE IF IT IS CLEAR
BGNSUB
MOV #64#,#LPERR
IF #SECXMIT SETIN @RCSR THEN
BIT #SECXMIT,@RCSR
BEQ #31
; SECXMIT DID NOT RESET IN RCSR
ERRHRD 25,,DIDNOT
ERROR 25
ENDIF
;31:
ENDSUB
; TRY TO SET SECXMIT BIT
BGNSUB
MOV #64#,#LPERR
LET @RCSR := @RCSR SET.BY #SECXMIT
BIS #SECXMIT,@RCSR
; STUCK TO 0
IF #SECXMIT NOTSETIN @RCSR THEN
BIT #SECXMIT,@RCSR
BNE #32
; SECXMIT DID NOT SET IN RCSR
ERRHRD 26,,DIDNOT
ERROR 26
ENDIF
;32:
ENDSUB
; TRY TO CLEAR A SET BIT
BGNSUB
MOV #64#,#LPERR
LET @RCSR := @RCSR CLR.BY #SECXMIT
BIC #SECXMIT,@RCSR
; SHOULD HAVE CLEARED
IF #SECXMIT SETIN @RCSR THEN
BIT #SECXMIT,@RCSR
BEQ #33
; SECXMIT DID NOT CLEAR IN RCSR
ERRHRD 27,,DIDNOT
ERROR 27
ENDIF
;33:
ENDSUB
BGNSUB
MOV #64#,#LPERR
; NOW SEE IF RESET CLEARS IT

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 45  
CVDVAD.P11 12-JUL-84 05:04 T7 SECXMIT - RCSR3

SET, CLEAR, RESET

```

1708
1709 003370          LET   @RCSR := @RCSR SET.BY @SECXMIT
1710 003370 052777 000010 175662      BIS   @SECXMIT,@RCSR          ; ISSUE BUS RESET
1711                                     BRESET
1712 003376          RESET
1713 003376 000005
1714 003400          IF    @SECXMIT SETIN @RCSR THEN
1715 003400 032777 000010 175652      BIT   @SECXMIT,@RCSR
1716 003406 001401      BEQ   $34          ; SECXMIT DID NOT RESET IN RCSR
1717                                     ERRMRD 30,,DIDNOT
1718 003410          ERROR 30
1719 003410 104030
1720 003412          ENDIF
1721 003412          $34:
1722 003412          ENDSUB
1723 003412          ENDTST
1724
1725
1726
1727          ;;.....

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 46  
CVDVAD.P11 12-JUL-84 05:04 T10 DATAIE - RCSR5 SET, CLEAR, RESET

```

1728                                     ;*****
1729 ;*TEST 10          DATAIE - RCSR5 SET, CLEAR, RESET
1730                                     ;*****
1731 003412 000004 TST10: SCOPE
1732 003414 012767 000010 175536      MOV    #10,#TIMES      ;;DO 10 ITERATIONS
1733 003422 012767 000010 175550      MOV    #10,#TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1734                                     ; SEE IF IT IS CLEAR
1735 003430                                     BGNSUB
1736 003430 012767 003436 175452      MOV    #64#,#LPERR
1737
1738 003436                                     IF    #DATAIE SETIN @RCSR THEN
1739 003436 032777 000040 175614      BIT    #DATAIE,@RCSR
1740 003444 001401                                     BEQ    #35
1741                                     ; DATAIE DID NOT RESET IN RCSR
1742 003446                                     ERRHRD 31,,DIDNOT
1743 003446 104031      ERROR 31
1744 003450                                     ENDIF
1745 003450 #35:
1746 003450                                     ENDSUB
1747
1748                                     ; TRY TO SET DATAIE BIT
1749 003450                                     BGNSUB
1750 003450 012767 003456 175432      MOV    #64#,#LPERR
1751 003456                                     LET    @RCSR := @RCSR SET.BY #DATAIE
1752 003456 052777 000040 175574      BIS    #DATAIE,@RCSR
1753                                     ; STUCK TO 0
1754 003464                                     IF    #DATAIE NOTSETIN @RCSR THEN
1755 003464 032777 000040 175566      BIT    #DATAIE,@RCSR
1756 003472 001001                                     BNE    #36
1757                                     ; DATAIE DID NOT SET IN RCSR
1758 003474                                     ERRHRD 32,,DIDNOT
1759 003474 104032      ERROR 32
1760 003476                                     ENDIF
1761 003476 #36:
1762 003476                                     ENDSUB
1763
1764                                     ; TRY TO CLEAR A SET BIT
1765 003476                                     BGNSUB
1766 003476 012767 003504 175404      MOV    #64#,#LPERR
1767
1768 003504                                     LET    @RCSR := @RCSR CLR.BY #DATAIE
1769 003504 042777 000040 175546      BIC    #DATAIE,@RCSR
1770                                     ; SHOULD HAVE CLEARED
1771 003512                                     IF    #DATAIE SETIN @RCSR THEN
1772 003512 032777 000040 175540      BIT    #DATAIE,@RCSR
1773 003520 001401                                     BEQ    #37
1774                                     ; DATAIE DID NOT CLEAR IN RCSR
1775 003522                                     ERRHRD 33,,DIDNOT
1776 003522 104033      ERROR 33
1777 003524                                     ENDIF
1778 003524 #37:
1779 003524                                     ENDSUB
1780
1781                                     ; NOW SEE IF RESET CLEARS IT
1782 003524                                     BGNSUB
1783 003524 012767 003532 175356      MOV    #64#,#LPERR

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 47  
CVDVAD.P11 12-JUL-84 05:04 T10 DATAIE - RCSR5 SET, CLEAR, RESET

```

1784
1785 003532          LET @RCSR := @RCSR SET.BY #DATAIE
1786 003532 052777 000040 175520      BIS #DATAIE,@RCSR      ; ISSUE BUS RESET
1787                                     BRESET
1788 003540          IF #DATAIE SETIN @RCSR THEN
1789 003540 000005      RESET
1790 003542          ; DATAIE DID NOT RESET IN RCSR
1791 003542 032777 000040 175510      BIT #DATAIE,@RCSR      ERRHRD 34,,DIDNOT
1792 003550 001401      BEQ #40
1793                                     ; DATAIE DID NOT RESET IN RCSR
1794 003552          ; DATAIE DID NOT RESET IN RCSR
1795 003552 104034      ERROR 34      ERRHRD 34,,DIDNOT
1796 003554          ENDF
1797 003554          $40:
1798 003554          ENDSUB
1799 003554          ENDTST
1800
1801
1802
1803 ;*****

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 48  
CVDVAD.P11 12-JUL-84 05:04 T11 RCVRIE - RCSR6 SET, CLEAR, RESET

```

1804 ;*****
1805 ;*TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET
1806 ;*****
1807 003554 000004 TST11: SCOPE
1808 003556 012767 000010 175374 MOV #10,#TIMES ;DO 10 ITERATIONS
1809 003564 012767 000011 175406 MOV #11,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1810 ; SEE IF IT IS CLEAR
1811 003572 ; BGNSUB
1812 003572 012767 003600 175310 MOV #64#,#LPERR
1813
1814 003600 IF #RCVRIE SETIN @RCSR THEN
1815 003600 032777 000100 175452 BIT #RCVRIE,@RCSR
1816 003606 001401 BEQ #41
1817 ; RCVRIE DID NOT RESET IN RCSR
1818 003610 ; ERRHRD 35,,DIDNOT
1819 003610 104035 ERROR 35
1820 003612 ENDF
1821 003612 ;41: ENDSUB
1822 003612
1823
1824 ; TRY TO SET RCVRIE BIT
1825 003612 ; BGNSUB
1826 003612 012767 003620 175270 MOV #64#,#LPERR
1827 003620 LET @RCSR := @RCSR SET.BY #RCVRIE
1828 003620 052777 000100 175432 BIS #RCVRIE,@RCSR
1829 ; STUCK TO 0
1830 003626 IF #RCVRIE NOTSETIN @RCSR THEN
1831 003626 032777 000100 175424 BIT #RCVRIE,@RCSR
1832 003634 001001 BNE #42
1833 ; RCVRIE DID NOT SET IN RCSR
1834 003636 ; ERRHRD 36,,DIDNOT
1835 003636 104036 ERROR 36
1836 003640 ENDF
1837 003640 ;42: ENDSUB
1838 003640
1839
1840 ; TRY TO CLEAR A SET BIT
1841 003640 ; BGNSUB
1842 003640 012767 003646 175242 MOV #64#,#LPERR
1843
1844 003646 LET @RCSR := @RCSR CLR.BY #RCVRIE
1845 003646 042777 000100 175404 BIC #RCVRIE,@RCSR
1846 ; SHOULD HAVE CLEARED
1847 003654 IF #RCVRIE SETIN @RCSR THEN
1848 003654 032777 000100 175376 BIT #RCVRIE,@RCSR
1849 003662 001401 BEQ #43
1850 ; RCVRIE DID NOT CLEAR IN RCSR
1851 003664 ; ERRHRD 37,,DIDNOT
1852 003664 104037 ERROR 37
1853 003666 ENDF
1854 003666 ;43: ENDSUB
1855 003666
1856
1857 ; NOW SEE IF RESET CLEARS IT
1858 003666 ; BGNSUB
1859 003666 012767 003674 175214 MOV #64#,#LPERR

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 49  
CVDVAD.P11 12-JUL-84 05:04 T11 RCVRIE - RCSR6 SET, CLEAR, RESET

```

1860
1861 003674          LET  @RCSR := @RCSR SET.BY #RCVRIE
1862 003674 052777 000100 175356  BIS  @RCVRIE,@RCSR      ; ISSUE BUS RESET
1863                                     BRESET
1864 003702          RESET
1865 003702 000005
1866 003704          IF  @RCVRIE SETIN @RCSR THEN
1867 003704 032777 000100 175346  BIT  @RCVRIE,@RCSR
1868 003712 001401  BEQ  $44
1869                                     ; RCVRIE DID NOT RESET IN RCSR
1870 003714          ERRHRD 40,,DIDNOT
1871 003714 104040  ERROR 40
1872 003716
1873 003716          $44:
1874 003716          ENDIF
1875 003716          CKLOOP
1876 003716          ENDSUB
1877                                     ENDTST
1878
1879
1880
1881
1882
1883
1884
1885
1886

```

```

;*****
;* THE FOLLOWING 4 TESTS VERIFY
;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
;*****

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 50  
CVDVAD.P11 12-JUL-84 05:04 T12 TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT

```

1887
1888
1889
1890 003716 000004
1891 003720 012767 000010 175232
1892 003726 012767 000012 175244
1893
1894
1895
1896
1897
1898 003734
1899 003734 012767 003742 175146
1900 003742
1901 003742 032777 000200 175310
1902 003750 001402
1903
1904
1905
1906 003752
1907 003752 104041
1908
1909 003754
1910 003754 000005
1911 003756
1912 003756
1913
1914 003756
1915 003756
1916 003756
1917
1918
1919

```

```

;*****
;*TEST 12 TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
;*****
TST12: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #12,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX

BGNSUB
MOV #64,$LPERR
IF #RCVRDONE SETIN @RCSR THEN
BIT #RCVRDONE,@RCSR
BEQ $45
;RCVRDONE SHOULD HAVE CLEARED BY INIT
; RCVRDONE DID NOT CLEAR IN RCSR
ERRHRD 41,HRESET, DIDNOT
;REISSUE RESET
BRESET
ENDIF
;ALLOW LOOPING AFTER ERROR
CKLOOP
ENDSUB
ENDTST

```

```

$45:

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 51  
CVDVAD.P11 12-JUL-84 05:04 T13 TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT

1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952

003756 000004  
003760 012767 000010 175172  
003766 012767 000013 175204  
  
003774  
003774 012767 004002 175106  
  
004002  
004002 032777 000200 175254  
004010 001002  
  
004012  
004012 104042  
  
004014  
004014 000005  
004016  
004016  
004016  
004016  
004016

```
;;*****  
;TEST 13 TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT  
;;*****  
TST13: SCOPE  
MOV @10,$TIMES ;;DO 10 ITERATIONS  
MOV @13,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX  
  
BGNSUB  
MOV @64,$LPERR  
  
IF @XMITRDY NOTSETIN @TCSR THEN  
BIT @XMITRDY,@TCSR  
BNE $46  
  
;RESET SHOULD HAVE SET BIT.  
;XMITRDY DID NOT SET IN TCSR (AFTER RESE  
ERRHRD 42,HRESET,DIDNOT  
  
;ISSUE ANOTHER RESET  
BRESET  
  
ENDIF  
  
;ALLOW LOOPING ON ERROR  
CKLOOP  
ENDSUB  
ENDTST
```

\$46:

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 52  
CVDVAD.P11 12-JUL-84 05:04 T14 TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.

```

1953 ;:*****
1954 ;*TEST 14 TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
1955 ;:*****
1956 004016 000004 TST14: SCOPE
1957 004020 012767 000010 175132 MOV #10,$TIMES ;:DO 10 ITERATIONS
1958 004026 012767 000014 175144 MOV #14,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1959
1960
1961
1962
1963 004034 BGNSUB
1964 004034 012767 004042 175046 MOV #64,$LPERR
1965 004042 IF #DATAINT SETIN @RCSR THEN
1966 004042 032777 100000 175210 BIT #DATAINT,@RCSR
1967 004050 001402 BEQ $47
1968
1969 004052 ERRHRD 43, HRESET, DIDNOT
1970 004052 104043 ERROR 43
1971
1972 ;:TESTING EFFECT OF RESET ON BIT
1973
1974 ;:DATAINT DID NOT CLEAR IN RCSR
1975 ;:ALLOW A FRESH START
1976 004054 BRESET
1977 004054 000005 RESET
1978 004056 ENDIF
1979 004056 $47:
1980 004056 CKLOOP
1981 004056 ENDSUB
1982 004056 ENDTST
1983
1984
1985
1986 ;:*****

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 53  
 CVDVAD.P11 12-JUL-84 05:04 T15 TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT

```

1987
1988
1989
1990 004056 000004
1991 004060 012767 000010 175072
1992 004066 012767 000015 175104
1993
1994
1995 004074
1996 004074 032767 020000 175116
1997 004102 001004
1998
1999 004104
2000 004104 012767 000001 175046
2001 004112 000411
2002 004114
2003 004114
2004
2005
2006
2007 004114
2008 004114 012767 004122 174766
2009
2010 004122
2011 004122 032777 004000 175130
2012 004130 001402
2013
2014
2015 004132
2016 004132 104044
2017
2018
2019
2020
2021
2022
2023 004134
2024 004134 000005
2025 004136
2026 004136
2027
2028 004136
2029 004136
2030 004136
2031

;*****
;*TEST 15 TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
;*****
TST15: SCOPE
MOV #10,#TIMES ;;DO 10 ITERATIONS
MOV #15,#TESTN ;;SET TEST NUMBER IN APT MAIL BOX

IF #CABLE NOTSETIN #USWR THEN
BIT #CABLE,#USWR
BNE #50
; CAN'T TEST WITHOUT BERG OR M315.
EXIT TST
MOV #1,#TIMES
BR TST16 ;;EXIT THIS TEST
ENDIF

$50:
BGNSUB
IF #RCVRACT SETIN #RCSR THEN
BIT #RCVRACT,#RCSR
BEQ #51
;RESET SHOULD HAVE CLEARED RCVRACT
ERRHRD 44, MRESET, DIDNOT
; TESTING EFFECT OF RESET ON BIT
;RCVRACT DID NOT CLEAR IN RCSR
;ALLOW ANOTHER TRY
BRESET
ENDIF
;ALLOW LOOPING ON ERROR
CKLOOP
ENDSUB
ENDTST
    
```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 54  
 CVDVAD.P1: 12-JUL-84 05:04 T15 TEST THAT RCVRCT - RCSR 11 - 15 CLEARED BY INIT

```

2032 ;*****
2033 ;* THE FOLLOWING 4 TESTS VERIFY
2034 ;* THAT THE EIA SIGNALS CAN BE TRANSMITTED
2035 ;* AND RECEIVED THROUGH THE CABLE
2036 ;*****
2037
2038
2039
2040 ;*****
2041 ;*TEST 16 TEST THAT CARDET SETS AND CLEARS
2042 ;* AS DTR SETS AND CLEARS
2043 ;* THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
2044 ;*****
2045 004136 000004 TST16: SCOPE
2046 004140 012767 000010 175012 MOV #10,$TIMES ;;DO 10 ITERATIONS
2047 004146 012767 000016 175024 MOV #16,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2048 ; CAN WE USE THE WRAPAROUND??
2049 004154 032767 060000 175036 BIT #CABLE+FRFD,$USWR IF #CABLE+FRFD NOTSETIN $USWR THEN
2050 004154 032767 060000 175036 BIT #CABLE+FRFD,$USWR
2051 004162 001004 BNE $52
2052 ; CAN'T TEST WITHOUT BERG OR H315
2053 ; OR WITH (-FD) JUMPER OUT.
2054 ; OR WITH (-FR) JUMPER OUT.
2055 004164 EXIT TST
2056 004164 012767 000001 174766 MOV #1,$TIMES
2057 004172 000441 BR TST17 ;;EXIT THIS TEST
2058 004174 ENDIF
2059 004174 $52:
2060
2061 ; DTR AND
2062 ; CARDET ARE CONNECTED
2063
2064 ; BY THE H315 OR EQUIV.
2065
2066 ; CLEAR
2067 004174 ; CLEAR BGNSUB
2068 004174 012767 004202 174706 MOV #64,$LPERR
2069
2070 ; CLEAR DTR
2071 004202 LET @RCSR := @RCSR CLR.BY @DTR
2072 004202 042777 000002 175050 BIC @DTR,@RCSR
2073 ; CARDET SHOULD FOLLOW
2074 004210 IF @CARDET SETIN @RCSR THEN
2075 004210 032777 010000 175042 BIT @CARDET,@RCSR
2076 004216 001401 BEQ $53
2077 ; CARDET DID NOT
2078 004220 ERRHRD 45,,FORCE
2079 004220 104045 ERROR 45
2080
2081 ; CLEAR WITH DTR
2082 004222 ENDIF
2083 004222 $53:
2084 004222 ENDSUB
2085
2086 ; SET
2087 004222 BGNSUB
    
```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 55  
CVDVAD.P11 12-JUL-84 05:04 T16 TEST THAT CARDET SETS AND CLEARS

```

2088 004222 012767 004230 174660      MOV      #641, #LPERR
2089
2090                                     ; SET DTR
2091 004230                                     LET      @RCSR := @RCSR SET.BY #DTR
2092 004230 052777 000002 175022      BIS      #DTR, @RCSR
2093                                     ; CARDET SHOULD FOLLOW
2094 004236                                     IF      #CARDET NOTSETIN @RCSR THEN
2095 004236 032777 010000 175014      BIT      #CARDET, @RCSR
2096 004244 001001                                     BNE      #54
2097                                     ; CARDET DID NOT SET
2098 004246                                     ERRHRD 46,,FORCE
2099 004246 104046      ERROR      46
2100
2101                                     ; WITH DTR
2102 004250                                     ENDIF
2103 004250      #54:
2104 004250                                     ENDSUB
2105
2106                                     ; CLEAR
2107 004250                                     BGNSUB
2108 004250 012767 004256 174632      MOV      #641, #LPERR
2109
2110                                     ; CLEAR DTR
2111 004256                                     LET      @RCSR := @RCSR CLR.BY #DTR
2112 004256 042777 000002 174774      BIC      #DTR, @RCSR
2113                                     ; CARDET SHOULD FOLLOW
2114 004264                                     IF      #CARDET SETIN @RCSR THEN
2115 004264 032777 010000 174766      BIT      #CARDET, @RCSR
2116 004272 001401                                     BEQ      #55
2117                                     ; CARDET DID NOT
2118 004274                                     ERRHRD 47,,FORCE
2119 004274 104047      ERROR      47
2120
2121                                     ; CLEAR WITH DTR
2122 004276                                     ENDIF
2123 004276      #55:
2124 004276                                     ENDSUB
2125 004276      ENDTST
2126
2127
2128
2129

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 56  
CVDVAD.P11 12-JUL-84 05:04 T16 TEST THAT CARDET SETS AND CLEARS

```

2130 ;:.....
2131 ;:.....
2132 ;*TEST 17 TEST THAT CLRSEND SETS AND CLEARS
2133 ;* AS DTR SETS AND CLEARS
2134 ;* (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK
2135 ;:.....
2136 004276 000004 TST17: SCOPE
2137 004300 012767 000010 174652 MOV #10,#TIMES ;:DO 10 ITERATIONS
2138 004306 012767 000017 174664 MOV #17,#TESTN ;:SET TEST NUMBER IN APT MAIL BOX
2139 ;: ; CAN WE USE THE WRAPAROUND??
2140 004314 IF #CABLE*FRFD NOTSETIN #USMR THEN
2141 004314 032767 060000 174676 BIT #CABLE*FRFD,#USMR
2142 004322 001004 BNE #56
2143 ;: ; CAN'T TEST WITHOUT BERG OR H315
2144 004324 EXIT TST
2145 004324 012767 000001 174626 MOV #1,#TIMES
2146 004332 000441 BR TST20 ;:EXIT THIS TEST
2147 004334 ENDF
2148 004334 #56:
2149 ;:
2150 ;: DTR AND
2151 ;: CLRSEND ARE CONNECTED
2152 ;:
2153 ;: BY THE H315 OR EQUIV.
2154 ;:
2155 ;: CLEAR
2156 004334 BGNSUB
2157 004334 012767 004342 174546 MOV #64,#LPERR
2158 ;:
2159 ;: CLEAR DTR
2160 004342 LET #RCSR := #RCSR CLR.BY #DTR
2161 004342 042777 000002 174710 BIC #DTR,#RCSR
2162 ;: CLRSEND SHOULD FOLLOW
2163 004350 IF #CLRSEND SETIN #RCSR THEN
2164 004350 032777 020000 174702 BIT #CLRSEND,#RCSR
2165 004356 001401 BEQ #57
2166 ;: CLRSEND DID NOT
2167 004360 ERROR 50 ERRHAD 50,,FORCE
2168 004360 104050
2169 ;:
2170 ;: CLEAR WITH DTR
2171 004362 ENDF
2172 004362 #57:
2173 004362 ENDSUB
2174 ;:
2175 ;: SET
2176 004362 BGNSUB
2177 004362 012767 004370 174520 MOV #64,#LPERR
2178 ;:
2179 ;: SET DTR
2180 004370 LET #RCSR := #RCSR SET.BY #DTR
2181 004370 052777 000002 174662 BIS #DTR,#RCSR
2182 ;: CLRSEND SHOULD FOLLOW
2183 004376 IF #CLRSEND NOTSETIN #RCSR THEN
2184 004376 032777 020000 174654 BIT #CLRSEND,#RCSR
2185 004404 001001 BNE #60

```





MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 58  
 CVDVAD.P11 12-JUL-84 05:04 T17 TEST THAT CLRSEND SETS AND CLEARS

```

2219 ;*****
2220 ;*****
2221 ;*TEST 20 TEST THAT RING SETS AND CLEARS
2222 ;* AS REQSEND SETS AND CLEARS
2223 ;* THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
2224 ;*****
2225 004436 000004 TST20: SCOPE
2226 004440 012767 000010 174512 MOV #10,#TIMES ;DO 10 ITERATIONS
2227 004446 012767 000020 174524 MOV #20,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
2228 ; CAN WE USE THE WRAPAROUND??
2229 004454 ; IF #CABLE+FRFD NOTSETIN $USWR THEN
2230 004454 032767 060000 174536 BIT #CABLE+FRFD,$USWR
2231 004462 001004 BNE $62
2232 ; CAN'T TEST WITHOUT BERG OR H315
2233 ; OR WITH (-FR) JUMPER OUT.
2234 004464 EXIT TST
2235 004464 012767 000001 174466 MOV #1,#TIMES
2236 004472 000441 BR TST21 ;EXIT THIS TEST
2237 004474 ;
2238 004474 $62: ;
2239 ;
2240 ;
2241 ;
2242 ;
2243 ;
2244 ;
2245 ; CLEAR
2246 004474 ;
2247 004474 012767 004502 174406 MOV #64,$LPERR BGNSUB
2248 ;
2249 ;
2250 004502 ; CLEAR REQSEND
2251 004502 042777 000004 174550 BIC #REQSEND,@RCSR LET @RCSR := @RCSR CLR.BY #REQSEND
2252 ;
2253 004510 ; RING SHOULD FOLLOW
2254 004510 032777 040000 174542 BIT #RING,@RCSR IF @RING SETIN @RCSR THEN
2255 004516 001401 BEQ $63
2256 ;
2257 004520 ; RING DID NOT
2258 004520 104053 ERROR 53 ERRMRD 53..FORCE
2259 ;
2260 ; CLEAR WITH REQSEND
2261 004522 ;
2262 004522 $63: ;
2263 004522 ENDSUB
2264 ;
2265 ; SET
2266 004522 ;
2267 004522 012767 004530 174360 MOV #64,$LPERR BGNSUB
2268 ;
2269 ;
2270 004530 ; SET REQSEND
2271 004530 052777 000004 174522 BIS #REQSEND,@RCSR LET @RCSR := @RCSR SET.BY #REQSEND
2272 ;
2273 004536 ; RING SHOULD FOLLOW
2274 004536 032777 040000 174514 BIT #RING,@RCSR IF @RING NOTSETIN @RCSR THEN

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 60  
 CVDVAD.P11 12-JUL-84 05:04 T20 TEST THAT RING SETS AND CLEARS

```

2309 ;*****
2310 ;*****
2311 ;*TEST 21 TEST THAT SECRC SETS AND CLEARS
2312 ;* AS SECXMIT SETS AND CLEARS
2313 ;*****
2314 004576 000004 TST21: SCOPE
2315 004600 012767 000010 174352 MOV #10,#TIMES ;;DO 10 ITERATIONS
2316 004606 012767 000021 174364 MOV #21,#TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2317 ; CAN WE USE THE WRAPAROUND??
2318 004614 IF #CABLE,#USWR #CABLE NOTSETIN #USWR THEN
2319 004614 032767 020000 174376 BIT #CABLE,#USWR
2320 004622 001004 BNE #66
2321 ; CAN'T TEST WITHOUT BERG OR H315.
2322 004624 EXIT TST
2323 004624 012767 000001 174326 MOV #1,#TIMES
2324 004632 000441 BR TST22 ;;EXIT THIS TEST
2325 004634 ENDIF
2326 004634 $66:
2327
2328 ; SECXMIT AND
2329 ; SECRC ARE CONNECTED
2330
2331 ; BY THE H315 OR EQUIV.
2332
2333 ; CLEAR
2334 004634 BGNSUB
2335 004634 012767 004642 174246 MOV #64#,#LPERR
2336
2337 ; CLEAR SECXMIT
2338 004642 LET @RCSR := @RCSR CLR.BY #SECXMIT
2339 004642 042777 000010 174410 BIC #SECXMIT,@RCSR
2340
2341 ; SECRC SHOULD FOLLOW
2342 004650 IF #SECRC SETIN @RCSR THEN
2343 004656 032777 002000 174402 BIT #SECRC,@RCSR
2344 001401 BEQ #67
2345 ; SECRC DID NOT
2346 004660 104056 ERROR 56 ERRHRD 56..FORCE
2347
2348 ; CLEAR WITH SECXMIT
2349 004662 ENDIF
2350 004662 $67:
2351 004662 ENDSUB
2352
2353 ; SET
2354 004662 BGNSUB
2355 004662 012767 004670 174220 MOV #64#,#LPERR
2356
2357 ; SET SECXMIT
2358 004670 LET @RCSR := @RCSR SET.BY #SECXMIT
2359 004670 052777 000010 174362 BIS #SECXMIT,@RCSR
2360
2361 ; SECRC SHOULD FOLLOW
2362 004676 IF #SECRC NOTSETIN @RCSR THEN
2363 004704 032777 002000 174354 BIT #SECRC,@RCSR
2364 001001 BNE #70
; SECRC DID NOT SET

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 62  
 CVDVAD.P11 12-JUL-84 05:04 T21 TEST THAT SECURE SETS AND CLEARS

```

2397 ;*****
2398 ;*****
2399 ;*TEST 22 TEST THAT DATAINT (RCSR-15) SETS
2400 ;* WHEN DTR CHANGES STATE
2401 ;* AND THAT DATAINT IS CLEARED AFTER READING RCSR
2402 ;* NOTE DTR IS TIED TO BOTH CARDET AND CLRSEND BY THE H315
2403 ;* THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
2404 ;*****
2405 004736 000004 TST22: SCOPE
2406 004740 012767 000010 174212 MOV #10,#TIMES ;;DO 10 ITERATIONS
2407 004746 012767 000022 174224 MOV #22,#TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2408 ; CAN WE USE THE WRAPAROUND??
2409 004754 IF #CABLE+FRFD,#USWR THEN
2410 004754 032767 060000 174236 BIT #CABLE+FRFD,#USWR
2411 004762 001004 BNE #72
2412 ; CAN'T TEST WITHOUT BERG OR H315
2413 ; OR WITH (-FD) JUMPER OUT.
2414 004764 EXIT TST
2415 004764 012767 000001 174166 MOV #1,#TIMES
2416 004772 000463 BR TST23 ;;EXIT THIS TEST
2417 004774 ENDIF
2418 004774 $72:
2419
2420 ;MAKE SURE NOTHING UNEXPECTED HAPPENS
2421 004774 012746 000340 MOV #PR7,-(SP) ;;PUT NEW PS ON STACK
2422 005000 012746 005006 MOV #64,-(SP) ;;PUT NEW PC ON STACK
2423 005004 000002 RTI ;;POP NEW PC AND PS
2424 005006 64:
2425
2426 ;READ TWICE - CLEARS
2427 005006 BGNSUB
2428 005006 012767 005014 174074 MOV #65,#LPERR
2429 ; CLEAR DTR
2430 005014 LET @RCSR := @RCSR CLR.BY @DTR
2431 005014 042777 000002 174236 BIC @DTR,@RCSR
2432 ;WAIT 1 MILLI-SEC FOR CABLE
2433 005022 WAITMS 1
2434 005022 010546 MOV R5,-(SP)
2435 005024 012745 000001 MOV #1,-(R5)
2436 005030 004767 004716 JSR PC,WAIT
2437 005034 012605 MOV (SP)+,R5
2438
2439 005036 LET ; READ RCSR - TO CLEAR DATAINT
2440 005036 017703 174216 MOV @RCSR,R3
2441 ; READ RCSR AGAIN
2442 005042 IF #DATAINT SETIN @RCSR THEN
2443 005042 032777 100000 174210 BIT #DATAINT,@RCSR
2444 005050 001401 BEQ #73
2445 ; READING RCSR DID NOT CLEAR DATAINT
2446 005052 ERRHRD 61,EDATAINT
2447 005052 104061 ERROR 61
2448 005054 ENDIF
2449 005054 $73:
2450
2451 005054 ENDSUB
2452

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 64  
 CVDVAD.P11 12-JUL-84 05:04 T22 TEST THAT DATAINT (RCSR-15) SETS

```

2499 ;*****
2500 ;*****
2501 ;*TEST 23      TEST THAT DATAINT SETS WHEN RING SETS
2502 ;*              AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
2503 ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
2504 ;*****
2505 005142 000004 TST23: SCOPE
2506 005144 012767 000010 174006      MOV      #10,#TIMES      ;;DO 10 ITERATIONS
2507 005152 012767 000023 174020      MOV      #23,#TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2508 ;*              ; CAN WE USE THE WRAPAROUND??
2509 005160 ;*              IF      #CABLE+FRFD NOTSETIN #USWR THEN
2510 005160 032767 060000 174032      BIT      #CABLE+FRFD,#USWR
2511 005166 001004 ;*              ; CAN'T TEST WITHOUT BERG OR H315
2512 ;*              ; OR WITH (-FR) JUMPER OUT.
2513 ;*              EXIT TST
2514 005170 ;*              ;;EXIT THIS TEST
2515 005170 012767 000001 173762      MOV      #1,#TIMES
2516 005176 000473 ;*              ENDF
2517 005200 ;*
2518 005200 ;*
2519 ;*
2520 ;*
2521 005200 012746 000340 ;*              ;NO INTERRUPTS
2522 005204 012746 005212 ;*              ;;PUT NEW PS ON STACK
2523 005210 000002 ;*              ;;PUT NEW PC ON STACK
2524 005212 ;*              ;;POP NEW PC AND PS
2525 ;*
2526 ;*              ;START OFF WITH EVERYTHING CLEAR
2527 005212 ;*              BGNSUB
2528 005212 012767 005220 173670      MOV      #65#,#LPERR
2529 ;*
2530 ;*
2531 005220 ;*              ;CLEAR RING
2532 005220 042777 000004 174032      BIC      #REQSEND,#RCSR      LET      @RCSR := @RCSR CLR.BY #REQSEND
2533 ;*
2534 005226 ;*              ;WAIT 1 MILLI-SEC FOR CABLE
2535 005226 010546 ;*              WAITMS 1
2536 005230 012745 000001 ;*
2537 005234 004767 004512 ;*
2538 005240 012605 ;*
2539 ;*
2540 005242 ;*              ;READ ONCE
2541 005242 017703 174012 ;*              LET      R3 := @RCSR
2542 ;*
2543 005246 ;*              ;READ TWICE
2544 005246 032777 100000 174004 ;*              IF      #DATAINT SETIN @RCSR THEN
2545 005254 001401 ;*              #DATAINT SETIN @RCSR THEN
2546 ;*
2547 005256 ;*              ;READING RCSR DID NOT CLEAR DATAINT
2548 005256 104065 ;*              ERRHRD 65, EDATAINT
2549 005260 ;*
2550 005260 ;*              ENDF
2551 005260 ;*              ENDSUB
2552 ;*
2553 ;*              ;
2554 005260 ;*              SET RING --> SET DATAINT
                ;*              BGNSUB

```





MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 66  
 CVDVAD.P11 12-JUL-84 05:04 T23 TEST THAT DATAINT SETS WHEN RING SETS

```

2605      ;:*****
2606      ;:*****
2607      ;*TEST 24      TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE
2608      ;:*****
2609      005366 000004      TST24: SCOPE
2610      005370 012767 000010 173562      MOV      @10,@TIMES      ;;DO 10 ITERATIONS
2611      005376 012767 000024 173574      MOV      @24,@TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2612      ;:*****
2613      005404      ; CAN WE USE THE WRAPAROUND??
2614      005404 032767 020000 173606      BIT      @CABLE,@USWR      IF      @CABLE NOTSETIN @USWR THEN
2615      005412 001004      BNE      @103
2616      ; CAN'T TEST WITHOUT BERG OR M315.
2617      005414      EXIT TST
2618      005414 012767 000001 173536      MOV      @1,@TIMES
2619      005422 000454      BR       TST25      ;;EXIT THIS TEST
2620      005424      ENDF
2621      005424      @103:
2622
2623      ;NO INTERRUPTS
2624      005424 012746 000340      MOV      @PR7,-(SP)      ;;PUT NEW PS ON STACK
2625      005430 012746 005436      MOV      @64,-(SP)      ;;PUT NEW PC ON STACK
2626      005434 000002      RTI      ;;POP NEW PC AND PS
2627      005436      64:
2628
2629
2630      ;START FRESH
2631      ;CLEAR SECREC
2632      005436      LET      @RCSR := @RCSR CLR.BY @SECXMIT
2633      005436 042777 000010 173614      BIC      @SECXMIT,@RCSR
2634      005444      LET      R3 := @RCSR
2635      005444 017703 173610      MOV      @RCSR,R3
2636
2637      ;SET SECREC --> DATAINT SET
2638      005450      BGNSUB
2639      005450 012767 005456 173432      MOV      @65,@LPERR
2640
2641      ;SET SECREC
2642      005456      LET      @RCSR := @RCSR SET.BY @SECXMIT
2643      005456 052777 000010 173574      BIS      @SECXMIT,@RCSR
2644      ;WAIT 1 MILLI-SEC FOR CABLE
2645      005464      WAITMS 1
2646      005464 010546      MOV      R5,-(SP)
2647      005466 012745 000001      MOV      @1,-(R5)
2648      005472 004767 004254      JSR      PC,WAIT
2649      005476 012605      MOV      (SP)+,R5
2650
2651      005500      IF      @DATAINT NOTSETIN @RCSR THEN
2652      005500 032777 100000 173552      BIT      @DATAINT,@RCSR
2653      005506 001001      BNE      @104
2654      ;SETTING SECREC DID NOT SET DATAINT
2655      005510      ERRHRD 124,, E2DATA
2656      005510 104124      ERROR 124
2657      005512      ENDF
2658      005512      @104:
2659
2660      ENDSUB
;CLEAR SECREC --> DATAINT SET

```

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 67  
 CVDVAD.P11      12-JUL-84 05:04      T24      TEST THAT DATAINT SETS WHEN SECURE CHANGES STATE

2661	005512							BGNSUB
2662	005512	012767	005520	173370	MOV	#64#, #LPERR		
2663								;CLEAR SECURE
2664	005520							LET   @RCSR := @RCSR CLR.BY #SECXMIT
2665	005520	042777	000010	173532	BIC	#SECXMIT, @RCSR		;WAIT 1 MILLI-SEC FOR CABLE
2666								WAITMS 1
2667	005526							
2668	005526	010546			MOV	R5, -(SP)		
2669	005530	012745	000001		MOV	#1, -(R5)		
2670	005534	004767	004212		JSR	PC, WAIT		
2671	005540	012605			MOV	(SP)+, R5		
2672	005542							IF    #DATAINT NOTSETIN @RCSR THEN
2673	005542	032777	100000	173510	BIT	#DATAINT, @RCSR		
2674	005550	001001			BNE	#105		;CLEARING SECURE DID NOT SET DATAINT
2675								ERRHRD 125,, E2DATA
2676	005552							
2677	005552	104125			ERROR	125		ENDIF
2678	005554							
2679	005554							
2680	005554							ENDSUB
2681	005554							ENDTST
2682								
2683								
2684								

#105:

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 68  
CVDVAD.P11 12-JUL-84 05:04 T24

TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE

```

2685
2686
2687
2688
2689
2690
2691 005554 000004
2692 005556 012767 000001 173374
2693 005564 012767 000025 173406
2694
2695
2696 005572
2697 005572 032767 000001 173414
2698 005600 001404
2699 005602
2700 005602 012767 000001 173350
2701 005610 000454
2702 005612
2703 005612
2704 005612
2705 005612 012767 005620 173270
2706
2707
2708
2709
2710
2711
2712 005620
2713 005620 105077 173444
2714
2715
2716
2717 005624
2718 005624 010546
2719 005626 012745 177777
2720 005632 016745 173426
2721 005636 012745 000200
2722 005642 012745 001000
2723 005646 004767 003622
2724 005652 012605
2725
2726
2727 005654
2728 005654 103001
2729
2730 005656
2731 005656 104066
2732 005660
2733 005660
2734 005660
2735
2736 005660
2737 005660 012767 005666 173222
2738
2739
2740

```

```

;*****
;*****
;TEST 25      TEST THAT XMIT RDY - TCSR 7 - CLEARS
;             WHEN TBUF IS LOADED WITH A CHARACTER
;             AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
;*****
TST25: SCOPE
        MOV     #1,#TIMES      ;DO 1 ITERATION
        MOV     #25,#TESTN    ;SET TEST NUMBER IN APT MAIL BOX

; THIS TEST IS 'BREAK OR HALT' SENSITIVE.
; IF #APTENV SET IN #ENV THEN
        BIT     #APTENV,#ENV
        BEQ     #106

                EXIT TEST
        MOV     #1,#TIMES
        BR      TST26          ;;;EXIT THIS TEST
                                ENDF

;*****
;*****
                                BGNSUB
        MOV     #64,#LPERR
; LOAD TBUF WITH ONE CHARACTER
; WAIT FOR READY TO SET
; (SHOULD BE VERY SHORT WAIT
; SINCE UART DOUBLE BUFFERS ITS INPUT)

                                ;SEND A CHARACTER
                                LET #TBUF :B= #0
                                ;WAIT A MAXIMUM
                                ;OF 50 MSEC FOR
                                ;XMIT RDY TO SET IN TCSR
                                CALL TIMER IN <#1000,#XMITRDY,TCSR,#SET>

        MOV     R5,-(SP)
        MOV     #SET,-(R5)
        MOV     TCSR,-(R5)
        MOV     #XMITRDY,-(R5)
        MOV     #1000,-(R5)
        JSR     PC,TIMER
        MOV     (SP),R5

                                ;TIMER RETURNS AN ERROR IF BIT DID
                                ;NOT MEET CONDITION WITHIN TIME LIMIT
                                IF.ERROR THEN

; XMIT RDY DID NOT SET IN TCSR
ERRHRD 66,,DIDNOT

                                ENDF
                                ENDSUB
                                BGNSUB
        MOV     #64,#LPERR
; LOAD TBUF WITH A SECOND CHARACTER
; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
; AND THEN WAIT FOR IT TO SET

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 69  
 CVDVAD.P11 12-JUL-84 05:04 T25 TEST THAT XMIT RDY - TCSR 7 - CLEARS

```

2741
2742
2743 005666
2744 005666 105077 173376
2745 005672 000240
2746
2747
2748 005674
2749 005674 032777 000200 173362
2750 005702 001401
2751
2752 005704
2753 005704 104067
2754 005706
2755 005706
2756
2757
2758
2759
2760 005706
2761 005706 010546
2762 005710 012745 177777
2763 005714 016745 173344
2764 005720 012745 000200
2765 005724 012745 001000
2766 005730 004767 003540
2767 005734 012605
2768 005736
2769 005736 103001
2770
2771 005740
2772 005740 104070
2773 005742
2774 005742
2775 005742
2776 005742

;SEND SECOND CHARACTER
LET @TBUF :B= #0
; GIVE IT TIME TO CLEAR
; XMITRDY SHOULD HAVE CLEARED UPON
; RECEIPT OF A CHARACTER
IF @XMITRDY SET IN @TCSR THEN
; XMITRDY DID NOT CLEAR IN TCSR
ERRHRD 67,,DIDNOT
ENDIF

;WAIT A MAXIMUM
;OF 50 MSEC FOR
;XMIT RDY TO SET IN TCSR
CALL TIMER IN <@1000,@XMITRDY,TCSR,@SET>

MOV R5,-(SP)
MOV @SET,-(R5)
MOV TCSR,-(R5)
MOV @XMITRDY,-(R5)
MOV @1000,-(R5)
JSR PC,TIMER
MOV (SP),R5

IF .ERROR THEN
;XMIT RDY DID NOT SET IN TCSR
ERRHRD 70,,DIDNOT
ENDIF

$110:
$111:

ENDSUB
ENDTST

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 70  
 CVDVAD.P11 12-JUL-84 05:04 T25 TEST THAT XMIT RDY - TCSR 7 - CLEARS

```

2777 ;:*****
2778 ;:*****
2779 ;*TEST 26 TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
2780 ;* RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
2781 ;* AND THAT RESET CLEARS THE BIT.
2782 ;:*****
2783 005742 000004 TST26: SCOPE
2784 005744 012767 000010 173206 MOV #10,#TIMES ;:DO 10 ITERATIONS
2785 005752 012767 000026 173220 MOV #26,#TESTN ;:SET TEST NUMBER IN APT MAIL BOX
2786
2787 ; SET THE MAINTENANCE BIT
2788 005760 LET @TCSR :* @TCSR SET.BY @MAINT
2789 005760 052777 000004 173276 BIS @MAINT,@TCSR
2790
2791 005766 BGNSUB
2792 005766 012767 005774 173114 MOV #64,#LPERR
2793 ; SEND A CHARACTER AND LET IT WRAP AROUND
2794
2795 005774 LET @TBUF :B= #0
2796 005774 105077 173270 CLRB @TBUF
2797
2798 ; WAIT A MAXIMUM OF 50 MSEC
2799 ; FOR RCVR DONE TO SET IN
2800 ; RCSR
2801 006000 CALL TIMER IN <#1000,@RCVRDONE,RCSR,#SET>
2802 006000 010546 MOV R5,-(SP)
2803 006002 012745 177777 MOV #SET,-(R5)
2804 006006 016745 173246 MOV RCSR,-(R5)
2805 006012 012745 000200 MOV @RCVRDONE,-(R5)
2806 006016 012745 001000 MOV #1000,-(R5)
2807 006022 004767 003446 JSR PC,TIMER
2808 006026 012605 MOV (SP)+,R5
2809
2810 ;DIDN'T SET IN TIME
2811 006030 IF.ERROR THEN
2812 006030 103001 BCC #112
2813
2814 ; RCVRDONE DID NOT SET IN RCSR
2815 006032 104071 ERROR 71
2816 006034
2817 006034 ;112:
2818
2819 006034 ENDSUB
2820
2821 BGNSUB
2822 006034 012767 006042 173046 MOV #64,#LPERR
2823 ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
2824 ; THIS ALSO WILL CLEAR THE MAINT. BIT
2825 006042 BRESET
2826 006042 000005 RESET
2827
2828 006044 IF @RCVRDONE SETIN @RCSR THEN

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 71  
CVDVAD.P11 12-JUL-84 05:04 T26 TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)

2829 006044 032777 000200 173206  
2830 006052 001401  
2831  
2832 006054  
2833 006054 104072  
2834 006056  
2835 006056  
2836 006056  
2837 006056

BIT @RCVROONE,@RCSR  
BEQ #113

; RCVROONE DID NOT RESET IN RCSR.  
ERRHRD 72,,DIDNOT

ERROR 72

ENDIF

#113:

ENDSUB  
ENDTST

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 72  
 CVDVAD.P11 12-JUL-84 05:04 T26 TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)

```

2838                                     ;*****
2839                                     ;*****
2840                                     ;*TEST 27      TEST THAT RCVRDONE IS CLEARED BY READING RBUF
2841                                     ;*****
2842 006056 000004 TST27: SCOPE
2843 006060 012767 000010 173072      MOV    #10,#TIMES      ;;DO 10 ITERATIONS
2844 006066 012767 000027 173104      MOV    #27,#TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2845
2846                                     ; SET MAINT. BIT
2847 006074                                     LET @TCSR := @TCSR SET.BY #MAINT
2848 006074 052777 000004 173162      BIS    #MAINT,@TCSR
2849 006102                                     BGNSUB
2850 006102 012767 006110 173000      MOV    #64#,#LPERR
2851                                     ; OUTPUT A CHARACTER WITH MAINTENANCE
2852                                     ; SET, AND WAIT FOR XMITRDY TO SET.
2853
2854                                     ; OUTPUT A CHARACTER
2855 006110                                     LET @TBUF :B= #0
2856 006110 105077 173154      CLRB  @TBUF
2857                                     ; WAIT MAXIMUM OF 1000 MSEC
2858                                     ; FOR RCVRDONE TO SET IN
2859                                     ; RCSR
2860                                     CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
2861 006114
2862 006114 010546      MOV    R5,-(SP)
2863 006116 012745 177777      MOV    #SET,-(R5)
2864 006122 016745 173132      MOV    RCSR,-(R5)
2865 006126 012745 000200      MOV    #RCVRDONE,-(R5)
2866 006132 012745 001000      MOV    #1000,-(R5)
2867 006136 004767 003332      JSR   PC,TIMER
2868 006142 012605      MOV    (SP)+,R5
2869
2870                                     ; DID IT BECAME READY?
2871                                     IF.ERROR THEN
2872                                     ;RCVRDONE DID NOT SET IN RCSR
2873                                     ERRHRD 73.. DIDNOT
2874                                     ENDIF
2875 006150                                     $114:
2876 006150                                     ENDSUB
2877
2878                                     ; NOW THAT IT IS SET LETS SEE IF READING THE
2879                                     ; BUFFER CLEARS RCVRDONE.
2880
2881                                     ;READ BUFFER
2882 006150                                     LET RO :B= @RBUF
2883 006150 117700 173106      MOVB  @RBUF,RO
2884
2885                                     IF #RCVRDONE SETIN @RCSR THEN
2886 006154 032777 000200 173076      BIT   #RCVRDONE,@RCSR
2887 006162 001401      BEQ   #115
2888
2889                                     ;RCVRDONE DID NOT CLEAR IN RCSR
2890                                     ERRHRD 74,DIDNOT
2891                                     ENDIF
2892 006166                                     $115:
2893 006166                                     ENDTST

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 73  
 CVDVAD.P11 12-JUL-84 05:04 T27 TEST THAT RCVRDONE IS CLEARED BY READING RBUF

```

2894 ;*****
2895 ;*****
2896 ;*TEST 30      TEST THAT RCVRACT - RCSR 11 - SETS
2897 ;*            WHEN A START BIT IS RECEIVED AND
2898 ;*            CLEARS WHEN RCVRDONE - RCSR 7 - SETS
2899 ;*****
2900 TST30: SCOPE
2901     MOV     #10,#TIMES      ;;DO 10 ITERATIONS
2902     MOV     #30,#TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2903
2904 ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
2905 ; IF #APTENV SETIN #ENV THEN
2906     BIT     #APTENV,#ENV
2907     BEQ     #116
2908
2909 ; EXIT TEST
2910     MOV     #1,#TIMES
2911     BR      TST31          ;;EXIT THIS TEST
2912
2913 ; ENDIF
2914
2915 ; LET #TCSR := #TCSR SET.BY #MAINT
2916     BIS     #MAINT,#TCSR
2917
2918 ; LET R0 := #CLR
2919     MOV     #CLR,R0
2920
2921 ; LET R1 := #0
2922     CLR     R1
2923
2924 ;LOAD A CHARACTER INTO TBUF
2925 ;WAIT FOR RCVRACT TO SET
2926
2927 ;SEND A CHARACTER
2928     LET #TBUF :B= #0
2929
2930 ; REPEAT
2931
2932 ; IF #RCVRACT SETIN #RCSR THEN
2933
2934     LET R0 := #SET
2935
2936 ; ELSE
2937
2938     LET R1 := R1 + #1
2939
2940 ; ENDF
2941
2942 ; UNTIL R0 EQ #SET OR R1 HI MAX
2943
2944     CMP     R0,#SET
2945     BEQ     #122
2946     CMP     R1,MAX
2947     BLOS   #117
2948
2949 ; IF R1 HI MAX THEN
2950
2951 ; IT NEVER SET
2952 ; RCVRACT DID NOT SET IN RCSR.

```





MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 75  
 CVDVAD.P11 12-JUL-84 05:04 T30 TEST THAT RCVRCT - RCSR 11 - SETS

```

3006
3007 006402
3008 006402 017700 172654      MOV      @RBUF,R0      ;READ CHAR,
                                     LET R0 := @RBUF
3009
3010 006406
3011 006406 032777 000200 172644  BIT      @RCVRDONE,@RCSR  IF @RCVRDONE SETIN @RCSR THEN
3012 006414 001401      BEQ      #131
3013
3014 006416
3015 006416 104100      ERROR  100      ;RCVRDONE DID NOT CLEAR IN RCSR
                                     ERRHRD 100,,DIDNOT
3016 006420
3017 006420      #131:      ENDIF
3018
3019 006420
3020 006420 000401      BR      TST31      EXIT
3021 006422 070000      MAX:70000      :::EXIT THIS TEST
3022
3023 006424
3024
                                     ENDTST

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 76  
 CVDVAD.P11 12-JUL-84 05:04 T30 TEST THAT RCVRCT - RCSR 11 - SETS

```

3025
3026
3027
3028
3029
3030 006424 000004
3031 006426 012767 000010 172524
3032 006434 012767 000031 172536
3033
3034 006442
3035 006442 012767 006450 172440
3036
3037
3038
3039
3040
3041 006450
3042 006450 105077 172614
3043
3044 006454
3045 006454 010546
3046 006456 012745 002000
3047 006462 004767 003264
3048 006466 012605
3049
3050
3051 006470
3052 006470 105077 172574
3053
3054 006474
3055 006474 010546
3056 006476 012745 002000
3057 006502 004767 003244
3058 006506 012605
3059
3060
3061 006510
3062 006510 017704 172546
3063
3064
3065 006514
3066 006514 032704 040000
3067 006520 001005
3068
3069 006522
3070 006522 104101
3071
3072
3073 006524
3074 006524 012767 000001 172426
3075 006532 000456
3076 006534
3077 006534
3078 006534
3079
3080

```

```

;*****
;*****
;TEST 31 TEST THE OVERRUN BIT - RBUF 14
;*****
TST31: SCOPE
MOV #10,#TIMES ;DO 10 ITERATIONS
MOV #31,#TESTN ;SET TEST NUMBER IN APT MAIL BOX

BGNSUB
MOV #64,#LPERR
;OUTPUT 2 CHARACTERS WITH
;AMPLE DELAYS BETWEEN FOR RECEPTION.
;THIS SHOULD AN CAUSE OVERRUN ERROR.

;OUTPUT 1 CHARACTER
LET @TBUF :B= #0

;GO AWAY FOR 1024. 14 SEC
WAITMS 1024.

MOV R5,-(SP)
MOV #1024,-(R5)
JSR PC,WAIT
MOV (SP)+,R5

;OUTPUT 2ND CHARACTER
LET @TBUF :B= #0

;LET OVERRUN HAPPEN
WAITMS 1024.

MOV R5,-(SP)
MOV #1024,-(R5)
JSR PC,WAIT
MOV (SP)+,R5

;READ BUFFER AND ERROR BITS
LET R4 := @RBUF

;IT DIDN'T SET
IF #ORERR NOTSET IN R4 THEN

;ORERR DID NOT SET IN RBUF
ERRHRD 101,,DIDNOT

;NO USE COMPOUNDING ERRORS
EXIT TST

MOV #1,#TIMES
BR TST32 ;;;EXIT THIS TEST
ENDIF

$132:
ENDSUB

;NOW SEE IF ERROR BIT SET WITH OVERRUN ERROR:

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 78  
 CVDVAD.P11 12-JUL-84 05:04 T31 TEST THE OVERRUN BIT - RBUF 14

```

3137 006642 104104          ERROR 104
3138
3139
3140                                ; -AFTER RECEIVING ANOTHER CHAR
3141                                ; SKIP AROUND REST
3141 006644                                EXIT TST
3142 006644 012767 000001 172306      MOV    #1, #TIMES
3143 006652 000406                                BR    TST32          ;;:EXIT THIS TEST
3144 006654                                ENDIF
3145 006654                                #135:
3146
3147 006654                                IF #ERROR SETIN @RBUF THEN
3148 006654 032777 100000 172400      BIT    #ERROR, @RBUF
3149 006662 001401                                BEQ    #136
3150
3151                                ; ERROR DID NOT CLEAR IN RBUF
3151 006664                                ERRHRD 105, .DIDNOT
3152 006664 104105          ERROR 105
3153
3154                                ENDIF
3155 006666                                #136:
3156 006666
3157 006666                                ENDSUB
3158 006666 000400          BR    TST32          ;;:EXIT THIS TEST
3159                                .EVEN
3160 006670                                ENDTST
3161

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 79  
 CVDVAD.P11 12-JUL-84 05:04 T31 TEST THE OVERRUN BIT - RBUF 14

```

3162
3163
3164
3165
3166
3167
3168
3169
3170 006670 000004
3171 006672 012767 000010 172260
3172 006700 012767 000032 172272
3173 006706
3174 006706 032767 000200 172304
3175 006714 001004
3176 006716
3177 006716 012767 000001 172234
3178 006724 000552
3179 006726
3180 006726
3181
3182 006726
3183 006726 032767 000001 172260
3184 006734 001404
3185 006736
3186 006736 012767 000001 172214
3187 006744 000542
3188 006746
3189 006746
3190
3191 006746
3192 006746 012767 177777 000272
3193 006754
3194 006754 012767 177777 000266
3195 006762
3196 006762 052777 000004 172274
3197
3198 006770
3199 006770 005003
3200 006772 000401
3201 006774
3202 006774 005203
3203 006776
3204 006776 020327 000017
3205 007002 003060
3206 007004
3207 007004 017700 172252
3208
3209 007010
3210 007010 116377 007170 172250
3211
3212 007016
3213 007016 005002
3214
3215 007020
3216 007020 005077 172244
3217

```

```

;*****
;*****
;*TEST 32 PROGRAMMABLE BAUD RATE TEST
;* TEST AT ALL SPEEDS AVAILABLE
;* A COMPARISON WILL BE MADE TO SEE
;* IF NEW TIME IS LESS THAN PREVIOUS.
;*****
TST32: SCOPE
MOV @10,$TIMES ;;DO 10 ITERATIONS
MOV @32,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
IF @PBR NOTSETIN $USMR THEN
EXIT TST
MOV @1,$TIMES
BR TST33 ;;EXIT THIS TEST
ENDIF
$137:
; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
IF @APTENV SETIN $ENV THEN
BIT @APTENV,$ENV
BEQ $140
EXIT TEST
MOV @1,$TIMES
BR TST33 ;;EXIT THIS TEST
ENDIF
$140:
LET OLD := @-1
LET OLD+2 := @-1
LET @TCSR := @TCSR SET.BY @MAINT
;EACH BAUD RATE
INCR R3 FROM @0 TO @15. BY @1
$142:
INC R3
$141:
CMP R3,@15.
BGT $143
LET R0 := @RBUF
;CHANGE BAUDE RATE
LET @TCSRHI := @RATES(R3)
;FLAG
LET BIT := @0
;OUTPUT THE CHARACTER
LET @TBUF := @0
;INITIALIZE COUNTER

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 80  
 CVDVAD.P11 12-JUL-84 05:04 T32 PROGRAMMABLE BAUD RATE TEST

3218	007024						LET NEW := #0
3219	007024	005067	000212		CLR	NEW	LET NEW+2 := #0
3220	007030						WHILE BIT EQ #0 DO
3221	007030	005067	000210		CLR	NEW+2	
3222	007034						
3223	007034			\$144:			
3224	007034	005702			TST	BIT	
3225	007036	001014			BNE	\$145	
3226	007040						IF #RCVRDONE SETIN @RCSR THEN
3227	007040	032777	000200	172212	BIT	@RCVRDONE,@RCSR	
3228	007046	001403			BEQ	\$146	
3229							;DONE - ITS READY
3230	007050						LET BIT := #1
3231	007050	012702	000001		MOV	#1,BIT	
3232	007054						ELSE
3233	007054	000404			BR	\$147	
3234	007056			\$146:			
3235							;OTHERWISE-INCREMENT TIME
3236	007056						LET NEW := NEW + #1
3237	007056	005267	000160		INC	NEW	
3238	007062						LET NEW+2 := NEW+2 + CARRY
3239	007062	005567	000156		ADC	NEW+2	
3240	007066						ENDIF
3241	007066			\$147:			
3242							;SIGNALS DONE
3243	007066						ENDDO
3244	007066	000762			BR	\$144	
3245	007070			\$145:			
3246							
3247	007070						IF NEW+2 LO OLD+2 THEN
3248	007070	026767	000150	000152	CMP	NEW+2,OLD+2	
3249	007076	103001			BHIS	\$150	
3250							; OK
3251	007100						ELSE
3252	007100	000412			BR	\$151	
3253	007102			\$150:			
3254							; NEW+2 >= OLD+2
3255	007102						IF NEW+2 EQ OLD+2 AND NEW LO OLD THEN
3256	007102	026767	000136	000140	CMP	NEW+2,OLD+2	
3257	007110	001005			BNE	\$152	
3258	007112	026767	000124	000126	CMP	NEW,OLD	
3259	007120	103001			BHIS	\$152	
3260							;OK
3261	007122						ELSE
3262	007122	000401			BR	\$153	
3263	007124			\$152:			
3264							;NEW+2 > OLD+2 OR
3265							;(NEW+2 = OLD+2 AND
3266							; NEW >= OLD)
3267							;BAUD RATE DIDN'T CHANGE
3268	007124						ERRHRD 126, BAUDRATE
3269	007124	104126			ERROR	126	
3270	007126						ENDIF
3271	007126			\$153:			
3272	007126						ENDIF
3273	007126			\$151:			

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 81  
 CVDVAD.P11 12-JUL-84 05:04 T32 PROGRAMMABLE BAUD RATE TEST

```

3274                                     ;UPDATE OLD TIME
3275 007126                                     LET OLD := NEW
3276 007126 016767 000110 000112      MOV     NEW,OLD
3277 007134                                     LET OLD*2 := NEW*2
3278 007134 016767 000104 000106      MOV     NEW*2,OLD*2
3279
3280 007142                                     ENDINC  ;BAUD RATE
3281 007142 000714      BR      #142
3282 007144      #143:
3283 007144                                     LET R3 :B= #USMR*1 AND #17      ; PUT BAUD BACK
3284 007144 116703 172051      MOVB   #USMR*1,R3
3285 007150 110346      MOVB   R3,-(SP)
3286 007152 142716 000017      BICB   #17,(SP)
3287 007156 142603      BICB   (SP),R3
3288 007160                                     LET #TCSRHI :B= RATES(R3)      ; LIKE HE WANTED IT
3289 007160 116377 007170 172100      MOVB   RATES(R3),#TCSRHI
3290
3291 007166                                     EXIT   ;SKIP TABLE
3292 007166 000431      BR      TST33      ;;;EXIT THIS TEST
3293
3294 007170
    
```

RATES: ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE  
 ;UPPER BYTE OF XCSR FOR EACH BAUD RATE  
 ;\*\* NOTE:: THE VALUE INDICATED IN THE COLUMN 'OFFSET  
 ;\*\* INTO TABLE' CAN BE PLACED INTO BITS<11:8>  
 ;\*\* OF LOCATION '#USMR' TO CAUSE THE CORROSPONDING  
 ;\*\* BAUD TO BE SELECTED IN THE DLV1.1-E UPON  
 ;\*\* COMPLETION OF THIS TEST.

					BAUD	OFFSET INTO TABLE
3303	007170	010	R0050:	.BYTE	010	0
3304	007171	030	R0070:	.BYTE	030	1
3305	007172	050	R0110:	.BYTE	050	2
3306	007173	070	R0135:	.BYTE	070	3
3307	007174	110	R0150:	.BYTE	110	4
3308	007175	130	R0300:	.BYTE	130	5
3309	007176	150	R0600:	.BYTE	150	6
3310	007177	170	R0200:	.BYTE	170	7
3311	007200	210	R1800:	.BYTE	210	10
3312	007201	230	R2000:	.BYTE	230	11
3313	007202	250	R2400:	.BYTE	250	12
3314	007203	270	R3600:	.BYTE	270	13
3315	007204	310	R4800:	.BYTE	310	14
3316	007205	330	R7200:	.BYTE	330	15
3317	007206	350	R9600:	.BYTE	350	16
3318	007207	370	R10000:	.BYTE	370	17
3319						



MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 82  
 CVDVAD.P11      12-JUL-84 05:04      T32      PROGRAMMABLE BAUD RATE TEST

3320	007210	040502	042125	051040	BAUDRATE:	.ASCIZ /BAUD RATE DIDN'T CHANGE./
3321	007216	052101	020105	044504		
3322	007224	047104	052047	041440		
3323	007232	040510	043516	027105		
3324	007240	000				
3325		007242			.EVEN	
3326	007242	000000	000000		NEW: 0.0	
3327	007246	000000	000000		OLD: 0.0	
3328	007252					ENDTST
3329						
3330						
3331						

```

3332 ;*****
3333 ;*****
3334 ;*TEST 33 TRANSMITTER INTERRUPT LOGIC TEST
3335 ;* LOGICALLY THIS IS 4 SEPARATE TESTS
3336 ;* A) DOES TRANSMITTER INTERRUPT LOGIC WORK
3337 ;* B) AT PRIORITY OF 0
3338 ;* C) AND ONLY ONCE
3339 ;* D) BUT NOT WITH INTERRUPT ENABLE CLEAR
3340 ;*****
3341 007252 000004 TST33: SCOPE
3342 007254 012767 000010 171676 MOV #10,#TIMES ;DO 10 ITERATIONS
3343 007262 012767 000033 171710 MOV #33,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
3344 ;CLEAR 'INTERRUPT OCCURED' FLAG
3345 007270 LET INTFLAG := #0
3346 007270 005067 002544 CLR INTFLAG
3347
3348 ;GET VECTOR ADDRESS
3349 007274 LET R3 := DLVEC
3350 007274 016703 171756 MOV DLVEC,R3
3351 ;FOR THE TRANSMITTER
3352 007300 LET R3 := R3 * #4
3353 007300 062703 000004 ADD #4,R3
3354 ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
3355 007304 SETVEC R3, #INTSRV, #PR7
3356 007304 010146 MOV R1,-(SP)
3357 007306 010301 MOV R3,R1
3358 007310 012721 012032 MOV #INTSRV,(R1)+
3359 007314 012711 000340 MOV #PR7,(R1)
3360 007320 012601 MOV (SP)+,R1
3361 007322 BGNSUB
3362 007322 012767 007330 171560 MOV #64#,#LPERR
3363 ;CLEAR INTERRUPT ENABLE
3364 007330 LET @TCSR := @TCSR CLR.BY #XMITIE
3365 007330 042777 000100 171726 BIC #XMITIE,@TCSR
3366
3367 ;SET IT TO 0
3368 007336 012746 000000 MOV #PRO,-(SP) ;PUT NEW PS ON STACK
3369 007342 012746 007350 MOV #65#,-(SP) ;PUT NEW PC ON STACK
3370 007346 000002 RTI ;POP NEW PC AND PS
3371 007350 65#:
3372
3373 ;NOW SET I.E. BIT
3374 007350 LET @TCSR := @TCSR SET.BY #XMITIE
3375 007350 052777 000100 171706 BIS #XMITIE,@TCSR
3376
3377 ;LET INTERRUPT HAVE TIME TO OCCUR
3378 007356 WAITMS 1024.
3379 007356 010546 MOV R5,-(SP)
3380 007360 012745 002000 MOV #1024..-(R5)
3381 007364 004767 002362 JSR PC,WAIT
3382 007370 012605 MOV (SP)+,R5
3383
3384 ;DID EXACTLY 1 INTERRUPT OCCUR
3385 007372 IF INTFLAG NE #1 THEN
3386 007372 026727 002442 000001 CMP INTFLAG,#1
3387 007400 001406 BEQ #154

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 84  
 CVDVAD.P11 12-JUL-84 05:04 T33 TRANSMITTER INTERRUPT LOGIC TEST

```

3388
3389 007402
3390 007402 005767 002432          TST   INTFLAG
3391 007406 001002                    BNE   #155
3392
3393 007410
3394 007410 104106          ERROR   106
3395 007412
3396 007412 000401          BR     #156
3397 007414          #155:
3398
3399
3400 007414
3401 007414 104107          ERROR   107
3402 007416
3403 007416          #156:
3404 007416
3405 007416          #154:
3406 007416
3407
3408 007416
3409 007416 012767 007424 171464      MOV   #64#,#LPERR
3410
3411 007424
3412 007424 005067 002410          CLR   INTFLAG
3413
3414 007430
3415 007430 042777 000100 171626      BIC   #XMITIE,#TCSR
3416
3417 007436 012746 000000          MOV   #PRO,-(SP)      ;;PUT NEW PS ON STACK
3418 007442 012746 007450          MOV   #65#,-(SP)      ;;PUT NEW PC ON STACK
3419 007446 000002          RTI                    ;;POP NEW PC AND PS
3420 007450          65#:
3421
3422 007450
3423 007450 010546          MOV   R5,-(SP)
3424 007452 012745 000002          MOV   #2,-(R5)
3425 007456 004767 002270          JSR   PC,WAIT
3426 007462 012605          MOV   (SP)+,R5
3427 007464
3428 007464 005767 002350          TST   INTFLAG
3429 007470 001401          BEQ   #157
3430
3431 007472
3432 007472 104110          ERROR   110
3433 007474
3434 007474          #157:
3435 007474
3436 007474 000005          RESET
3437 007476
3438
3439 007476
3440 007476 017704 171554          MOV   #DLVEC,R4
3441 007502
3442 007502 010146          MOV   R1,-(SP)      ;;PUSH R1 ON STACK
3443 007504 010246          MOV   R2,-(SP)      ;;PUSH R2 ON STACK

```

;NO - WAS IT 0 OR MORE THAN ONCE  
 IF INTFLAG EQ #0 THEN

;TRANSMITTER DID NOT INTERRUPT IN TIME  
 ERRHRD 106,,DIDNOT

ELSE

;TWICE  
 ;TRANSMITTER INTERRUPTED TWICE  
 ERRHRD 107,,TWICE

ENDIF

ENDIF

ENDSUB

;INTERRUPT WITHOUT INTERRUPT ENABLE SET

BGNSUB

;CLEAR 'INTERRUPT OCCURED' FLAG  
 LET INTFLAG := #0

;CLEAR INTERRUPT ENABLE  
 LET #TCSR := #TCSR CLR.BY #XMITIE

;NO INTERRUPTS SHOULD OCCUR.

;;PUT NEW PS ON STACK  
 ;;PUT NEW PC ON STACK  
 ;;POP NEW PC AND PS

;DARE IT TO HAPPEN  
 WAITMS 2

IF INTFLAG NE #0 THEN

;INTERRUPT OCCURED WITH I E CLEARED  
 ERRHRD 110,NOTENAB

ENDIF

BRESET

ENDSUB

;RESTORE VECTOR AREA  
 LET R4 := #DLVEC

CLRVEC R4



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 86  
 CVDVAD.P11 12-JUL-84 05:04 T33 TRANSMITTER INTERRUPT LOGIC TEST

```

3458 ;*****
3459 ;*****
3460 ;*TEST 34 RECIIVER INTERRUPT LOGIC TEST
3461 ;* THIS TEST COVERS ALL OF THE RECEIVER
3462 ;* SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
3463 ;* AND CHARACTER MODES.
3464 ;*****
3465 007530 000004 TST34: SCOPE
3466 007532 012767 000010 171420 MOV #10,#TIMES ;;DO 10 ITERATIONS
3467 007540 012767 000034 171432 MOV #34,#TESTN ;;SET TEST NUMBER IN APT MAIL BOX
3468 ;CLEAR INTERRUPT OCCURED FLAG
3469 ;SET UP RECEIVER INTER.VECTOR
3470 007546 SETVEC DLVEC,#INTSRV,#PR7
3471 007546 010146 MOV R1,-(SP)
3472 007550 016701 171502 MOV DLVEC,R1
3473 007554 012721 012032 MOV #INTSRV,(R1)
3474 007560 012711 000340 MOV #PR7,(R1)
3475 007564 012601 MOV (SP),R1
3476 ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
3477 007566 BGNSUB
3478 007566 012767 007574 171314 MOV #64,#LPERR
3479 007574 LET INTFLAG := #0
3480 007574 005067 002240 CLR INTFLAG
3481 ;SET MAINT. BIT
3482 007600 LET @TCSR := @TCSR SET.BY #MAINT
3483 007600 052777 000004 171456 BIS #MAINT,@TCSR
3484 ;CLEAR INTERRUPTS
3485 007606 LET @RCSR := @RCSR CLR.BY #RCVRIE
3486 007606 042777 000100 171444 BIC #RCVRIE,@RCSR
3487 ;CHANGE PRIORITY
3488 ;..TO 0
3489 007614 012746 000000 MOV #PRO,-(SP) ;;PUT NEW PS ON STACK
3490 007620 012746 007626 MOV #65,-(SP) ;;PUT NEW PC ON STACK
3491 007624 000002 RTI ;;POP NEW PC AND PS
3492 007626 65:
3493
3494 ;SEND A CHARACTER
3495 007626 LET @TBUF :B= #0
3496 007626 105077 171436 CLRB @TBUF
3497 ;WAIT A MAXIMUM
3498 ;OF 1000 MSEC FOR
3499 ;RCVR RDY TO SET IN RCSR
3500 CALL TIMER IN <#1000,@RCVRDONE,RCSR,#SET>
3501 007632 MOV R5,-(SP)
3502 007634 010546 MOV #SET,-(R5)
3503 007640 016745 171414 MOV RCSR,-(R5)
3504 007644 012745 000200 MOV #RCVRDONE,-(R5)
3505 007650 012745 001000 MOV #1000,-(R5)
3506 007654 004767 001614 JSR PC,TIMER
3507 007660 012605 MOV (SP),R5
3508 ;SET INTERRUPT ENABLE
3509 007662 LET @RCSR := @RCSR SET.BY #RCVRIE
3510 007662 052777 000100 171370 BIS #RCVRIE,@RCSR
3511 ;LET IT COME IN.
3512 007670 WAITMS 1
3513 007670 010546 MOV R5,-(SP)

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 87  
 CVDVAD.P11 12-JUL-84 05:04 T34 RECEIVER INTERRUPT LOGIC TEST

```

3514 007672 012745 000001      MOV    #1,-(R5)
3515 007676 004767 002050      JSR    PC,WAIT
3516 007702 012605              MOV    (SP)+,R5
3517 007704                      LET R0 := @RBUF ; CLEAR RCVRDONE
3518 007704 017700 171352      MOV    @RBUF,R0
3519
3520                      ;DID HE DO IT RIGHT?
3521 007710                      IF INTFLAG NE #1 THEN
3522 007710 026727 002124 000001  CMP    INTFLAG,#1
3523 007716 001406              BEQ    #160
3524                      ;NONE OCCURED
3525 007720                      IF INTFLAG EQ #0 THEN
3526 007720 005767 002114      TST    INTFLAG
3527 007724 001002              BNE    #161
3528                      ;RECEIVER DID NOT INTERRUPT IN TIME
3529 007726                      ERRHRD 111,,DIDNOT
3530 007726 104111              ERROR  111
3531                      ;TWICE OR MORE
3532 007730                      ELSE
3533 007730 000401              BR     #162
3534 007732                      ;RECEIVER INTERRUPTED TWICE
3535                      ERRHRD 112,,TWICE
3536 007732                      ENDIF
3537 007732 104112              ERROR  112
3538 007734                      ENDIF
3539 007734                      ;RECEIVER INTERRUPTED TWICE
3540 007734                      ERRHRD 112,,TWICE
3541 007734                      ENDIF
3542                      ;RESET MAINT. BIT.
3543 007734                      LET @TCSR := @TCSR CLR.BY @MAINT
3544 007734 042777 000004 171322  BIC    @MAINT,@TCSR
3545                      ; CLEAR INTERRUPT ENABLE
3546 007742                      LET @RCSR := @RCSR CLR.BY @RCVRIE
3547 007742 042777 000100 171310  BIC    @RCVRIE,@RCSR
3548 007750                      ENDSUB
3549
3550
3551
3552
3553
3554
3555
3556                      ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3557 007750                      BGNSUB
3558 007750 012767 007756 171132  MOV    #64,@LPERR
3559 007756                      IF @CABLE NOTSETIN @USWR THEN
3560 007756 032767 020000 171234  BIT    @CABLE,@USWR
3561 007764 001004              BNE    #163
3562                      ;CAN'T TEST WITHOUT A CABLE
3563 007766                      EXIT TST
3564 007766 012767 000001 171164  MOV    #1,@TIMES
3565 007774 000466              BR     TST35
3566 007776                      ;:EXIT THIS TEST
3567 007776                      ENDIF
3568                      ; CLEAR 'INTFLAG'
3569 007776                      LET INTFLAG := #0

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 88  
 CVDVAD.P11 12-JUL-84 05:04 T34 RECEIVER INTERRUPT LOGIC TEST

```

3570 007776 005067 002036          CLR      INTFLAG
3571                                ;CLEAR INTERRUPTS
3572 010002                                LET @RCSR := @RCSR CLR.BY @DATAIE
3573 010002 042777 000040 171250      BIC      @DATAIE,@RCSR
3574                                ;CHANGE PRIORITY
3575                                ;..TO 0
3576 010010 012746 000000          MOV      @PRO,-(SP)      ;;PUT NEW PS ON STACK
3577 010014 012746 010022          MOV      @64,-(SP)      ;;PUT NEW PC ON STACK
3578 010020 000002                                RTI                ;;POP NEW PC AND PS
3579 010022                                64:
3580 010022                                LET @RCSR := @RCSR CLR.BY @REQSEND
3581 010022 042777 000004 171230      BIC      @REQSEND,@RCSR
3582                                ;SET INTERRUPT ENABLE
3583 010030                                LET @RCSR := @RCSR SET.BY @DATAIE
3584 010030 052777 000040 171222      BIS      @DATAIE,@RCSR
3585 010036                                LET @RCSR := @RCSR SET.BY @REQSEND
3586 010036 052777 000004 171214      BIS      @REQSEND,@RCSR
3587                                ;LET IT COME IN.
3588 010044                                WAITMS 1
3589 010044 010546          MOV      R5,-(SP)
3590 010046 012745 000001          MOV      @1,-(R5)
3591 010052 004767 001674          JSR      PC,WAIT
3592 010056 012605          MOV      (SP)+,R5
3593
3594                                ; DID IT DO IT RIGHT?
3595 010060                                IF INTFLAG NE @1 THEN
3596 010060 026727 001754 000001      CMP      INTFLAG,@1
3597 010066 001406          BEQ      @164
3598                                ;NONE OCCURED
3599 010070                                IF INTFLAG EQ @0 THEN
3600 010070 005767 001744          TST      INTFLAG
3601 010074 001002          BNE      @165
3602                                ;DATAINT DID NOT INTERRUPT IN TIME
3603 010076                                ERRHRD 113,,DIDNOT
3604 010076 104113          ERROR    113
3605                                ;TWICE OR MORE
3606 010100                                ELSE
3607 010100 000401          BR      @166
3608 010102                                @165:
3609                                ; DATAINT INTERRUPTED TWICE
3610 010102                                ERRHRD 114,,TWICE
3611 010102 104114          ERROR    114
3612 010104                                @166:
3613 010104                                @164:
3614 010104                                @164:
3615 010104                                @164:
3616 010104                                @164:
3617 010104 042777 000040 171146      BIC      @DATAIE,@RCSR
3618 010112                                LET @RCSR := @RCSR CLR.BY @REQSEND
3619 010112 042777 000004 171140      BIC      @REQSEND,@RCSR
3620 010120                                ENDSUB
3621
3622 010120                                LET R4 := @DLVEC
3623 010120 017704 171132          MOV      @DLVEC,R4
3624 010124                                CLRVEC R4
3625 010124 010146          MOV      R1,-(SP)      ;;PUSH R1 ON STACK

```

MAINDEC-ZZ-CVDVA-D    MACY11 30A(1052) 12-JUL-84 09:52 PAGE 89  
CVDVAD.P11    12-JUL-84 05:04    T34    RECEIVER INTERRUPT LOGIC TEST

3626	010126	010246		MOV	R2,-(SP)	::PUSH R2 ON STACK
3627	010130	012701	000004	MOV	#R4,R1	
3628	010134	010102		MOV	R1,R2	
3629	010136	062702	000002	ADD	#2,R2	
3630	010142	010221		MOV	R2,(R1)+	
3631	010144	005011		CLR	(R1)	
3632	010146	012602		MOV	(SP)+,R2	::POP STACK INTO R2
3633	010150	012601		MOV	(SP)+,R1	::POP STACK INTO R1
3634	010152					ENDTST



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 90  
 CVDVAD.P11 12-JUL-84 05:04 T34 RECEIVER INTERRUPT LOGIC TEST

```

3635 ;*****
3636 ;*****
3637 ;*TEST 35 TEST ACTUAL DATA TRANSFERED
3638 ;* NON-INTERRUPT MAINTENANCE BIT SET
3639 ;*****
3640 010152 000004 TST35: SCOPE
3641 010154 012767 000001 170776 MOV #1,#TIMES ;DO 1 ITERATION
3642 010162 012767 000035 171010 MOV #35,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
3643 ;SET MAINT. BIT
3644 010170 LET @TCSR := @TCSR SET.BY #MAINT
3645 010170 052777 000004 171066 BIS #MAINT,@TCSR
3646
3647 ;CHANGE PRIORITY
3648 ;..TO 0
3649 010176 012746 000000 MOV #PRO,-(SP) ;PUT NEW PS ON STACK
3650 010202 012746 010210 MOV #64,-(SP) ;PUT NEW PC ON STACK
3651 010206 000002 RTI ;POP NEW PC AND PS
3652 010210 64:
3653 ;GET DATA MASK.
3654 010210 CALL DATLNG OUT <R1>
3655 010210 162705 000002 SUB #1*2,R5
3656 010214 004767 001432 JSR PC,DATLNG
3657 010220 012501 MOV (R5)+,R1
3658
3659 010222 ; START CLEAN
3660 010222 017700 171034 MOV @RBUF,R0
3661
3662 ;ALL BINARY CHAR.
3663 010226 INCR R2 FROM #0 TO #377 BY #1
3664 010226 005002 CLR R2
3665 010230 000401 BR #167
3666 010232 $170:
3667 010232 005202 INC R2
3668 010234 $167:
3669 010234 020227 000377 CMP R2,#377
3670 010240 003047 BGT #171
3671
3672
3673 ;TRANSMIT CHAR IN R2
3674 CALL TIMER IN <#1000,#XMITRDY,TCSR,#SET>
3675 010242
3676 010242 010546 MOV R5,-(SP)
3677 010244 012745 177777 MOV #SET,-(R5)
3678 010250 016745 171010 MOV TCSR,-(R5)
3679 010254 012745 000200 MOV #XMITRDY,-(R5)
3680 010260 012745 001000 MOV #1000,-(R5)
3681 010264 004767 001204 JSR PC,TIMER
3682 010270 012605 MOV (SP)+,R5
3683
3684 ;TRANSMIT IT
3685 010272 LET @TBUF :B= R2
3686 010272 110277 170772 MOVB R2,@TBUF
3687
3688 CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
3689 010276 010546 MOV R5,-(SP)
3690 010300 012745 177777 MOV #SET,-(R5)

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 92  
 CVDVAD.P11 12-JUL-84 05:04 T35 TEST ACTUAL DATA TRANSFERED

```

3734      ;:*****
3735      ;:*****
3736      ;*TEST 36      TEST DATA THROUGH CABLE
3737      ;:*****
3738      010366 000004      TST36: SCOPE
3739      010370 012767 000001 170562      MOV      @1,@TIMES      ;;DO 1 ITERATION
3740      010376 012767 000036 170574      MOV      @36,@TESTN    ;;SET TEST NUMBER IN APT MAIL BOX
3741      010404      BIT      @CABLE,@USWR      IF @CABLE NOTSETIN @USWR THEN
3742      010404 032767 020000 170606      BNE      @173
3743      010412 001004
3744      ;CAN'T TEST WITHOUT A CABLE
3745      010414      EXIT TST
3746      010414 012767 000001 170536      MOV      @1,@TIMES
3747      010422 000474      BR       TST37      ;;EXIT THIS TEST
3748      010424      ;ENDIF
3749      010424      @173:
3750      ;DON'T USE MAINT.
3751      010424      LET @TCSR := @TCSR CLR.BY @MAINT
3752      010424 042777 000004 170632      BIC      @MAINT,@TCSR
3753      ;CHANGE PRIORITY
3754      ;..TO 0
3755      010432 012746 000000      MOV      @PRO,-(SP)    ;;PUT NEW PS ON STACK
3756      010436 012746 010444      MOV      @64@,-(SP)   ;;PUT NEW PC ON STACK
3757      010442 000002      RTI      ;;POP NEW PC AND PS
3758      010444      64@:
3759      ;GET DATA MASK
3760      010444      CALL DATLNG OUT <R1>
3761      010444 162705 000002      SUB      @1*2,R5
3762      010450 004767 001176      JSR      PC,DATLNG
3763      010454 012501      MOV      (R5)+,R1
3764      010456
3765      010456 017700 170600      MOV      @RBUF,R0
3766      ;BINARY COUNT PATTERN
3767      010462      INCR R2 FROM @0 TO @377 BY @1
3768      010462 005002      CLR      R2
3769      010464 000401      BR       @174
3770      010466      @175:
3771      010466 005202      INC      R2
3772      010470      @174:
3773      010470 020227 000377      CMP      R2,@377
3774      010474 003047      BGT      @176
3775
3776
3777      ;TRANSMIT THE CHAR. IN R2.
3778      CALL TIMER IN <@1000,@XMITRDY,TCSR,@SET>
3779      010476
3780      010476 010546      MOV      R5,-(SP)
3781      010500 012745 177777      MOV      @SET,-(R5)
3782      010504 016745 170554      MOV      TCSR,-(R5)
3783      010510 012745 000200      MOV      @XMITRDY,-(R5)
3784      010514 012745 001000      MOV      @1000,-(R5)
3785      010520 004767 000750      JSR      PC,TIMER
3786      010524 012605      MOV      (SP)+,R5
3787
3788      ;START IT ON ITS WAY
3789      010526      LET @TBUF :B= R2

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 93  
 CVDVAD.P11 12-JUL-84 05:04 T36 TEST DATA THROUGH CABLE

```

3790 010526 110277 170536      MOVB   R2,@TBUF
3791 010532
3792 010532 010546
3793 010534 012745 177777      MOV    R5,-(SP)
3794 010540 016745 170514      MOV    #SET,-(R5)
3795 010544 012745 000200      MOV    RCSR,-(R5)
3796 010550 012745 001000      MOV    #RCVRDONE,-(R5)
3797 010554 004767 000714      MOV    #1000,-(R5)
3798 010560 012605
3799
3800
3801 010562
3802 010562 017703 170474      JSR    PC,TIMER
3803
3804
3805 010566
3806 010566 010204      MOV    @RBUF,R3
3807 010570 040104
3808 010572
3809 010572 040103      BIC   R1,R3
3810
3811
3812 010574
3813 010574 020403      BIC   R1,R4
3814 010576 001405      BIC   R1,R3
3815
3816 010600
3817 010600 104117      CMP   R4,R3
3818 010602
3819 010602 012767 000001 170350      BEQ   #177
3820 010610 000401
3821 010612
3822 010612
3823
3824 010612
3825 010612 000725      ERROR 117
3826 010614
3827
3828
3829
3830 010614      MOV   #1,@TIMES
3831
3832
3833
3834

```

CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>

;RETRIEVE  
LET R3 := @RBUF

;STRIP OFF JUNK ON BOTH  
LET R4 := R2 CLR.BY R1

LET R3 := R3 CLR.BY R1

;WE HAVE TROUBLE  
IF R4 NE R3 THEN

;DATA COMPARE ERROR  
ERRHRD 117,COMP,SBWAS  
EXIT TST ; ON ERROR

:::EXIT THIS TEST  
ENDIF

#177:

ENDINC ; R2

#176:

ENDTST

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 94  
CVDVAD.P11 12-JUL-84 05:04 T36 TEST DATA THROUGH CABLE

```

3835 ;:.....
3836 ;:.....
3837 ;*TEST 37 FULL DATA TRANSFER WITH INTERRUPTS
3838 ;* AND MAINTENANCE MODE.
3839 ;:.....
3840 010614 000004 TST37: SCOPE
3841 010616 012767 000001 170334 MOV #1,#TIMES ;:DO 1 ITERATION
3842 010624 012767 000037 170346 MOV #37,#TESTN ;:SET TEST NUMBER IN APT MAIL BOX
3843
3844
3845 ; THIS TEST IS 'BREAK OR HALT' SENSITIVE.
3846 010632 ; IF #APTENV SETIN #ENV THEN
3847 010632 032767 000001 170354 BIT #APTENV,#ENV
3848 010640 001404 BEQ #200
3849 010642 ; EXIT TEST
3850 010642 012767 000001 170310 MOV #1,#TIMES
3851 010650 000550 BR TST40 ;:EXIT THIS TEST
3852 010652 ;:ENDIF
3853 010652 ;:200:
3854 ;:GET DATA MASK
3855 010652 ;:CALL DATLNG OUT <R3>
3856 010652 162705 000002 SUB #1*2,R5
3857 010656 004767 000770 JSR PC,DATLNG
3858 010662 012503 MOV (R5),R3
3859
3860
3861 ; THIS TEST WILL RUN BOTH TRANSMITTER AND
3862 ; RECIEVER AT FULL SPEED TESTING
3863 ; THE ABILITY OF THE MODULE
3864 ; TO HANDLE INTERRUPTS FROM BOTH SIDES
3865 ; AT ONCE. ALSO, THE DOUBLE BUFFERING LOGIC
3866 ; OF THE UART WILL BE FULLY TESTED.
3867 ; THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
3868 ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
3869 ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.
3870
3871
3872 ;:CHANGE PRIORITY
3873 ;:..TO 0
3874 010664 012746 000000 MOV #PRO,-(SP) ;:PUT NEW PS ON STACK
3875 010670 012746 010676 MOV #64,-(SP) ;:PUT NEW PC ON STACK
3876 010674 000002 RTI ;:POP NEW PC AND PS
3877 010676 ;:644:
3878 ;:GET VECTOR ADDRESS
3879 010676 LET R1 := DLVEC
3880 010676 016701 170354 MOV DLVEC,R1
3881 ;:RCVR VECTOR
3882 010702 LET (R1) := #REC
3883 010702 012721 011076 MOV #REC,(R1)
3884 010706 LET (R1) := #PR7
3885 010706 012721 000340 MOV #PR7,(R1)
3886 ;:POINT TO TRANSMITTER VECTOR
3887 ;:AND SET IT UP ALSO
3888 010712 LET (R1) := #TRAN
3889 010712 012721 011034 MOV #TRAN,(R1)
3890 010716 LET (R1) := #PR7

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 95  
 CVDVAD.P11 12-JUL-84 05:04 T37 FULL DATA TRANSFER WITH INTERRUPTS

```

3891 010716 012711 000340      MOV    @PR7,(R1)
3892
3893
3894 010722                      ; INITIALIZE COUNTERS
3895 010722 012701 177777      MOV    @-1,R1      LET R1 := @-1
3896                      ;RECEIVER STORAGE
3897 010726                      LET R2 := @0
3898 010726 005002      CLR    R2
3899                      ;@ OF RECEIVED CHAR. COUNT.
3900 010730                      LET R4 := @-1
3901 010730 012704 177777      MOV    @-1,R4
3902
3903                      ; CLEAR ERROR COUNT.
3904 010734                      LET ERRCNT := @0
3905 010734 005067 000066      CLR    ERRCNT
3906
3907 010740                      BRESET ;SET UP ALL REGISTERS
3908 010740 000005      RESET
3909                      ;SET UP MAINTENANCE
3910 010742                      LET @TCSR := @TCSR SET.BY @MAINT
3911 010742 052777 000004 170314  BIS    @MAINT,@TCSR
3912
3913                      ;SET I.E. IN TRANSMITTER
3914 010750                      LET @TCSR := @TCSR SET.BY @XMITIE
3915 010750 052777 000100 170306  BIS    @XMITIE,@TCSR
3916                      ;AND RECEIVER
3917 010756                      LET @RCSR := @RCSR SET.BY @RCVRIE
3918 010756 052777 000100 170274  BIS    @RCVRIE,@RCSR
3919
3920
3921                      ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3922 010764                      REPEAT
3923 010764                      UNTIL R4 EQ NUMBER OR ERRCNT GT @0
3924 010764                      $201:
3925 010764 020467 000040      CMP    R4,NUMBER
3926 010770 001403      BEQ    $202
3927 010772 005767 000030      TST    ERRCNT
3928 010776 003772      BLE    $201
3929 011000                      $202:
3930
3931                      ;DATA COMPARE ERRORS.
3932 011000                      IF ERRCNT NE @0 THEN
3933 011000 005767 000022      TST    ERRCNT
3934 011004 001401      BEQ    $203
3935                      ;DATA COMPARE ERROR
3936 011006                      ERRHRD 120,COMP,FIRST
3937 011006 104120      ERROR  120
3938 011010                      ENDIF
3939 011010                      $203:
3940
3941 011010                      LET @TCSR := @TCSR CLR.BY @XMITIE
3942 011010 042777 000100 170246  BIC    @XMITIE,@TCSR
3943 011016                      LET @RCSR := @RCSR CLR.BY @XMITIE
3944 011016 042777 000100 170234  BIC    @XMITIE,@RCSR
3945
3946 011024                      EXIT   ;SKIP OVER SUPPORT ROUTINES & STORAGE

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 96  
 CVDVAD.P11 12-JUL-84 05:04 T37 FULL DATA TRANSFER WITH INTERRUPTS

```

3947 011024 000462          BR      TST40          ;;;EXIT THIS TEST
3948
3949 011026 000000          ERRCNT: 0
3950 011030 000400          NUMBER: 400
3951 011032      000          SB:      .BYTE 0
3952 011033      000          WAS:      .BYTE 0
3953
3954
3955
3956          ;*****
3957          ;TRANSMIT INTERRUPT HANDLER
3958
3959 011034          BGNSRV  TRAN
3960 011034
3961          TRAN:
3962          ;*****
3963          ;INCREMENT CHAR COUNT
3964 011034          LET R1 := R1 + #1
3965 011034 005201          INC      R1
3966          ;SET UP FOR TRANSFER
3967 011036          LET HOLD := R1      CLR.BY R3
3968 011036 010167 000030      MOV     R1,HOLD
3969 011042 040367 000024      BIC     R3,HOLD
3970          ;AND SEND.
3971 011046          LET @TBUF := HOLD
3972 011046 016777 000020 170214  MOV     HOLD,@TBUF
3973          ;ALL DONE
3974 011054          IF R1 EQ NUMBER THEN
3975 011054 020167 177750      CMP     R1,NUMBER
3976 011060 001003          BNE     #204
3977          ;STOP INTERRUPT PROCESSING
3978 011062          LET @TCSR := @TCSR CLR.BY #XMITIE
3979 011062 042777 000100 170174  BIC     #XMITIE,@TCSR
3980 011070          ENDIF
3981 011070          #204:
3982
3983 011070 000401          BR      ZZZ          ; EXIT SRV
3984
3985 011072 000000          HOLD:0
3986
3987 011074          ZZZ:          ENDSRV
3988 011074 000002          RTI
3989
3990
3991          ;*****
3992          ;RECEIVER INTERRUPT HANDLER
3993          BGNSRV  REC
3994 011076
3995 011076
3996          REC:
3997          ;*****
3998          ;COUNT THIS CHAR.
3999 011076          LET R4 := R4 + #1
4000 011076 005204          INC      R4
4001          ;GET CHAR IN + MASK IT
4002 011100          LET R2 := @RBUF CLR.BY R3

```

```

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 97
CVDVAD.P11      12-JUL-84 05:04      T37      FULL DATA TRANSFER WITH INTERRUPTS
4003 011100 017702 170156          MOV  @RBUF,R2
4004 011104 040302          BIC  R3,R2
4005                                     ;RHLD WILL CONTAIN EXPECTED INPUT
4006 011106          LET RHLD := R4 CLR.BY R3
4007 011106 010467 000054          MOV  R4,RHLD
4008 011112 040367 000050          BIC  R3,RHLD
4009
4010                                     ;DO THEY COMPARE
4011 011116          IF R2 NE RHLD THEN
4012 011116 020267 000044          CMP  R2,RHLD
4013 011122 001412          BEQ  $205
4014
4015                                     ;FIRST ERROR
4016 011124 005767 177676          TST  ERRCNT
4017 011130 001005          BNE  $206
4018
4019                                     ;SAVE RECORD OF FIRST MISS
4020 011132 116767 000030 177672    MOVB RHLD,SB
4021 011140          LET SB := RHLD
4022 011140 110267 177667    MOVB R2,WAS
4023 011144          LET WAS := R2
4024 011144          ENDIF
4025                                     ;COUNT IT.
4026 011144          LET ERRCNT := ERRCNT + #1
4027 011144 005267 177656          INC  ERRCNT
4028 011150          ENDIF
4029 011150          $205:
4030
4031                                     ;ALL DONE?
4032 011150          IF R4 EQ NUMBER THEN
4033 011150 020467 177654          CMP  R4,NUMBER
4034 011154 001003          BNE  $207
4035
4036                                     ;STOP RECEIVER INTERRUPTS
4037 011156 042777 000100 170074    BIC  @RCVRIE,@RCSR
4038
4039                                     ;MAIN REPEAT LOOP IS CHECKING
4040 011164          ENDIF
4041
4042                                     ;FOR 'R4 = NUMBER' ALSO
4043 011164 000401          BR   ZZZZ
4044
4045                                     ;EXIT SRV
4046          ZZZZ:          RHLD:0
4047          RTI
4048 011170 000002          RTI
4049
4050 011172          ENDSRV
4051
4052          ENDTST
4053

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 98  
 CVDVAD.P11 12-JUL-84 05:04 T37 FULL DATA TRANSFER WITH INTERRUPTS

```

4054 ;*****
4055 ;*****
4056 ;*TEST 40      TEST BREAK GENERATION LOGIC
4057 ;*            TRANSMIT KNOWN CHAR WITH BREAK SET
4058 ;*            AND COMPARE RECEIVED WITH 0.
4059 ;*****
4060 011172 000004 TST40: SCOPE
4061 011174 012767 000010 167756      MOV      #10,#TIMES      ;;DO 10 ITERATIONS
4062 011202 012767 000040 167770      MOV      #40,#TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
4063 011210                                IF #BRK NOTSETIN #USWR THEN
4064 011210 032767 010000 170002      BIT      #BRK,#USWR
4065 011216 001004                                BNE      #210
4066 011220                                EXIT TST
4067 011220 012767 000001 167732      MOV      #1,#TIMES
4068 011226 000452                                BR       TST41          ;;EXIT THIS TEST
4069 011230                                ENDIF
4070 011230                                $210:
4071                                ;SET MAINTENANCE BIT
4072 011230                                LET @TCSR := @TCSR SET.BY #MAINT
4073 011230 052777 000004 170026      BIS      #MAINT,@TCSR
4074
4075                                ; CLEAR RCVRDONE JUST IN CASE
4076 011236                                LET RO := @RBUF
4077 011236 017700 170020      MOV      @RBUF,RO
4078
4079
4080                                ;SET BREAK BIT
4081 011242                                LET @TCSR := @TCSR SET.BY #BREAK
4082 011242 052777 000001 170014      BIS      #BREAK,@TCSR
4083
4084                                ;NON-ZERO CHAR. '*'
4085 011250                                LET @TBUF := #252
4086 011256                                CALL TIMER IN <#1000,@RCVRDONE,RCSR,#SET>
4087 011256 010546      MOV      R5,-(SP)
4088 011260 012745 177777      MOV      #SET,-(R5)
4089 011264 016745 167770      MOV      RCSR,-(R5)
4090 011270 012745 000200      MOV      @RCVRDONE,-(R5)
4091 011274 012745 001000      MOV      #1000,-(R5)
4092 011300 004767 000170      JSR      PC,TIMER
4093 011304 012605      MOV      (SP),R5
4094 011306
4095 011306 103001      BCC      #211
4096
4097 011310                                ; RECIEVER DONE DID NOT SET
4098 011310 104115      ERROR    115
4099 011312                                ENDIF
4100 011312                                $211:
4101
4102 011312                                IFB @RBUF NE #0 THEN
4103 011312 105777 167744      TSTB    @RBUF
4104 011316 001401      BEQ     #212
4105
4106                                ; BREAK DID NOT EQUAL 0
4107 011320                                ERRHRD 121 ,BADBRK
4108 011322                                ENDIF
4109 011322                                $212:

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 99  
CVDVAD.P11 12-JUL-84 05:04 T40 TEST BREAK GENERATION LOGIC

```
4110 011322                                BRESET ;CLEAN UP
4111 011322 000005                        RESET
4112 011324                                EXIT
4113 011324 000413                        BR      TST41      ;;;EXIT THIS TEST
4114 011326 051102 040505 020113          BADBRK: .ASCIZ /BREAK DID NOT EQUAL 0/
4115 011334 044504 020104 047516
4116 011342 020124 050505 040525
4117 011350 020114 000060
4118
4119 011354                                ENDTST
```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 100  
 CVDVAD.P11 12-JUL-84 05:04 T40 TEST BREAK GENERATION LOGIC

```

4120
4121
4122
4123
4124 011354 000004
4125 011356 012767 000001 167574
4126 011364 104401 011372
4127 011370 000404
4128
4129 011402
4130 011402 016746 167646
4131 011406 104402
4132 011410 104401 011416
4133 011414 000405
4134
4135 011430
4136 011430 016746 167622
4137 011434 104402
4138 011436 104401 011444
4139 011442 000405
4140
4141 011456
4142 011456 016746 167430
4143 011462 104405
4144 011464 005067 167422
4145 011470 000167 170250

;*****
;TEST 41 NOT A TEST - SEND BACK TO LOOP
;*****
TST41: SCOPE
MOV #1,#TIMES ;;DO 1 ITERATION
TYPE ,65# ;;TYPE ASCIZ STRING
BR 64# ;;GET OVER THE ASCIZ
;;65#: .ASCIZ <CRLF>*LSR: *
64#:
MOV DLADD,-(SP) ;;SAVE DLADD FOR TYPEOUT
TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
TYPE ,67# ;;TYPE ASCIZ STRING
BR 66# ;;GET OVER THE ASCIZ
;;67#: .ASCIZ *,VECTOR: *
66#:
MOV DLVEC,-(SP) ;;SAVE DLVEC FOR TYPEOUT
TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
TYPE ,69# ;;TYPE ASCIZ STRING
BR 68# ;;GET OVER THE ASCIZ
;;69#: .ASCIZ *,ERRORS: *
68#:
MOV #ERTTL,-(SP) ;;SAVE #ERTTL FOR TYPEOUT
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
CLR #ERTTL ; RESET FOR NEXT DEVICE/PASS
JMP LOOP ; BACK UP TO THE BEGINNING

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 101  
 CVDVAD.P11 12-JUL-84 05:04 T41 NOT A TEST - SEND BACK TO LOOP

```

4146
4147
4148
4149 011474
4150 011474
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171          000001
4172          000000
4173
4174 011474
4175 011474 016567 000004 000136
4176 011502
4177 011502 016567 000000 000132
4178 011510
4179 011510 112767 000000 000126
4180
4181
4182
4183
4184 011516
4185 011516
4186
4187 011516
4188 011516 036577 000002 000114
4189 011524 001004
4190 011526
4191 011526 112767 000000 000111
4192 011534
4193 011534 000403
4194 011536
4195 011536
4196 011536 112767 177777 000101
4197 011544
4198 011544
4199
4200
4201 011544

```

```

          ;:BGNMOD          SUBS
          ;:*****
          ;:ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
TIMER:
;* ROUTINE:TIMER
;* THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
;* IN ANY REGISTER.
;* INPUTS:
;* HOWLONG THE MAXIMUM AMOUNT OF TIME TO SPEND IN
;* THIS ROUTINE.
;* WHICHBIT A MASK WITH THE BIT(S) SET THAT ARE
;* TO BE CHECKED.
;* REG A POINTER TO THE REGISTER TO BE CHECKED
;* SETCLR THE DESIRED RESULTS
;* EITHER #SET OR #CLEAR
;* OUTPUT:
;* THE 'C' BIT IS SET TO INDICATE AN ERROR
;* BUT IT IS TESTED BY THE IF.ERROR STATEMENT
;*
;* NOTE:: THE USE OF (R5) IS PART OF THE LINKAGE
;* MECHANISM BETWEEN THE CALLER AND THE CALLED
;*:*****
          TRUE= 1
          FALSE= 0
          LET REGSAV := REG(R5) ; GET POINTER TO REGIST
          LET TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
          LET FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
          ; START OF AN INFINITE LOOP
          LOOP
          ; TEST TO SEE IF WHICHBIT IS SET
          IF WHICHBIT(R5) NOTSETIN @REGSAV THEN
          LET HOLDSC :B= #CLR
          ELSE
          LET HOLDSC :B= #SET ; REMEMBER THIS
          ENDIF
          ; NOW SEE IF THAT WAS WHAT WE WANTED
          IFB HOLDSC EQ SETCLR(R5) THEN

```

```

          $215:
          BIT WHICHBIT(R5),@REGSAV
          BNE $217
          MOV #CLR,HOLDSC
          BR $220
          $217:
          MOVB #SET,HOLDSC
          $220:

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 102  
 CVDVAD.P11 12-JUL-84 05:04 ROUTINE - TIMER

```

4202 011544 126765 000075 000006      CMPB  HOLDSC,SETCLR(R5)
4203 011552 001003                    BNE   $221
4204                                     ; JUST THE THING WE NEEDED
4205 011554                                     LET   FLAG :B= #TRUE
4206 011534 112767 000001 000062      MOVB  #TRUE,FLAG
4207 011562                                     ENDIF
4208 011562                                $221:
4209
4210 011562                                     EXIFB FLAG EQ #TRUE OR TIMSAV LE #0
4211 011562 126727 000056 000001      CMPB  FLAG,#TRUE
4212 011570 001414                    BEQ   $216
4213 011572 005767 000044                    TST  TIMSAV
4214 011576 003411                    BLE  $216
4215
4216                                     ; ONE WAY OR THE OTHER, WE ARE DONE
4217                                     ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4218 011600                                     WAITMS 1           ;WAIT FOR 10 MILLI-SECONDS
4219 011600 010546                    MOV   R5,-(SP)
4220 011602 012745 000001                    MOV   #1,-(R5)
4221 011606 004767 000140                    JSR   PC,WAIT
4222 011612 012605                    MOV   (SP)+,R5
4223 011614                                     LET   TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4224 011614 005367 000022                    DEC  TIMSAV
4225 011620                                     ENDLOOP           ; CONTINUED AT THE TOP
4226 011620 000736                    BR   $215
4227 011622                                $216:
4228
4229                                     ; ONLY 2 WAYS TO GET HERE
4230                                     ; 1). WE RAN OUT OF TIME---ERROR !!
4231                                     ; 2). THE BIT IS IN THE CORRECT CONDITION--GOOD !!
4232
4233 011622                                     IFB   FLAG EQ #TRUE THEN
4234 011622 126727 000016 000001      CMPB  FLAG,#TRUE
4235 011630 001001                    BNE  $222
4236 011632                                     RETURN NO.ERROR   ; GOOD
4237 011632 000405                    BR   $213
4238 011634                                     ENDIF
4239 011634                                $222:
4240 011634                                     RETURN ERROR      ; BAD
4241 011634 000261                    SEC
4242 011636 000404                    BR   $214
4243
4244 011640 000000                                     REGSAV: .WORD 0
4245 011642 000000                                     TIMSAV: .WORD 0
4246 011644 000                                     FLAG:   .BYTE 0
4247 011645 000                                     HOLDSC: .BYTE 0
4248                                     ; WE ARE DONE GO BACK HOME
4249 011646                                     ENDRTN
4250 011646                                $213:
4251 011646 000241                    CLC
4252 011650                                $214:
4253 011650 000207                    RTS   PC

```

```

4254
4255
4256 011652
4257 011652
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271 011652
4272 011652 005065 000000
4273 011656
4274 011656 016767 167336 000062
4275 011664 016746 000056
4276 011670 042716 000017
4277 011674 042667 000046
4278
4279 011700
4280 011700 012767 000001 167364
4281 011706 000402
4282 011710
4283 011710 005267 167356
4284 011714
4285 011714 026767 167352 000024
4286 011722 003006
4287 011724
4288 011724 006365 000000
4289 011730
4290 011730 052765 000001 000000
4291 011736
4292 011736 000764
4293 011740
4294 011740
4295 011740 005165 000000
4296 011744
4297 011744 000401
4298 011746 000000
4299 011750
4300 011750
4301 011750
4302 011750 000207

```

```

*****
ROUTINE DATLNG <MASK>
DATLNG:
;* ROUTINE:DATLNG
;* THIS ROUTINE SETS UP A MASK FOR DATA, WITH
;* INPUT - NOTHING IS PASSED TO THIS ROUTINE
;* BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:
;* $USMR-- THE WORD FOR SOFTWARE PARAMETERS
;* DATA-- A MASK FOR THE LOCATION OF THE OCTAL
;* NUMBER OF DATA BITS
;* OUTPUT----
;* MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
;* THE NUMBER OF WHICH IS DEFINED IN $USMF WORD.
*****
LET MASK(R5) := #0 ; START
LET NUMBR := $USMR AND #DATA
MOV $USMR,NUMBR
MOV NUMBR,-(SP)
BIC #DATA,(SP)
BIC (SP)+,NUMBR
INCR I FROM #1 TO NUMBR BY #1
MOV #1,I
BR $225
$226: INC I
$225: CMP I,NUMBR
BGT $227
LET MASK(R5) := MASK(R5) SHIFT #1
LET MASK(R5) := MASK(R5) SET.BY #1
ENDINC
LET MASK(R5) := COMP MASK(R5)
RETURN
$227:
COM MASK(R5)
BR $223
NUMBR:0
ENDRTN
$223:
$224: RTS PC

```

```

4303
4304
4305 011752
4306 011752
4307
4308
4309
4310
4311
4312
4313
4314 011752 010146
4315 011754 010246
4316 011756 010346
4317 011760
4318 011760 016501 000000
4319 011764
4320 011764 012702 000001
4321 011770 000402
4322 011772
4323 011772 062702 000001
4324 011776
4325 011776 020201
4326 012000 101010
4327 012002
4328 012002 005003
4329 012004 000401
4330 012006
4331 012006 005203
4332 012010
4333 012010 020327 000100
4334 012014 003001
4335 012016
4336 012016 000773
4337 012020
4338 012020
4339 012020 000764
4340 012022
4341 012022 012603
4342 012024 012602
4343 012026 012601
4344 012030
4345 012030
4346 012030
4347 012030 000207

```

```

;*****
ROUTINE WAIT <TIME>
WAIT:
;* ROUTINE:WAIT
;* THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
;* MAIN PROGRAM FOR A SPECIFIED AMOUNT OF TIME.
;* THIS IS ACCOMPLISHED BY INCREMENTING A
;* REGISTER UP TO A LIMIT. THE INNER LOOP IS SET
;* TO APPROXIMATE 1 MILLI SEC.
;*****
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
MOV R3,-(SP) ;;PUSH R3 ON STACK
LET R1 := TIME(R5)
MOV TIME(R5),R1
INCRU R2 FROM #1 TO R1 BY #1
MOV #1,R2
BR #232
$233: ADD #01,R2
$232: CMP R2,R1
BHI #234
INCR R3 FROM #0 TO #100 BY #1
CLR R3
BR #235
$236: INC R3
$235: CMP R3,#100
BGT #237
ENDINC
BR #236
$237: ENDINC
BR #233
$234: MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
ENDRTN
$230:
$231: RTS PC

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 105  
CVDVAD.P11 12-JUL-84 05:04 ROUTINE - WAIT

```

4348
4349
4350
4351 012032
4352
4353
4354
4355
4356
4357
4358
4359
4360 012032
4361 012032 005267 000002
4362 012036
4363 012036 000002
4364 012040 000000

.SBTTL INTSRV INTERRUPT SERVICE ROUTINE
;*****
INTSRV:
;* SERVICE ROUTINE: INTSRV
;* THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
;* 'INTFLAG' EACH TIME IT IS CALLED. IT ASSUMES
;* THAT THE MAIN CALLING ROUTINE WILL KNOW WHAT
;* TO LOOK FOR.
;*****
;ADD 1 TO 'INTERRUPT OCCURED' FLAG
LET INTFLAG := INTFLAG + #1
INC INTFLAG
ENDSRV
RTI
INTFLAG: 0
;THAT'S ALL

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 106  
 CVDVAD.P11 12-JUL-84 05:04 INTSRV INTERRUPT SERVICE ROUTINE

```

4365
4366 012042
4367 012042
4368
4369 012042 104401 012050
4370 012046 000405
4371
4372 012062
4373 012062 016746 167112
4374 012066 104402
4375 012070 104401 012076
4376 012074 000405
4377
4378 012110
4379 012110 116767 167000 167060
4380 012116 016746 167054
4381 012122 104402
4382 012124 104401 012132
4383 012130 000404
4384
4385 012142
4386 012142 016746 166750
4387 012146 104402
4388 012150 104401 012156
4389 012154 000404
4390
4391 012166
4392 012166 016746 167062
4393 012172 104402
4394 012174 104401 012202
4395 012200 000405
4396
4397 012214
4398 012214 016746 167036
4399 012220 104402
4400 012222
4401 012222
4402 012222
4403 012222 000207

ROUTINE MYTYPE
MYTYPE:
*****
        TYPE      ,65$          ;;TYPE ASCIZ STRING
        BR        64$          ;;GET OVER THE ASCIZ
;;65$: .ASCIZ  <CRLF>*TEST *
64$:
        MOV      $TESTN,-(SP)   ;;SAVE $TESTN FOR TYPEOUT
        TYPOC
        TYPE      ,67$          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        66$          ;;TYPE ASCIZ STRING
        ;;67$: .ASCIZ  *,ERROR *
66$:
        MOVB    $ITMB,$FATAL   ; APT FATAL ERROR NUMBER
        MOV      $FATAL,-(SP)  ;;SAVE $FATAL FOR TYPEOUT
        TYPOC
        TYPE      ,69$          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        68$          ;;TYPE ASCIZ STRING
        ;;69$: .ASCIZ  *,PC = *
68$:
        MOV      $ERRPC,-(SP)  ;;SAVE $ERRPC FOR TYPEOUT
        TYPOC
        TYPE      ,71$          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        70$          ;;TYPE ASCIZ STRING
        ;;71$: .ASCIZ  *,CSR: *
70$:
        MOV      DLADD,-(SP)   ;;SAVE DLADD FOR TYPEOUT
        TYPOC
        TYPE      ,73$          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        72$          ;;TYPE ASCIZ STRING
        ;;73$: .ASCIZ  *,VECTOR: *
72$:
        MOV      DLVEC,-(SP)   ;;SAVE DLVEC FOR TYPEOUT
        TYPOC
        ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
ENDRTN
$240:
$241:
RTS    PC
    
```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 107  
CVDVAD.P11 12-JUL-84 05:04 ROUTINE - MYTYPE

```

4404 012224 ROUTINE CYCLE
4405 012224 CYCLE:
4406 ;:.....
4407 ;* ROUTINE: CYCLE
4408 ;* THIS ROUTINE CAUSES ADRS TO POINT TO THE
4409 ;* ADDRESS OF DLV11-E UNDER TEST, ADRS +2 TO
4410 ;* POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
4411 ;* IT KEEPS TRACK OF THE CURRENT DEVICE AND BIT
4412 ;* MASKS.
4413 ;:.....
4414 012224 REPEAT
4415 012224 $244:
4416 012224 IF BITMASK EQ #0 THEN
4417 012224 005767 000122 TST BITMASK
4418 012230 001027 BNE $245
4419 012232 IF INITFLAG EQ #1 THEN
4420 012232 026727 000116 000001 CMP INITFLAG,#1
4421 012240 001003 BNE $246
4422 012242 LET INITFLAG := #0
4423 012242 005067 000106 CLR INITFLAG
4424 012246 ELSE
4425 012246 000403 BR $247
4426 012250 $246:
4427 012250 CALL $EOP ; AS A SUBROUTINE
4428 012250 004767 000110 JSR PC,$EOP
4429
4430 012254 SPECIALADDRESS: ; BECAUSE $EOP RETURNS AS A JUMP
4431 012254 LET RO := POP
4432 012254 012600 MOV (SP)+,RO
4433 012256 ENDIF
4434 012256 $247:
4435 012256 LET BITMASK := #1
4436 012256 012767 000001 000066 MOV #1,BITMASK
4437 012264 LET $DEVCT := #1
4438 012264 012767 000001 166712 MOV #1,$DEVCT
4439 012272 LET ADDRESS := $BASE
4440 012272 016767 166752 000056 MOV $BASE,ADDRESS
4441 012300 LET VECTOR := $VECT1
4442 012300 016767 166740 000052 MOV $VECT1,VECTOR
4443 012306 ELSE
4444 012306 000410 BR $250
4445 012310 $245:
4446 012310 LET R4 := #10
4447 012310 012704 000010 MOV #10,R4
4448 012314 LET BITMASK := BITMASK ROTATE 1
4449 012314 006167 000032 ROL BITMASK
4450 012320 LET ADDRESS := ADDRESS + R4
4451 012320 060467 000032 ADD R4,ADDRESS
4452 012324 LET VECTOR := VECTOR + R4
4453 012324 060467 000030 ADD R4,VECTOR
4454 012330 ENDIF
4455 012330 $250:
4456 012330 UNTIL BITMASK SETIN $DEVN
4457 012330 036767 000016 166714 BIT BITMASK,$DEVN
4458 012336 001732 BEQ $244
4459

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 108  
CVDVAD.P11 12-JUL-84 05:04 ROUTINE - CYCLE

```

4460 012340          LET ADRS := #ADDRESS
4461 012340 012701 012356      MOV  #ADDRESS,ADRS
4462 012344          LET #DEVCT := #DEVCT + #1
4463 012344 005267 166634      INC  #DEVCT
4464 012350          RETURN
4465 012350 000404          BR   #242
4466 012352 000000      BITMASK: 0
4467 012354 000001      INITFLAG: 1
4468 012356 000000      ADDRESS:  0
4469 012360 000000      VECTOR:   0
4470
4471 012362          ENDRTN
4472 012362          #242:
4473 012362          #243:
4474 012362 000207      RTS   PC
4475
4476

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 109  
 CVDVAD.P11 12-JUL-84 05:04 END OF PASS ROUTINE

```

4477          .SBTTL  END OF PASS ROUTINE
4478
4479          ;:*****
4480          ;:INCREMENT THE PASS NUMBER (#PASS)
4481          ;:INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
4482          ;:TYPE "END PASS @XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
4483          ;:IF THERES A MONITOR GO TO IT
4484          ;:IF THERE ISN'T JUMP TO SPECIALADDRESS
4485
4486          $EOP:
4487          012364 000004          SCOPE
4488          012366 005067 166510  CLR          $TSTNM          ;;ZERO THE TEST NUMBER
4489          012372 005067 166562  CLR          $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
4490          012376 005267 166600  INC          $PASS          ;;INCREMENT THE PASS NUMBER
4491          012402 042767 100000 166572  BIC          $100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
4492          012410 005327          DEC          (PC).          ;;LOOP?
4493          012412 000001          $EOPCT: .WORD 1
4494          012414 003022          BGT          $DOAGN          ;;YES
4495          012416 012737          MOV          (PC).,$(PC).  ;;RESTORE COUNTER
4496          012420 000001          $ENDCT: .WORD 1
4497          012422 012412          $EOPCT
4498          012424 104401 012471  TYPE          .$ENDMG          ;;TYPE "END PASS @"
4499          012430 016746 166546  MOV          $PASS,-(SP)    ;;SAVE $PASS FOR TYPEOUT
4500          012434 104405          TYPDS          ;;GO TYPE--DECIMAL ASCII WITH SIGN
4501          012436 104401 012466  TYPE          .$ENULL          ;;TYPE A NULL CHARACTER
4502          012442 013700 000042  $GET42: MOV          $M42,R0  ;;GET MONITOR ADDRESS
4503          012446 001405          BEQ          $DOAGN          ;;BRANCH IF NO MONITOR
4504          012450 000005          RESET          ;;CLEAR THE WORLD
4505          012452 004710          $ENDAD: JSR          PC,(R0)  ;;GO TO MONITOR
4506          012454 000240          NOP          ;;SAVE ROOM
4507          012456 000240          NOP          ;;FOR
4508          012460 000240          NOP          ;;ACT11
4509          012462          $DOAGN:
4510          012462 000137          JMP          $(PC).          ;;RETURN
4511          012464 012254          $RTNAD: .WORD  SPECIALADDRESS
4512          012466 377 377 000  $ENULL: .BYTE  -1,-1,0      ;;NULL CHARACTER STRING
4513          012471 015 042412 042116  $ENDMG: .ASCIZ  <15><12>/END PASS @/
4514          012476 050040 051501 020123
4515          012504 000043

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 110  
 CVDVAD.P11 12-JUL-84 05:04 POWER DOWN AND UP ROUTINES

```

4516          .SBTTL  POWER DOWN AND UP ROUTINES
4517
4518          ;;*****
4519          ;POWER DOWN ROUTINE
4520 012506 012737 012652 000024 $PWRDN: MOV    @ILLUP,@PWRVEC ;;SET FOR FAST UP
4521 012514 012737 000340 000026      MOV    @340,@PWRVEC+2 ;;PRIO:7
4522 012522 010046          MOV    R0,-(SP) ;;PUSH R0 ON STACK
4523 012524 010146          MOV    R1,-(SP) ;;PUSH R1 ON STACK
4524 012526 010246          MOV    R2,-(SP) ;;PUSH R2 ON STACK
4525 012530 010346          MOV    R3,-(SP) ;;PUSH R3 ON STACK
4526 012532 010446          MOV    R4,-(SP) ;;PUSH R4 ON STACK
4527 012534 010546          MOV    R5,-(SP) ;;PUSH R5 ON STACK
4528 012536 017746 166376      MOV    @SWR,-(SP) ;;PUSH @SWR ON STACK
4529 012542 010667 000110      MOV    SP,$SAVR6 ;;SAVE SP
4530 012546 012737 012560 000024      MOV    @PWRUP,@PWRVEC ;;SET UP VECTOR
4531 012554 000000          HALT
4532 012556 000776          BR     .-2          ;;HANG UP
4533
4534          ;;*****
4535          ;POWER UP ROUTINE
4536 012560 012737 012652 000024 $PWRUP: MOV    @ILLUP,@PWRVEC ;;SET FOR FAST DOWN
4537 012566 016706 000064      MOV    $SAVR6,SP ;;GET SP
4538 012572 005067 000060      CLR    $SAVR6 ;;WAIT LOOP FOR THE TTY
4539 012576 005267 000054      1$: INC    $SAVR6 ;;WAIT FOR THE INC
4540 012602 001375          BNE    1$ ;;OF WORD
4541 012604 012677 166330      MOV    (SP)+,@SWR ;;POP STACK INTO @SWR
4542 012610 012605          MOV    (SP)+,R5 ;;POP STACK INTO R5
4543 012612 012604          MOV    (SP)+,R4 ;;POP STACK INTO R4
4544 012614 012603          MOV    (SP)+,R3 ;;POP STACK INTO R3
4545 012616 012602          MOV    (SP)+,R2 ;;POP STACK INTO R2
4546 012620 012601          MOV    (SP)+,R1 ;;POP STACK INTO R1
4547 012622 012600          MOV    (SP)+,R0 ;;POP STACK INTO R0
4548 012624 012737 012506 000024      MOV    @PWRDN,@PWRVEC ;;SET UP THE POWER DOWN VECTOR
4549 012632 012737 000340 000026      MOV    @340,@PWRVEC+2 ;;PRIO:7
4550 012640 104401          TYPE ;;REPORT THE POWER FAILURE
4551 012642 012660      $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
4552 012644 012716      MOV    (PC)+,(SP) ;;RESTART AT START
4553 012646 001336      $PWRAD: .WORD START ;;RESTART ADDRESS
4554 012650 000002          RTI
4555 012652 000000      $ILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
4556 012654 000775          BR     .-2          ;; BEFORE THE POWER DOWN WAS COMPLETE
4557 012656 000000      $SAVR6: 0 ;;PUT THE SP HERE
4558 012660 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
4559 012666 000122
4560          .EVEN

```

```

4561 .SBTTL TYPE ROUTINE
4562
4563 ;*****
4564 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4565 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4566 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4567 ;*NOTE2: $FILLC CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4568 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4569 ;*
4570 ;*CALL:
4571 ;*1) USING A TRAP INSTRUCTION
4572 ;* TYPE ,MESADR ;:MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4573 ;*OR
4574 ;* TYPE
4575 ;* MESADR
4576 ;*
4577
4578 012670 105767 166263 $TYPE: TSTB $TPFLG ;:IS THERE A TERMINAL?
4579 012674 100002 BPL 1$ ;:BR IF YES
4580 012676 000000 HALT ;:HALT HERE IF NO TERMINAL
4581 012700 000430 BR 3$ ;:LEAVE
4582 012702 010046 1$: MOV RO,-(SP) ;:SAVE RO
4583 012704 017600 000002 MOV @2(SP),RO ;:GET ADDRESS OF ASCIZ STRING
4584 012710 122767 000001 166276 CMPB @APTENV,$ENV ;:RUNNING IN APT MODE
4585 012716 001011 BNE 62$ ;:NO,GO CHECK FOR APT CONSOLE
4586 012720 132767 000100 166267 BITB @APTPOOL,$ENVM ;:SPOOL MESSAGE TO APT
4587 012726 001405 BEQ 62$ ;:NO,GO CHECK FOR CONSOLE
4588 012730 010067 000004 MOV RO,61$ ;:SETUP MESSAGE ADDRESS FOR APT
4589 012734 004767 000774 JSR PC,$ATY3 ;:SPOOL MESSAGE TO APT
4590 012740 000000 61$: .WORD 0 ;:MESSAGE ADDRESS
4591 012742 132767 000040 166245 62$: BITB @APTCSUP,$ENVM ;:APT CONSOLE SUPPRESSED
4592 012750 001003 BNE 60$ ;:YES,SKIP TYPE OUT
4593 012752 112046 2$: MOVB (RO),-(SP) ;:PUSH CHARACTER TO BE TYPED ONTO STACK
4594 012754 001005 BNE 4$ ;:BR IF IT ISN'T THE TERMINATOR
4595 012756 005726 TST (SP)+ ;:IF TERMINATOR POP IT OFF THE STACK
4596 012760 012600 60$: MOV (SP)+,RO ;:RESTORE RO
4597 012762 062716 000002 3$: ADD @2,(SP) ;:ADJUST RETURN PC
4598 012766 000002 RTI ;:RETURN
4599 012770 122716 000011 4$: CMPB @HT,(SP) ;:BRANCH IF <HT>
4600 012774 001430 BEQ 8$
4601 012776 122716 000200 CMPB @CRLF,(SP) ;:BRANCH IF NOT <CRLF>
4602 013002 001006 BNE 5$
4603 013004 005726 TST (SP)+ ;:POP <CR><LF> EQUIV
4604 013006 104401 TYPE ;:TYPE A CR AND LF
4605 013010 001171 $CRLF
4606 013012 105067 000130 CLRB $CHARCNT ;:CLEAR CHARACTER COUNT
4607 013016 000755 BR 2$ ;:GET NEXT CHARACTER
4608 013020 004767 000056 5$: JSR PC,$TYPEC ;:GO TYPE THIS CHARACTER
4609 013024 126726 166126 6$: CMPB $FILLC,(SP)+ ;:IS IT TIME FOR FILLER CHARS.?
4610 013030 001350 BNE 2$ ;:IF NO GO GET NEXT CHAR.
4611 013032 016746 166116 MOV $NULL,-(SP) ;:GET # OF FILLER CHARS. NEEDED
4612 ;:AND THE NULL CHAR.
4613 013036 105366 000001 7$: DECB 1(SP) ;:DOES A NULL NEED TO BE TYPED?
4614 013042 002770 BLT 6$ ;:BR IF NO--GO POP THE NULL OFF OF STACK
4615 013044 004767 000032 JSR PC,$TYPEC ;:GO TYPE A NULL
4616 013050 105367 000072 DECB $CHARCNT ;:DO NOT COUNT AS A COUNT
    
```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 112  
 CVDVAD.P11 12-JUL-84 05:04 TYPE ROUTINE

```

4617 013054 000770          BR      7#          ;;LOOP
4618
4619          ;HORIZONTAL TAB PROCESSOR
4620
4621 013056 112716 000040      8#:     MOVB     #' ,(SP)          ;;REPLACE TAB WITH SPACE
4622 013062 004767 000014      9#:     JSR      PC,$TYPEC          ;;TYPE A SPACE
4623 013066 132767 000007 000052      BITB     #7,$CHARCNT          ;;BRANCH IF NOT AT
4624 013074 001372          BNE     9#          ;;TAB STOP
4625 013076 005726          TST     (SP)+          ;;POP SPACE OFF STACK
4626 013100 000724          BR      2#          ;;GET NEXT CHARACTER
4627 013102 105777 166042      $TYPEC: TSTB     @#TPS          ;;WAIT UNTIL PRINTER IS READY
4628 013106 100375          BPL     $TYPEC
4629 013110 116677 000002 166034      MOVB     2(SP),@#TPB          ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4630 013116 122766 000015 000002      CMPB     @CR,2(SP)          ;;IS CHARACTER A CARRIAGE RETURN?
4631 013124 001003          BNE     1#          ;;BRANCH IF NO
4632 013126 105067 000014          CLRB     $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
4633 013132 000406          BR      $TYPEX          ;;EXIT
4634 013134 122766 000012 000002 1#:     CMPB     @LF,2(SP)          ;;IS CHARACTER A LINE FEED?
4635 013142 001402          BEQ     $TYPEX          ;;BRANCH IF YES
4636 013144 105227          INCB     (PC)+          ;;COUNT THE CHARACTER
4637 013146 000000      $CHARCNT: .WORD 0          ;;CHARACTER COUNT STORAGE
4638 013150 000207      $TYPEX: RTS      PC
4639

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 113  
 CVDVAD.P11 12-JUL-84 05:04 TTY INPUT ROUTINE

```

4640 .SBTTL TTY INPUT ROUTINE
4641
4642 ;*****
4643 .ENABL LSB
4644
4645 ;*****
4646 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4647 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
4648 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
4649 ;*WHEN OPERATING IN TTY FLAG MODE.
4650 013152 022767 000176 165760 $CKSMR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?
4651 013160 001074 BNE 15# ;;BRANCH IF NO
4652 013162 105777 165756 TSTB #TKS ;;CHAR THERE?
4653 013166 100071 BPL 15# ;;IF NO, DON'T WAIT AROUND
4654 013170 117746 165752 MOV #TKB,-(SP) ;;SAVE THE CHAR
4655 013174 042716 177600 BIC #C177,(SP) ;;STRIP-OFF THE ASCII
4656 013200 022726 000007 CMP #7,(SP)+ ;;IS IT A CONTROL G?
4657 013204 001062 BNE 15# ;;NO, RETURN TO USER
4658 013206 126727 165722 000001 CMPB $AUTOB,#1 ;;ARE WE RUNNING IN AUTO-MODE?
4659 013214 001456 BEQ 15# ;;BRANCH IF YES
4660
4661 013216 104401 013677 $GTSMR: TYPE ,#CNTLG ;;ECHO THE CONTROL-G (+G)
4662 013222 104401 013704 TYPE ,#MSWR ;;TYPE CURRENT CONTENTS
4663 013226 016746 164744 MOV SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT
4664 013232 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4665 013234 104401 013715 TYPE ,#MNEW ;;PROMPT FOR NEW SWR
4666 013240 005046 19#: CLR -(SP) ;;CLEAR COUNTER
4667 013242 005046 CLR -(SP) ;;THE NEW SWR
4668 013244 105777 165674 7#: TSTB #TKS ;;CHAR THERE?
4669 013250 100375 BPL 7# ;;IF NOT TRY AGAIN
4670
4671 013252 117746 165670 MOV #TKB,-(SP) ;;PICK UP CHAR
4672 013256 042716 177600 BIC #C177,(SP) ;;MAKE IT 7-BIT ASCII
4673
4674
4675
4676 013262 021627 000025 9#: CMP (SP),#25 ;;IS IT A CONTROL-U?
4677 013266 001005 BNE 10# ;;BRANCH IF NOT
4678 013270 104401 013672 TYPE ,#CNTLU ;;YES, ECHO CONTROL-U (+U)
4679 013274 062706 000006 20#: ADD #6,SP ;;IGNORE PREVIOUS INPUT
4680 013300 000757 BR 19# ;;LET'S TRY IT AGAIN
4681
4682
4683 013302 021627 000015 10#: CMP (SP),#15 ;;IS IT A <CR>?
4684 013306 001022 BNE 16# ;;BRANCH IF NO
4685 013310 005766 000004 TST 4(SP) ;;YES, IS IT THE FIRST CHAR?
4686 013314 001403 BEQ 11# ;;BRANCH IF YES
4687 013316 016677 000002 165614 MOV 2(SP),#SWR ;;SAVE NEW SWR
4688 013324 062706 000006 11#: ADD #6,SP ;;CLEAR UP STACK
4689 013330 104401 001171 14#: TYPE ,#CRLF ;;ECHO <CR> AND <LF>
4690 013334 126727 165575 000001 CMPB $INTAG,#1 ;;RE-ENABLE TTY KBD INTERRUPTS?
4691 013342 001003 BNE 15# ;;BRANCH IF NOT
4692 013344 012777 000100 165572 MOV #100,#TKS ;;RE-ENABLE TTY KBD INTERRUPTS
4693 013352 000002 15#: RTI ;;RETURN
4694 013354 004767 177522 16#: JSR PC,#TYPEC ;;ECHO CHAR
4695 013360 021627 000060 CMP (SP),#60 ;;CHAR < 0?

```



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 114  
 CVDVAD.P11 12-JUL-84 05:04 TTY INPUT ROUTINE

```

4696 013364 002420          BLT      18#          ;;BRANCH IF YES
4697 013366 021627 000067    CMP      (SP),#67      ;;CHAR > 7?
4698 013372 003015          BGT      18#          ;;BRANCH IF YES
4699 013374 042726 000060    BIC      #60,(SP)+    ;;STRIP-OFF ASCII
4700 013400 005766 000002    TST      2(SP)        ;;IS THIS THE FIRST CHAR
4701 013404 001403          BEQ      17#          ;;BRANCH IF YES
4702 013406 006316          ASL      (SP)         ;;NO, SHIFT PRESENT
4703 013410 006316          ASL      (SP)         ;; CHAR OVER TO MAKE
4704 013412 006316          ASL      (SP)         ;; ROOM FOR NEW ONE.
4705 013414 005266 000002    17#:    INC      2(SP)        ;;KEEP COUNT OF CHAR
4706 013420 056616 177776    BIS      -2(SP),(SP)  ;;SET IN NEW CHAR
4707 013424 000707          BR       7#          ;;GET THE NEXT ONE
4708 013426 104401 001170    18#:    TYPE     ,#QUES      ;;TYPE ?<CR><LF>
4709 013432 000720          BR       20#         ;;SIMULATE CONTROL-U
4710
4711
4712
4713          ;;*****
4714          ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4715          ;*CALL:
4716          ;*   RDCHR          ;;INPUT A SINGLE CHARACTER FROM THE TTY
4717          ;*   RETURN HERE   ;;CHARACTER IS ON THE STACK
4718          ;*                ;;WITH PARITY BIT STRIPPED OFF
4719          ;
4720
4721 013434 011646          #RDCHR: MOV      (SP),-(SP)    ;;PUSH DOWN THE PC
4722 013436 016666 000004 000002    MOV      4(SP),2(SP)  ;;SAVE THE PS
4723 013444 105777 165474    1#:    TSTB     #TKS        ;;WAIT FOR
4724 013450 100375          BPL      1#          ;;A CHARACTER
4725 013452 117766 165470 000004    MOVB     #TKB,4(SP)   ;;READ THE TTY
4726 013460 042766 177600 000004    BIC      #C<177>,4(SP) ;;GET RID OF JUNK IF ANY
4727 013466 026627 000004 000023    CMP      4(SP),#23    ;;IS IT A CONTROL-S?
4728 013474 001013          BNE      3#          ;;BRANCH IF NO
4729 013476 105777 165442    2#:    TSTB     #TKS        ;;WAIT FOR A CHARACTER
4730 013502 100375          BPL      2#          ;;LOOP UNTIL ITS THERE
4731 013504 117746 165436    MOVB     #TKB,-(SP)   ;;GET CHARACTER
4732 013510 042716 177600    BIC      #C177,(SP)  ;;MAKE IT 7-BIT ASCII
4733 013514 022627 000021    CMP      (SP)+,#21   ;;IS IT A CONTROL-Q?
4734 013520 001366          BNE      2#          ;;IF NOT DISCARD IT
4735 013522 000750          BR       1#          ;;YES, RESUME
4736 013524 026627 000004 000140    3#:    CMP      4(SP),#140  ;;IS IT UPPER CASE?
4737 013532 002407          BLT      4#          ;;BRANCH IF YES
4738 013534 026627 000004 000175    CMP      4(SP),#175  ;;IS IT A SPECIAL CHAR?
4739 013542 003003          BGT      4#          ;;BRANCH IF YES
4740 013544 042766 000040 000004    BIC      #40,4(SP)   ;;MAKE IT UPPER CASE
4741 013552 000002          4#:    RTI                ;;GO BACK TO USER
4742          ;;*****
4743          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4744          ;*CALL:
4745          ;*   RDLIN          ;;INPUT A STRING FROM THE TTY
4746          ;*   RETURN HERE   ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4747          ;*                ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
4748
4749 013554 010346          #RDLIN: MOV      R3, -(SP)  ;;SAVE R3
4750 013556 012703 013662    1#:    MOV      #TTYIN,R3   ;;GET ADDRESS
4751 013562 022703 013672    2#:    CMP      #TTYIN+8.,R3 ;;BUFFER FULL?

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 115  
 CVDVAD.P11 12-JUL-84 05:04 TTY INPUT ROUTINE

4752	013566	101405				BLOS	48		::BR IF YES
4753	013570	104410				RDCHR			::GO READ ONE CHARACTER FROM THE TTY
4754	013572	112613				MOVB	(SP)+,(R3)		::GET CHARACTER
4755	013574	122713	000177		108:	CMPB	#177,(R3)		::IS IT A RUBOUT
4756	013600	001003				BNE	38		::SKIP IF NOT
4757	013602	104401	001170		48:	TYPE	,#QUES		::TYPE A '?'
4758	013606	000763				BR	18		::CLEAR THE BUFFER AND LOOP
4759	013610	111367	000044		38:	MOVB	(R3),98		::ECHO THE CHARACTER
4760	013614	104401	013660			TYPE	,98		
4761	013620	122723	000015			CMPB	#15,(R3)+		::CHECK FOR RETURN
4762	013624	001356				BNE	28		::LOOP IF NOT RETURN
4763	013626	105063	177777			CLRB	-1(R3)		::CLEAR RETURN (THE 15)
4764	013632	104401	001172			TYPE	,#LF		::TYPE A LINE FEED
4765	013636	012603				MOV	(SP)+,R3		::RESTORE R3
4766	013640	011646				MOV	(SP),-(SP)		::ADJUST THE STACK AND PUT ADDRESS OF THE
4767	013642	016666	000004	000002		MOV	4(SP),2(SP)		::FIRST ASCII CHARACTER ON IT
4768	013650	012766	013662	000004		MOV	#TTYIN,4(SP)		
4769	013656	000002				RTI			::RETURN
4770	013660	000			98:	.BYTE	0		::STORAGE FOR ASCII CHAR. TO TYPE
4771	013661	000				.BYTE	0		::TERMINATOR
4772	013662	000010				#TTYIN: .BLKB	8.		::RESERVE 8 BYTES FOR TTY INPUT
4773	013672	052536	005015	000		#CNTLU: .ASCIZ	/'U/'<15><12>		::CONTROL "U"
4774	013677	136	006507	000012		#CNTLG: .ASCIZ	/'G/'<15><12>		::CONTROL "G"
4775	013704	005015	053523	020122		#MSWR: .ASCIZ	<15><12>/SWR = /		
4776	013712	020075	000						
4777	013715	040	047040	053505		#MNEW: .ASCIZ	/ NEW = /		
4778	013722	036440	000040						

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 116  
 CVDVAD.P11 12-JUL-84 05:04 APT COMMUNICATIONS ROUTINE

```

4779          .SBTTL  APT COMMUNICATIONS ROUTINE
4780
4781          ;;*****
4782 013726 112767 000001 000236 $ATY1:  MOVB  #1,$FFLG      ;;TO REPORT FATAL ERROR
4783 013734 112767 000001 000226 $ATY3:  MOVB  #1,$MFLG      ;;TO TYPE A MESSAGE
4784 013742 000403          BR      $ATYC
4785 013744 112767 000001 000220 $ATY4:  MOVB  #1,$FFLG      ;;TO ONLY REPORT FATAL ERROR
4786 013752          $ATYC:
4787 013752 010046          MOV    R0,-(SP)      ;;PUSH R0 ON STACK
4788 013754 010146          MOV    R1,-(SP)      ;;PUSH R1 ON STACK
4789 013756 105767 000206          TSTB  $MFLG      ;;SHOULD TYPE A MESSAGE?
4790 013762 001450          BEQ    5#          ;;IF NOT: BR
4791 013764 122767 000001 165222  CMPB  @APTENV,$ENV      ;;OPERATING UNDER APT?
4792 013772 001031          BNE   3#          ;;IF NOT: BR
4793 013774 132767 000100 165213  BITB  @APTSPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
4794 014002 001425          BEQ    3#          ;;IF NOT: BR
4795 014004 017600 000004          MOV    @4(SP),R0      ;;GET MESSAGE ADDR.
4796 014010 062766 000002 000004  ADD    @2,4(SP)      ;;BUMP RETURN ADDR.
4797 014016 005767 165152          1#:  TST    $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
4798 014022 001375          BNE   1#          ;;IF NOT: WAIT
4799 014024 010067 165160          MOV    R0,$MSGAD      ;;PUT ADDR IN MAILBOX
4800 014030 105720          2#:  TSTB  (R0)+      ;;FIND END OF MESSAGE
4801 014032 001376          BNE   2#
4802 014034 166700 165150          SUB    $MSGAD,R0      ;;SUB START OF MESSAGE
4803 014040 006200          ASR    R0          ;;GET MESSAGE LNGTH IN WORDS
4804 014042 010067 165144          MOV    R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
4805 014046 012767 000004 165120  MOV    @4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
4806 014054 000413          BR     5#
4807 014056 017667 000004 000016 3#:  MOV    @4(SP),4#      ;;PUT MSG ADDR IN JSR LINKAGE
4808 014064 062766 000002 000004  ADD    @2,4(SP)      ;;BUMP RETURN ADDRESS
4809 014072 016746 163700          MOV    177776,-(SP)  ;;PUSH 177776 ON STACK
4810 014076 004767 176566          JSR   PC,$TYPE      ;;CALL TYPE MACRO
4811 014102 000000          4#:  .WORD  0
4812 014104          5#:
4813 014104 105767 000062          10#: TSTB  $FFLG      ;;SHOULD REPORT FATAL ERROR?
4814 014110 001416          BEQ   12#      ;;IF NOT: BR
4815 014112 005767 165076          TST   $ENV      ;;RUNNING UNDER APT?
4816 014116 001413          BEQ   12#      ;;IF NOT: BR
4817 014120 005767 165050          11#: TST   $MSGTYPE      ;;FINISHED LAST MESSAGE?
4818 014124 001375          BNE   11#      ;;IF NOT: WAIT
4819 014126 017667 000004 165042  MOV    @4(SP),$FATAL  ;;GET ERROR #
4820 014134 062766 000002 000004  ADD    @2,4(SP)      ;;BUMP RETURN ADDR.
4821 014142 005267 165026          INC   $MSGTYPE      ;;TELL APT TO TAKE ERROR
4822 014146 105067 000020          12#: CLRB  $FFLG      ;;CLEAR FATAL FLAG
4823 014152 105067 000013          CLRB  $LFLG      ;;CLEAR LOG FLAG
4824 014156 105067 000006          CLRB  $MFLG      ;;CLEAR MESSAGE FLAG
4825 014162 012601          MOV   (SP)+,R1      ;;POP STACK INTO R1
4826 014164 012600          MOV   (SP)+,R0      ;;POP STACK INTO R0

```

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 117  
CVDVAD.P11      12-JUL-84 05:04      APT COMMUNICATIONS ROUTINE

4827	014166	000207				
4828	014170	000				
4829	014171	000				
4830	014172	000				
4831		014174				
4832		000200				
4833		000001				
4834		000100				
4835		000040				

  

			RTS	PC		::RETURN
			\$MFLG:	.BYTE	0	::MESSG. FLAG
			\$LFLG:	.BYTE	0	::LOG FLAG
			\$FFLG:	.BYTE	0	::FATAL FLAG
				.EVEN		
			APTSIZE=	200		
			APTENV=	001		
			APTPOOL=	100		
			APTCSUP=	040		

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 118  
 CVDVAD.P11 12-JUL-84 05:04 ERROR HANDLER ROUTINE

```

4836      .SBTTL  ERROR HANDLER ROUTINE
4837
4838      ;;*****
4839      ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
4840      ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4841      ;*AND GO TO MYTYPE ON ERROR
4842      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4843      ;*SW15=1      HALT ON ERROR
4844      ;*SW13=1      INHIBIT ERROR TYPEOUTS
4845      ;*SW10=1      BELL ON ERROR
4846      ;*SW09=1      LOOP ON ERROR
4847      ;*CALL
4848      ;*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
4849
4850      $ERROR:
4851      014174      104407      CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
4852      014176      105267      164701      7$:      INCB      $ERFLG      ;;SET THE ERROR FLAG
4853      014202      001775      BEQ      7$      ;;DON'T LET THE FLAG GO TO ZERO
4854      014204      016777      164672      164730      MOV      $TSTNM,$DISPLAY      ;;DISPLAY TEST NUMBER AND ERROR FLAG
4855      014212      032777      002000      164720      BIT      @BIT10,$SWR      ;;BELL ON ERROR?
4856      014220      001402      BEQ      1$      ;;NO - SKIP
4857      014222      104401      001164      TYPE      , $BELL      ;;RING BELL
4858      014226      005267      164660      1$:      INC      $ERTTL      ;;COUNT THE NUMBER OF ERRORS
4859      014232      011667      164660      MOV      (SP),$ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
4860      014236      162767      000002      164652      SUB      @2,$ERRPC
4861      014244      117767      164646      164642      MOV      @ $ERRPC,$ITEMB      ;;STRIP AND SAVE THE ERROR ITEM CODE
4862      014252      032777      020000      164660      BIT      @BIT13,$SWR      ;;SKIP TYPEOUT IF SET
4863      014260      001004      BNE      20$      ;;SKIP TYPEOUTS
4864      014262      004767      175554      JSR      PC,MYTYPE      ;;GO TO USER ERROR ROUTINE
4865      014266      104401      001171      TYPE      , $CRLF
4866      014272
4867      014272      122767      000001      164714      20$:      CMP      @APTENV,$ENV      ;;RUNNING IN APT MODE
4868      014300      001007      BNE      2$      ;;NO,SKIP APT ERROR REPORT
4869      014302      116767      164606      000004      MOV      $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
4870      014310      004767      177430      JSR      PC,$ATY4      ;;REPORT FATAL ERROR TO APT
4871      014314      000      21$:      .BYTE      0
4872      014315      000      .BYTE      0
4873      014316      000777      22$:      BR      22$      ;;APT ERROR LOOP
4874      014320      005777      164614      2$:      TST      $SWR      ;;HALT ON ERROR
4875      014324      100002      BPL      3$      ;;SKIP IF CONTINUE
4876      014326      000000      HALT      ;;HALT ON ERROR!
4877      014330      104407      CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
4878      014332      032777      001000      164600      3$:      BIT      @BIT09,$SWR      ;;LOOP ON ERROR SWITCH SET?
4879      014340      001402      BEQ      4$      ;;BR IF NO
4880      014342      016716      164542      MOV      $LPERR,(SP)      ;;FUDGE RETURN FOR LOOPING
4881      014346      005767      164610      4$:      TST      $ESCAPE      ;;CHECK FOR AN ESCAPE ADDRESS
4882      014352      001402      BEQ      5$      ;;BR IF NONE
4883      014354      016716      164602      MOV      $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE
4884      014360      5$:
4885      014360      022737      012452      000042      CMP      @ $ENDAD,$@42      ;;ACT-11 AUTO-ACCEPT?
4886      014366      001001      BNE      6$      ;;BRANCH IF NO
4887      014370      000000      HALT      ;;YES
4888      014372
4889      014372      000002      6$:      RTI      ;;RETURN

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 119  
 CVDVAD.P11 12-JUL-84 05:04 SCOPE HANDLER ROUTINE

```

4890      .SBTTL  SCOPE HANDLER ROUTINE
4891
4892      ;*****
4893      ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4894      ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4895      ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4896      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4897      ;*SW14=1      LOOP ON TEST
4898      ;*SW11=1      INHIBIT ITERATIONS
4899      ;*SW09=1      LOOP ON ERROR
4900      ;*SW08=1      LOOP ON TEST IN SWR<7:0>
4901      ;*CALL
4902      ;*      SCOPE      ;;SCOPE=IOT
4903
4904      $SCOPE:
4905      014374 104407      CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
4906      014376 032777 040000 164534 1$:      BIT      @BIT14,@SWR      ;;LOOP ON PRESENT TEST?
4907      014404 001114      BNE      $OVER      ;;YES IF SW14=1
4908      ;*****START OF CODE FOR THE XOR TESTER*****
4909      014406 000416      $XTSTR: BR      6$      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
4910      ;THIS INSTRUCTION TO A "NOP" (NOP=240)
4911      014410 013746 000004      MOV      @ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
4912      014414 012737 014434 000004      MOV      @5,@ERRVEC      ;;SET FOR TIMEOUT
4913      014422 005737 177060      TST      @177060      ;;TIME OUT ON XOR?
4914      014426 012637 000004      MOV      (SP)+,@ERRVEC      ;;RESTORE THE ERROR VECTOR
4915      014432 000463      BR      $SVLAD      ;;GO TO THE NEXT TEST
4916      014434 022626      5$:      CMP      (SP)+,(SP)+      ;;CLEAR THE STACK AFTER A TIME OUT
4917      014436 012637 000004      MOV      (SP)+,@ERRVEC      ;;RESTORE THE ERROR VECTOR
4918      014442 000423      BR      7$      ;;LOOP ON THE PRESENT TEST
4919      014444      6$:;*****END OF CODE FOR THE XOR TESTER*****
4920      014444 032777 000400 164466      BIT      @BIT08,@SWR      ;;LOOP ON SPEC. TEST?
4921      014452 001404      BEQ      2$      ;;BR IF NO
4922      014454 127767 164460 164420      CMPB     @SWR,$TSTNM      ;;ON THE RIGHT TEST? SWR<7:0>
4923      014462 001465      BEQ      $OVER      ;;BR IF YES
4924      014464 105767 164413      2$:      TSTB     $ERFLG      ;;HAS AN ERROR OCCURRED?
4925      014470 001421      BEQ      3$      ;;BR IF NO
4926      014472 126767 164417 164403      CMPB     $ERMAX,$ERFLG      ;;MAX. ERRORS FOR THIS TEST OCCURRED?
4927      014500 101015      BHI      3$      ;;BR IF NO
4928      014502 032777 001000 164430      BIT      @BIT09,@SWR      ;;LOOP ON ERROR?
4929      014510 001404      BEQ      4$      ;;BR IF NO
4930      014512 016767 164372 164366      7$:      MOV      $LPERR,$LPADR      ;;SET LOOP ADDRESS TO LAST SCOPE
4931      014520 000446      BR      $OVER
4932      014522 105067 164355      4$:      CLRB     $ERFLG      ;;ZERO THE ERROR FLAG
4933      014526 005067 164426      CLR      $TIMES      ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
4934      014532 000415      BR      1$      ;;ESCAPE TO THE NEXT TEST
4935      014534 032777 004000 164376      3$:      BIT      @BIT11,@SWR      ;;INHIBIT ITERATIONS?
4936      014542 001011      BNE      1$      ;;BR IF YES
4937      014544 005767 164432      TST      $PASS      ;;IF FIRST PASS OF PROGRAM
4938      014550 001406      BEQ      1$      ;;      INHIBIT ITERATIONS
4939      014552 005267 164326      INC      $ICNT      ;;INCREMENT ITERATION COUNT
4940      014556 026767 164376 164320      CMP      $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
4941      014564 002024      BGE      $OVER      ;;BR IF MORE ITERATION REQUIRED
4942      014566 012767 000001 164310      1$:      MOV      @1,$ICNT      ;;REINITIALIZE THE ITERATION COUNTER
4943      014574 016767 000052 164356      MOV      $MXCNT,$TIMES      ;;SET NUMBER OF ITERATIONS TO DO
4944      014602 105267 164274      $SVLAD: INCB     $TSTNM      ;;COUNT TEST NUMBERS
4945      014606 116767 164270 164364      MOVB     $TSTNM,$TESTN      ;;SET TEST NUMBER IN APT MAILBOX
    
```

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 120  
 CVDVAD.P11      12-JUL-84 05:04      SCOPE HANDLER ROUTINE

4946	014614	011667	164266		MOV	(SP),#LPADR	::SAVE SCOPE LOOP ADDRESS
4947	014620	011667	164264		MOV	(SP),#LPERR	::SAVE ERROR LOOP ADDRESS
4948	014624	005067	164332		CLR	#ESCAPE	::CLEAR THE ESCAPE FROM ERROR ADDRESS
4949	014630	112767	000001	164257	MOVB	#1,#ERMAX	::ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4950	014636	016777	164240	164276	#OVER: MOV	#TSTNM,#DISPLAY	::DISPLAY TEST NUMBER
4951	014644	016716	164236		MOV	#LPADR,(SP)	::FUDGE RETURN ADDRESS
4952	014650	000002			RTI		::FIXES PS
4953	014652	003720			#MXCNT:	2000.	::MAX. NUMBER OF ITERATIONS

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 121  
 CVDVAD.P11 12-JUL-84 05:04 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966 014654
4967 014654 010046
4968 014656 010146
4969 014660 010246
4970 014662 010346
4971 014664 010546
4972 014666 012746 020200
4973 014672 016605 000020
4974 014676 100004
4975 014700 005405
4976 014702 112766 000055 000001
4977 014710 005000
4978 014712 012703 015070
4979 014716 112723 000040
4980 014722 005002
4981 014724 016001 015060
4982 014730 160105
4983 014732 002402
4984 014734 005202
4985 014736 000774
4986 014740 060105
4987 014742 005702
4988 014744 001002
4989 014746 105716
4990 014750 100407
4991 014752 106316
4992 014754 103003
4993 014756 116663 000001 177777
4994 014764 052702 000060
4995 014770 052702 000040
4996 014774 110223
4997 014776 005720
4998 015000 020027 000010
4999 015004 002746
5000 015006 003002
5001 015010 010502
5002 015012 000764
5003 015014 105726
5004 015016 100003
5005 015020 116663 177777 177776
5006 015026 105013
5007 015030 012605
5008 015032 012603
5009 015034 012602
    
```

;:\*\*\*\*\*  
 ;:THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT  
 ;:SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE  
 ;:NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED  
 ;:BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE  
 ;:REPLACED WITH SPACES.  
 ;:CALL:  
 ;: MOV NUM,-(SP) ;:PUT THE BINARY NUMBER ON THE STACK  
 ;: TYPDS ;:GO TO THE ROUTINE  
 ;:TYPDS:  
 MOV R0,-(SP) ;:PUSH R0 ON STACK  
 MOV R1,-(SP) ;:PUSH R1 ON STACK  
 MOV R2,-(SP) ;:PUSH R2 ON STACK  
 MOV R3,-(SP) ;:PUSH R3 ON STACK  
 MOV R5,-(SP) ;:PUSH R5 ON STACK  
 MOV #20200,-(SP) ;:SET BLANK SWITCH AND SIGN  
 MOV 20(SP),R5 ;:GET THE INPUT NUMBER  
 BPL 1# ;:BR IF INPUT IS POS.  
 NEG R5 ;:MAKE THE BINARY NUMBER POS.  
 MOVB #'-.1(SP) ;:MAKE THE ASCII NUMBER NEG.  
 CLR R0 ;:ZERO THE CONSTANTS INDEX  
 MOV #DBLK,R3 ;:SETUP THE OUTPUT POINTER  
 MOVB #' ,(R3) ;:SET THE FIRST CHARACTER TO A BLANK  
 CLR R2 ;:CLEAR THE BCD NUMBER  
 MOV #DTBL(R0),R1 ;:GET THE CONSTANT  
 SUB R1,R5 ;:FORM THIS BCD DIGIT  
 BLT 4# ;:BR IF DONE  
 INC R2 ;:INCREASE THE BCD DIGIT BY 1  
 BR 3#  
 ADD R1,R5 ;:ADD BACK THE CONSTANT  
 TST R2 ;:CHECK IF BCD DIGIT=0  
 BNE 5# ;:FALL THROUGH IF 0  
 TSTB (SP) ;:STILL DOING LEADING 0'S?  
 BMI 7# ;:BR IF YES  
 ASLB (SP) ;:MSD?  
 BCC 6# ;:BR IF NO  
 MOVB 1(SP),-1(R3) ;:YES--SET THE SIGN  
 BIS #'0,R2 ;:MAKE THE BCD DIGIT ASCII  
 BIS #' ,R2 ;:MAKE IT A SPACE IF NOT ALREADY A DIGIT  
 MOVB R2,(R3) ;:PUT THIS CHARACTER IN THE OUTPUT BUFFER  
 TST (R0) ;:JUST INCREMENTING  
 CMP R0,#10 ;:CHECK THE TABLE INDEX  
 BLT 2# ;:GO DO THE NEXT DIGIT  
 BGT 8# ;:GO TO EXIT  
 MOV R5,R2 ;:GET THE LSD  
 BR 6# ;:GO CHANGE TO ASCII  
 TSTB (SP) ;:WAS THE LSD THE FIRST NON-ZERO?  
 BPL 9# ;:BR IF NO  
 MOVB -1(SP),-2(R3) ;:YES--SET THE SIGN FOR TYPING  
 CLRB (R3) ;:SET THE TERMINATOR  
 MOV (SP),R5 ;:POP STACK INTO R5  
 MOV (SP),R3 ;:POP STACK INTO R3  
 MOV (SP),R2 ;:POP STACK INTO R2



MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 122  
 CVDVAD.P11      12-JUL-84 05:04      CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

5010	015036	012601			MOV	(SP),R1	;;POP STACK INTO R1
5011	015040	012600			MOV	(SP),R0	;;POP STACK INTO R0
5012	015042	104401	015070		TYPE	,#DBLK	;;NOW TYPE THE NUMBER
5013	015046	016666	000002	000004	MOV	2(SP),4(SP)	;;ADJUST THE STACK
5014	015054	012616			MOV	(SP), (SP)	
5015	015056	000002			RTI		;;RETURN TO USER
5016	015060	023420			#DTBL:	10000.	
5017	015062	001750				1000.	
5018	015064	000144				100.	
5019	015066	000012				10.	
5020	015070	000004			#DBLK:	.BLKW 4	

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 123  
 CVDVAD.P11 12-JUL-84 05:04 BINARY TO OCTAL (ASCII) AND TYPE

```

5021 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
5022
5023 ;*****
5024 ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5025 ;OCTAL (ASCII) NUMBER AND TYPE IT.
5026 ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5027 ;CALL:
5028 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5029 ;*      TYPOS    ;;CALL FOR TYPEOUT
5030 ;*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5031 ;*      .BYTE   M              ;;M=1 OR 0
5032 ;*                                     ;;1=TYPE LEADING ZEROS
5033 ;*                                     ;;0=SUPPRESS LEADING ZEROS
5034 ;*
5035 ;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5036 ;$TYPOS OR $TYPOC
5037 ;CALL:
5038 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5039 ;*      TYPON    ;;CALL FOR TYPEOUT
5040 ;*
5041 ;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5042 ;CALL:
5043 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5044 ;*      TYPOC    ;;CALL FOR TYPEOUT
5045
5046 015100 017646 000000          $TYPOS: MOV      8(SP),-(SP)      ;;PICKUP THE MODE
5047 015104 116667 000001 000211  MOVB     1(SP),#OFILL      ;;LOAD ZERO FILL SWITCH
5048 015112 112667 000207          MOVB     (SP),#OMODE+1    ;;NUMBER OF DIGITS TO TYPE
5049 015116 062716 000002          ADD      #2,(SP)        ;;ADJUST RETURN ADDRESS
5050 015122 000406          BR       $TYPON
5051 015124 112767 000001 000171  $TYPOC: MOVB     #1,#OFILL      ;;SET THE ZERO FILL SWITCH
5052 015132 112767 000006 000165  MOVB     #6,#OMODE+1    ;;SET FOR SIX(6) DIGITS
5053 015140 112767 000005 000154  $TYPON: MOVB     #5,#OCNT      ;;SET THE ITERATION COUNT
5054 015146 010346          MOV      R3,-(SP)      ;;SAVE R3
5055 015150 010446          MOV      R4,-(SP)      ;;SAVE R4
5056 015152 010546          MOV      R5,-(SP)      ;;SAVE R5
5057 015154 116704 000145          MOVB     #OMODE+1,R4    ;;GET THE NUMBER OF DIGITS TO TYPE
5058 015160 005404          NEG      R4
5059 015162 062704 000006          ADD      #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
5060 015166 110467 000132          MOVB     R4,#OMODE      ;;SAVE IT FOR USE
5061 015172 116704 000125          MOVB     #OFILL,R4      ;;GET THE ZERO FILL SWITCH
5062 015176 016605 000012          MOV      12(SP),R5     ;;PICKUP THE INPUT NUMBER
5063 015202 005003          CLR      R3            ;;CLEAR THE OUTPUT WORD
5064 015204 006105          1$:  ROL      R5        ;;ROTATE MSB INTO "C"
5065 015206 000404          BR       3$           ;;GO DO MSB
5066 015210 006105          2$:  ROL      R5        ;;FORM THIS DIGIT
5067 015212 006105          ROL      R5
5068 015214 006105          ROL      R5
5069 015216 010503          MOV      R5,R3
5070 015220 006103          3$:  ROL      R3        ;;GET LSB OF THIS DIGIT
5071 015222 105367 000076          DECB     #OMODE        ;;TYPE THIS DIGIT?
5072 015226 100016          BPL      7$           ;;BR IF NO
5073 015230 042703 177770          BIC      #177770,R3    ;;GET RID OF JUNK
5074 015234 001002          BNE      4$           ;;TEST FOR 0
5075 015236 005704          TST      R4           ;;SUPPRESS THIS 0?
5076 015240 001403          BEQ      5$           ;;BR IF YES

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 124  
 CVDVAD.P11 12-JUL-84 05:04 BINARY TO OCTAL (ASCII) AND TYPE

5077	015242	005204		4:	INC	R4		::DON'T SUPPRESS ANYMORE 0'S
5078	015244	052703	000060		BIS	#'0,R3		::MAKE THIS DIGIT ASCII
5079	015250	052703	000040	5:	BIS	#',R3		::MAKE ASCII IF NOT ALREADY
5080	015254	110367	000040		MOVB	R3,8		::SAVE FOR TYPING
5081	015260	104401	015320		TYPE	,8		::GO TYPE THIS DIGIT
5082	015264	105367	000032	7:	DECB	%CNT		::COUNT BY 1
5083	015270	003347			BGT	2		::BR IF MORE TO DO
5084	015272	002402			BLT	6		::BR IF DONE
5085	015274	005204			INC	R4		::INSURE LAST DIGIT ISN'T A BLANK
5086	015276	000744			BR	2		::GO DO THE LAST DIGIT
5087	015300	012605		6:	MOV	(SP)+,R5		::RESTORE R5
5088	015302	012604			MOV	(SP)+,R4		::RESTORE R4
5089	015304	012603			MOV	(SP)+,R3		::RESTORE R3
5090	015306	016666	000002 000004		MOV	2(SP),4(SP)		::SET THE STACK FOR RETURNING
5091	015314	012616			MOV	(SP)+,(SP)		
5092	015316	000002			FTI			::RETURN
5093	015320	000		8:	.BYTE	0		::STORAGE FOR ASCII DIGIT
5094	015321	000			.BYTE	0		::TERMINATOR FOR TYPE ROUTINE
5095	015322	000		%CNT:	.BYTE	0		::OCTAL DIGIT COUNTER
5096	015323	000		%FILL:	.BYTE	0		::ZERO FILL SWITCH
5097	015324	000000		%MODE:	.WORD	0		::NUMBER OF DIGITS TO TYPE

MAINDEC-ZZ-CVDVA-D      MACY11 30A(1052) 12-JUL-84 09:52 PAGE 125  
 CVDVAD.P11      12-JUL-84 05:04      TRAP DECODER

```

5098                    .SBTTL TRAP DECODER
5099
5100                    ;*****
5101                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5102                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5103                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5104                    ;*GO TO THAT ROUTINE.
5105
5106                    $TRAP: MOV      RO,-(SP)                    ;;SAVE RO
5107                         MOV      2(SP),RO                    ;;GET TRAP ADDRESS
5108                         TST      -(RO)                    ;;BACKUP BY 2
5109                         MOVB    (RO),RO                    ;;GET RIGHT BYTE OF TRAP
5110                         ASL      RO                    ;;POSITION FOR INDEXING
5111                         MOV      $TRPAD(RO),RO                    ;;INDEX TO TABLE
5112                         RTS      RO                    ;;GO TO ROUTINE
5113
5114
5115                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
5116
5117                    $TRAP2: MOV    (SP),-(SP)                    ;;MOVE THE PC DOWN
5118                         MOV    4(SP),2(SP)                    ;;MOVE THE PSW DOWN
5119                         RTI                    ;;RESTORE THE PSW
5120
5121                    .SBTTL TRAP TABLE
5122
5123                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
5124                    ;*BY THE "TRAP" INSTRUCTION.
5125
5126                    ;                    ROUTINE
5127                    ;                    -----
5128                    $TRPAD: .WORD    $TRAP2                    TRAP+1(104401) TTY TYPEOUT ROUTINE
5129                                       $TYPE                    ;;CALL=TYPE                    TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
5130                                       $TYPOC                    ;;CALL=TYPOC                    TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
5131                                       $TYPOS                    ;;CALL=TYPOS                    TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
5132                                       $TYPON                    ;;CALL=TYPON                    TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
5133                                       $TYPDS                    ;;CALL=TYPDS
5134
5135                                       $GTSWR                    ;;CALL=GTSWR                    TRAP+6(104406) GET SOFT-SWR SETTING
5136
5137                                       $CKSWR                    ;;CALL=CKSWR                    TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
5138                                       $RDCHR                    ;;CALL=RDCHR                    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
5139                                       $RDLIN                    ;;CALL=RDLIN                    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
5140                                       .END

```

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 127  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- JCR SYMBOLS

ABASE = 175610	10	1017	1058					
ACDW1 = 000000	1017							
ACDW2 = 000000	1017							
ACPUOP = 000000	1017	1032						
ADDRES 012356	4440*	4451*	4461	44680				
ADDW0 = 000000	1017							
ADDW1 = 000000	1017							
ADDW10 = 000000	1017							
ADDW11 = 000000	1017							
ADDW12 = 000000	1017							
ADDW13 = 000000	1017							
ADDW14 = 000000	1017							
ADDW15 = 000000	1017							
ADDW2 = 000000	1017							
ADDW3 = 000000	1017							
ADDW4 = 000000	1017							
ADDW5 = 000000	1017							
ADDW6 = 000000	1017							
ADDW7 = 000000	1017							
ADDW8 = 000000	1017							
ADDW9 = 000000	1017							
ADEVCT = 000000	1017	1023						
ADEVN = 000001	10	1017	1059					
AENV = 000000	1017	1028						
AENVN = 000000	1017	1029						
AFATAL = 000000	1017	1020						
AMADR1 = 000000	1017	1045						
AMADR2 = 000000	1017	1049						
AMADR3 = 000000	1017	1052						
AMADR4 = 000000	1017	1055						
AMAMS1 = 000000	1017	1039						
AMAMS2 = 000000	1017	1047						
AMAMS3 = 000000	1017	1050						
AMAMS4 = 000000	1017	1053						
AMSGAD = 000000	1017	1025						
AMSGLG = 000000	1017	1026						
AMSGTY = 000000	1017	1019						
AMTYP1 = 000000	1017	1040						
AMTYP2 = 000000	1017	1048						
AMTYP3 = 000000	1017	1051						
AMTYP4 = 000000	1017	1054						
APASS = 000000	1017	1022						
APRIOR = 000000	1017							
APTCSU = 000040	4591	48350						
APTENV = 000001	2697	2906	3183	3847	4584	4791	48330	4867
APTSIZ = 000200	1128	48320						
APTSPO = 000100	4586	4793	48340					
ASWREG = 000000	1017	1030						
ATESTN = 000000	1017	1021						
AUNIT = 000000	1017	1024						
AUSMR = 071110	10	1017	1031					
AVECT1 = 000300	10	1017	1056					
AVECT2 = 000000	1017	1057						
BADBRK 011326	41140							
BAUD = 007400	9210							
BAUDRA 007210	33200							



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 129  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

EVENOD=	000040	9150												
FALSE =	000000	41720	4179											
FLAG =	011644	4179*	4206*	4211	4234	42460								
FRERR =	020000	8520												
FRFD =	040000	9240	1507	1574	2050	2141	2230	2410	2510					
GNS =	***** U	932	1151	4128	4134	4140	4371	4377	4384	4390	4396	5129	5130	5131
		5172	5133	5135	5137	5138	5139							
GTSWR =	104406	1146	51350											
HOLD =	011072	3968*	3969*	3972	39850									
HOLDSC =	011645	4191*	4196*	4202	42470									
HOMLON=	000000	41510	4177											
HT =	000011	7090	4599	4640										
I =	001272	10850	1205*	1226*	1229	1231	4280*	4283*	4285					
ILLMEM=	000004	8110	1200	1236										
INITFL =	012354	1154*	4420	4423*	44670									
INTFLA =	012040	1212*	1217	3346*	3386	3390	3412*	3428	3480*	3522	3526	3570*	3596	3600
		4361*	43640											
INTSRV =	012032	1201	3358	3473	43510									
IOTVEC=	000020	8040	1097*	1098*										
LF =	000012	7100	4634	4640										
LOOP =	001744	11550	4145											
MAINT =	000004	8840	1348	1361	1364	1378	1381	1395	1400	2789	2848	2914	3196	3483
		3544	3645	3729	3752	3911	4073							
MASK =	000000	42580	4272*	4288*	4290*	4295*								
MAX =	006422	2942	2946	30210										
MYTYPE =	012042	43670	4864											
NEW =	007242	3219*	3221*	3237*	3239*	3248	3256	3258	3276	3278	33260			
NUMBER =	011030	3925	39500	3975	4033									
NUMBR =	011746	4274*	4275	4277*	4285	42980								
OLD =	007246	3192*	3194*	3248	3256	3258	3276*	3278*	33270					
ORERR =	040000	8510	3066	3105	3133									
PARITY=	000020	9140												
PBAUDS=	004000	8740												
PBAUD0=	010000	8730												
PBAUD1=	020000	8720												
PBAUD2=	040000	8710												
PBAUD3=	100000	8700												
PBR =	000200	9170	3174											
PERR =	010000	8530												
PIRQ =	177772	7160												
PIRQVE=	000240	8100												
PRO =	000000	7330	3368	3417	3489	3576	3649	3755	3874					
PR1 =	000040	7340												
PR2 =	000100	7350												
PR3 =	000140	7360												
PR4 =	000200	7370												
PR5 =	000240	7380												
PR6 =	000300	7390												
PR7 =	000340	7400	1202	1420	2421	2521	2624	3359	3474	3885	3891			
PS =	177776	7130	714											
PSW =	177776	7140												
PWRVEC=	000024	8050	1103*	1104*	4520*	4521*	4530*	4536*	4548*	4549*				
RATES =	007170	3210	3289	32940										
RBUF =	001262	10810	1171*	1172*	2883	2953	3008	3062	3105	3133	3148	3207	3518	3660
		3698	3765	3802	4003	4077	4103							
RCSR =	001260	10800	1169*	1518*	1521	1535*	1537	1550*	1553	1587	1600*	1603	1617*	1620









MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 133  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

\$AUTOB	001134	994#	1148*	4658	4779									
\$BASE	001250	1058#	4440											
\$BDADR	001122	989#												
\$BDDAT	001126	991#												
\$BELL	001164	1009#	4857	4890										
\$BGNLE=	177777	1#												
\$CHARC	013146	4606*	4616*	4623	4632*	4637#								
\$CKSMR	013152	4650#	5137											
\$CMTAG	001100	977#	1090	1091	1099	1105	1106	1107						
\$CM3 =	000000	1007#												
\$CNTLG	013677	4661	4774#											
\$CNTLU	013672	4678	4773#											
\$CPUOP	001222	1032#												
\$CRLF	001171	1011#	4605	4640	4689	4773	4865	4890						
\$DBLK	015070	4978	5012	5020#										
\$DEVCT	001204	1023#	4438*	4463*										
\$DEVH	001252	1059#	4457											
\$DOAGN	012462	4494	4503	4509#										
\$DTBL	015060	4981	5016#											
\$ENDAD	012452	943	1136	4505#	4885									
\$ENDCT	012420	1105	4496#											
\$ENDMG	012471	4498	4513#											
\$ENULL	012466	4501	4512#											
\$ENV	001214	1028#	1142	2697	2906	3183	3847	4584	4791	4815	4867			
\$ENVH	001215	1029#	1128	4586	4591	4793								
\$EOP	012364	4428	4486#											
\$EOPCT	012412	1105*	4493#	4497										
\$ERFLG	001103	980#	4852*	4890	4895	4924	4926	4932*	4954					
\$ERMAX	001115	986#	1108*	4926	4949*	4954								
\$ERRFL=	000000	1154#	1155	1157#	1161#	1162	1164#	1165	1166#	1167	1169#	1170	1171#	1173
		1174#	1176	1177#	1179	1180#	1182	1183#	1184	1196#	1197	1199#	1200#	1201#
		1202#	1203#	1204	1205#	1206	1209#	1210	1212#	1213	1226#	1227	1228#	1230
		1236#	1237#	1239#	1240#	1241	1263#	1264	1269#	1270	1283#	1284	1285#	1286
		1299#	1300	1302#	1303	1316#	1317	1319#	1320	1345#	1346	1359#	1360	1361#
		1362	1375#	1376	1378#	1379	1392#	1393	1395#	1396	1427#	1428	1441#	1442
		1443#	1444	1457#	1458	1460#	1461	1474#	1475	1477#	1478	1510#	1511	1516#
		1517	1518#	1519	1532#	1533	1535#	1536	1548#	1549	1550#	1551	1577#	1578
		1584#	1585	1598#	1599	1600#	1601	1614#	1615	1617#	1618	1631#	1632	1634#
		1635	1660#	1661	1674#	1675	1676#	1677	1690#	1691	1693#	1694	1706#	1707
		1710#	1711	1736#	1737	1750#	1751	1752#	1753	1766#	1767	1769#	1770	1783#
		1784	1786#	1787	1812#	1813	1826#	1827	1828#	1829	1842#	1843	1845#	1846
		1859#	1860	1862#	1863	1899#	1900	1931#	1932	1964#	1965	2000#	2001	2008#
		2009	2056#	2057	2068#	2069	2072#	2073	2088#	2089	2092#	2093	2108#	2109
		2112#	2113	2145#	2146	2157#	2158	2161#	2162	2177#	2178	2181#	2182	2197#
		2198	2201#	2202	2235#	2236	2247#	2248	2251#	2252	2267#	2268	2271#	2272
		2287#	2288	2291#	2292	2323#	2324	2335#	2336	2339#	2340	2355#	2356	2359#
		2360	2375#	2376	2379#	2380	2415#	2416	2428#	2429	2431#	2432	2434#	2440#
		2441	2455#	2456	2459#	2460	2481#	2482	2485#	2486	2515#	2516	2528#	2529
		2532#	2533	2535#	2541#	2542	2555#	2556	2560#	2561	2563#	2579#	2580	2583#
		2584	2586#	2618#	2619	2633#	2634	2635#	2636	2639#	2640	2643#	2644	2646#
		2662#	2663	2665#	2666	2668#	2700#	2701	2705#	2706	2713#	2714	2718#	2737#
		2738	2744#	2745	2761#	2789#	2790	2792#	2793	2796#	2797	2802#	2822#	2823
		2848#	2849	2850#	2851	2856#	2857	2861#	2883#	2884	2909#	2910	2914#	2915
		2916#	2917	2918#	2919	2924#	2925	2931#	2932	2936#	2937	2953#	2954	2955#
		2956	2982#	2983	3008#	3009	3035#	3036	3042#	3043	3045#	3052#	3053	3055#
		3062#	3063	3074#	3075	3082#	3083	3094#	3095	3101#	3102	3119#	3120	3124#

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 134  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

	3125	3127	3142	3143	3177	3178	3186	3187	3192	3193	3194	3195	3196
	3197	3199	3200	3202	3206	3207	3208	3210	3211	3213	3214	3216	3217
	3219	3220	3221	3222	3231	3232	3237	3238	3239	3240	3276	3277	3278
	3279	3284	3288	3289	3290	3346	3347	3350	3351	3353	3354	3356	3357
	3358	3359	3360	3361	3362	3363	3365	3366	3375	3376	3379	3409	3410
	3412	3413	3415	3416	3423	3440	3441	3444	3445	3447	3448	3449	3471
	3472	3473	3474	3475	3476	3478	3479	3480	3481	3483	3484	3486	3487
	3496	3497	3501	3510	3511	3513	3518	3519	3544	3545	3547	3548	3558
	3559	3564	3565	3570	3571	3573	3574	3581	3582	3584	3585	3586	3587
	3589	3617	3618	3619	3620	3623	3624	3627	3628	3630	3631	3632	3645
	3646	3655	3660	3661	3664	3665	3667	3671	3676	3686	3687	3689	3698
	3699	3706	3708	3709	3710	3719	3720	3729	3730	3746	3747	3752	3753
	3761	3765	3766	3768	3769	3771	3775	3780	3790	3791	3792	3802	3803
	3806	3808	3809	3810	3819	3820	3850	3851	3856	3880	3881	3883	3884
	3885	3886	3889	3890	3891	3892	3895	3896	3898	3899	3901	3902	3905
	3906	3911	3912	3915	3916	3918	3919	3942	3943	3944	3945	3965	3966
	3968	3970	3972	3973	3979	3980	4000	4001	4003	4005	4007	4009	4020
	4021	4022	4023	4027	4028	4037	4038	4067	4068	4073	4074	4077	4078
	4082	4083	4085	4086	4087	4175	4176	4177	4178	4179	4180	4191	4192
	4196	4197	4206	4207	4219	4224	4225	4237	4238	4241	4243	4272	4273
	4274	4278	4280	4281	4283	4287	4288	4289	4290	4291	4295	4296	4318
	4319	4320	4321	4323	4327	4328	4329	4331	4335	4361	4362	4423	4424
	4428	4432	4433	4436	4437	4438	4439	4440	4441	4442	4443	4447	4448
	4449	4450	4451	4452	4453	4454	4461	4462	4463	4464			
\$ERROR	014174	1099	4850										
\$ERRPC	001116	987	4386	4859	4860	4861	4890						
\$ERRTB	001254	1076											
\$ERTTL	001112	984	4142	4144	4858	4890							
\$ESCAP	001162	1008	1107	4881	4883	4890	4948						
\$ETABL	001214	1027											
\$ETEND	001254	969	1060										
\$FATAL	001176	1020	4379	4380	4819								
\$FFLG	014172	4782	4785	4813	4822	4830							
\$FILLC	001156	1005	4609	4640									
\$FILLS	001155	1004	4640										
\$F\$AND	000310	10	1219	1262	1274	1290	1307	1326	1350	1366	1383	1402	1432
		1465	1484	1509	1523	1539	1555	1576	1589	1605	1622	1641	1665
		1698	1717	1741	1757	1774	1793	1817	1833	1850	1869	1903	1936
		1998	2013	2052	2077	2097	2117	2143	2166	2186	2206	2232	2256
		2296	2321	2344	2364	2384	2412	2445	2463	2472	2489	2512	2546
		2593	2616	2653	2675	2699	2751	2831	2888	2908	2930	2948	2969
		2976	2996	3013	3068	3086	3107	3135	3150	3176	3185	3226	3229
		3258	3388	3392	3430	3524	3528	3562	3598	3602	3715	3744	3815
		3935	3977	4014	4018	4035	4066	4105	4190	4204	4236	4419	4422
\$F\$BAD	000401	10	1219	1262	1274	1290	1307	1326	1350	1366	1383	1402	1432
		1465	1484	1509	1523	1539	1555	1576	1589	1605	1622	1641	1665
		1698	1717	1741	1757	1774	1793	1817	1833	1850	1869	1903	1936
		1998	2013	2052	2077	2097	2117	2143	2166	2186	2206	2232	2256
		2296	2321	2344	2364	2384	2412	2445	2463	2472	2489	2512	2546
		2593	2616	2653	2675	2699	2751	2831	2888	2908	2930	2948	2969
		2976	2996	3013	3068	3086	3107	3135	3150	3176	3185	3226	3229
		3260	3388	3392	3430	3524	3528	3562	3598	3602	3715	3744	3815
		3935	3977	4014	4018	4035	4066	4105	4190	4204	4236	4419	4422
\$F\$BLA	000170	10											
\$F\$CAS	000150	10											
\$F\$DEC	000220	10											











MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 139  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

1715	1716	1717	1736	1737	1739	1740	1741	1750	1751	1752	1753	1755
1756	1757	1766	1767	1769	1770	1772	1773	1774	1783	1784	1786	1787
1791	1792	1793	1812	1813	1815	1816	1817	1826	1827	1828	1829	1831
1832	1833	1842	1843	1845	1846	1848	1849	1850	1859	1860	1862	1863
1867	1868	1869	1899	1900	1901	1902	1903	1931	1932	1934	1935	1936
1964	1965	1966	1967	1968	1996	1997	1998	2000	2001	2008	2009	2011
2012	2013	2050	2051	2052	2056	2057	2068	2069	2072	2073	2075	2076
2077	2088	2089	2092	2093	2095	2096	2097	2108	2109	2112	2113	2115
2116	2117	2141	2142	2143	2145	2146	2157	2158	2161	2162	2164	2165
2166	2177	2178	2181	2182	2184	2185	2186	2197	2198	2201	2202	2204
2205	2206	2230	2231	2232	2235	2236	2247	2248	2251	2252	2254	2255
2256	2267	2268	2271	2272	2274	2275	2276	2287	2288	2291	2292	2294
2295	2296	2319	2320	2321	2323	2324	2335	2336	2339	2340	2342	2343
2344	2355	2356	2359	2360	2362	2363	2364	2375	2376	2379	2380	2382
2383	2384	2410	2411	2412	2415	2416	2428	2429	2431	2432	2434	2435
2436	2437	2438	2440	2441	2443	2444	2445	2455	2456	2459	2460	2461
2462	2463	2470	2471	2472	2481	2482	2485	2486	2487	2488	2489	2510
2511	2512	2515	2516	2528	2529	2532	2533	2535	2536	2537	2538	2539
2541	2542	2544	2545	2546	2555	2556	2560	2561	2563	2564	2565	2566
2567	2568	2569	2570	2579	2580	2583	2584	2586	2587	2588	2589	2590
2591	2592	2593	2614	2615	2616	2618	2619	2633	2634	2635	2636	2639
2640	2643	2644	2646	2647	2648	2649	2650	2651	2652	2653	2662	2663
2665	2666	2668	2669	2670	2671	2672	2673	2674	2675	2697	2698	2699
2700	2701	2705	2706	2713	2714	2718	2719	2720	2721	2722	2723	2724
2725	2728	2729	2737	2738	2744	2745	2749	2750	2751	2761	2762	2763
2764	2765	2766	2767	2768	2769	2770	2789	2790	2792	2793	2796	2797
2802	2803	2804	2805	2806	2807	2808	2809	2812	2813	2822	2823	2829
2830	2831	2848	2849	2850	2851	2856	2857	2861	2862	2863	2864	2865
2866	2867	2868	2870	2871	2883	2884	2886	2887	2888	2906	2907	2908
2909	2910	2914	2915	2916	2917	2918	2919	2924	2925	2928	2929	2930
2931	2932	2933	2934	2936	2937	2940	2941	2942	2943	2944	2946	2947
2948	2953	2954	2955	2956	2967	2968	2969	2971	2972	2973	2974	2975
2976	2982	2983	2989	2990	2994	2995	2996	3008	3009	3011	3012	3013
3035	3036	3042	3043	3045	3046	3047	3048	3049	3052	3053	3055	3056
3057	3058	3059	3062	3063	3066	3067	3068	3074	3075	3082	3083	3084
3085	3086	3094	3095	3101	3102	3105	3106	3107	3119	3120	3124	3125
3127	3128	3129	3130	3131	3133	3134	3135	3142	3143	3148	3149	3150
3174	3175	3176	3177	3178	3183	3184	3185	3186	3187	3192	3193	3194
3195	3196	3197	3199	3200	3201	3202	3203	3204	3205	3206	3207	3208
3210	3211	3213	3214	3216	3217	3219	3220	3221	3222	3224	3225	3226
3227	3228	3229	3231	3232	3233	3234	3237	3238	3239	3240	3244	3245
3248	3249	3250	3252	3253	3256	3257	3258	3259	3260	3262	3263	3276
3277	3278	3279	3281	3282	3284	3285	3286	3287	3288	3289	3290	3346
3347	3350	3351	3353	3354	3356	3357	3358	3359	3360	3361	3362	3363
3365	3366	3375	3376	3379	3380	3381	3382	3383	3386	3387	3388	3390
3391	3392	3396	3397	3409	3410	3412	3413	3415	3416	3423	3424	3425
3426	3427	3428	3429	3430	3440	3441	3444	3445	3446	3447	3448	3449
3471	3472	3473	3474	3475	3476	3478	3479	3480	3481	3483	3484	3486
3487	3496	3497	3501	3502	3503	3504	3505	3506	3507	3508	3510	3511
3513	3514	3515	3516	3517	3518	3519	3522	3523	3524	3526	3527	3528
3533	3534	3544	3545	3547	3548	3558	3559	3560	3561	3562	3564	3565
3570	3571	3573	3574	3581	3582	3584	3585	3586	3587	3589	3590	3591
3592	3593	3596	3597	3598	3600	3601	3602	3607	3608	3617	3618	3619
3620	3623	3624	3627	3628	3629	3630	3631	3632	3645	3646	3655	3656
3657	3658	3660	3661	3664	3665	3666	3667	3668	3669	3670	3671	3676
3677	3678	3679	3680	3681	3682	3683	3686	3687	3689	3690	3691	3692

3693	3694	3695	3696	3698	3699	3706	3707	3708	3709	3710	3713	3714
3715	3719	3720	3724	3725	3729	3730	3742	3743	3744	3746	3747	3752
3753	3761	3762	3763	3764	3765	3766	3768	3769	3770	3771	3772	3773
3774	3775	3780	3781	3782	3783	3784	3785	3786	3787	3790	3791	3792
3793	3794	3795	3796	3797	3798	3799	3802	3803	3806	3807	3808	3309
3810	3813	3814	3815	3819	3820	3825	3826	3847	3848	3849	3850	3851
3856	3857	3858	3859	3880	3881	3883	3884	3885	3886	3889	3890	3891
3892	3895	3896	3898	3899	3901	3902	3905	3906	3911	3912	3915	3916
3918	3919	3925	3926	3927	3928	3929	3933	3934	3935	3942	3943	3944
3945	3965	3966	3968	3969	3970	3972	3973	3975	3976	3977	3979	3980
4000	4001	4003	4004	4005	4007	4008	4009	4012	4013	4014	4016	4017
4018	4020	4021	4022	4023	4027	4028	4033	4034	4035	4037	4038	4064
4065	4066	4067	4068	4073	4074	4077	4078	4082	4083	4085	4086	4087
4088	4089	4090	4091	4092	4093	4094	4095	4096	4103	4104	4105	4175
4176	4177	4178	4179	4180	4188	4189	4190	4191	4192	4193	4194	4196
4197	4202	4203	4204	4206	4207	4211	4212	4213	4214	4215	4219	4220
4221	4222	4223	4224	4225	4226	4227	4234	4235	4236	4237	4238	4241
4242	4243	4251	4252	4253	4254	4272	4273	4274	4275	4276	4277	4278
4280	4281	4282	4283	4284	4285	4286	4287	4288	4289	4290	4291	4292
4293	4295	4296	4297	4298	4302	4303	4318	4319	4320	4321	4322	4323
4324	4325	4326	4327	4328	4329	4330	4331	4332	4333	4334	4335	4336
4337	4339	4340	4347	4348	4361	4362	4403	4404	4417	4418	4419	4420
4421	4422	4423	4424	4425	4426	4428	4429	4432	4433	4436	4437	4438
4439	4440	4441	4442	4443	4444	4445	4447	4448	4449	4450	4451	4452
4453	4454	4457	4458	4459	4461	4462	4463	4464	4465	4466	4474	4475
10	1207	1208	1217	1219	1223	1231	1260	1262	1266	1272	1274	1278
1288	1290	1294	1305	1307	1311	1324	1326	1330	1348	1350	1354	1364
1366	1370	1381	1383	1387	1400	1402	1406	1430	1432	1436	1446	1448
1452	1463	1465	1469	1482	1484	1488	1507	1509	1513	1521	1523	1527
1537	1539	1543	1553	1555	1559	1574	1576	1580	1587	1589	1593	1603
1605	1609	1620	1622	1626	1639	1641	1645	1663	1665	1669	1679	1681
1685	1696	1698	1702	1715	1717	1721	1739	1741	1745	1755	1757	1761
1772	1774	1778	1791	1793	1797	1815	1817	1821	1831	1833	1837	1848
1850	1854	1867	1869	1873	1901	1903	1912	1934	1936	1945	1966	1968
1979	1996	1998	2003	2011	2013	2026	2050	2052	2059	2075	2077	2083
2095	2097	2103	2115	2117	2123	2141	2143	2148	2164	2166	2172	2184
2186	2192	2204	2206	2212	2230	2232	2238	2254	2256	2262	2274	2276
2282	2294	2296	2302	2319	2321	2326	2342	2344	2350	2362	2364	2370
2382	2384	2390	2410	2412	2418	2443	2445	2449	2461	2463	2467	2470
2472	2476	2487	2489	2493	2510	2512	2518	2544	2546	2550	2568	2570
2574	2591	2593	2597	2614	2616	2621	2651	2653	2657	2673	2675	2679
2697	2699	2703	2728	2729	2733	2749	2751	2755	2769	2770	2774	2812
2813	2817	2829	2831	2835	2870	2871	2875	2886	2888	2892	2906	2908
2912	2926	2927	2928	2930	2933	2934	2935	2938	2940	2946	2948	2958
2966	2967	2969	2971	2973	2974	2976	2985	2987	2989	2990	2994	2996
3001	3011	3013	3017	3066	3068	3077	3084	3086	3097	3105	3107	3115
3133	3135	3145	3148	3150	3155	3174	3176	3180	3183	3185	3189	3199
3201	3202	3203	3206	3223	3224	3226	3227	3229	3233	3234	3235	3241
3244	3245	3248	3250	3252	3253	3254	3256	3260	3262	3263	3264	3271
3273	3281	3282	3386	3388	3390	3392	3396	3397	3398	3403	3405	3428
3430	3434	3522	3524	3526	3528	3533	3534	3535	3539	3541	3560	3562
3567	3596	3598	3600	3602	3607	3608	3609	3613	3615	3664	3666	3667
3668	3671	3713	3715	3722	3724	3725	3742	3744	3749	3768	3770	3771
3772	3775	3813	3815	3822	3825	3826	3847	3849	3853	3923	3924	3925
3933	3935	3939	3975	3977	3981	4012	4014	4016	4018	4024	4029	4033
4035	4040	4064	4066	4070	4095	4096	4100	4103	4105	4109	4151	4185

#LSTST= 17777

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 141  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

		4186	4188	4190	4193	4194	4195	4198	4202	4204	4208	4211	4226	4227
		4234	4236	4239	4250	4258	4280	4282	4283	4284	4287	4292	4293	4300
		4307	4320	4322	4323	4324	4327	4328	4330	4331	4332	4335	4336	4337
		4339	4340	4345	4368	4401	4406	4415	4416	4417	4419	4420	4422	4425
		4426	4427	4434	4444	4445	4446	4455	4457	4472				
\$LSTTA=	000000	10	1207	1208	1223	1224	1266	1267	1278	1279	1294	1295	1311	1312
		1330	1331	1354	1355	1370	1371	1387	1388	1406	1407	1436	1437	1452
		1453	1469	1470	1488	1489	1513	1514	1527	1528	1543	1544	1559	1560
		1580	1581	1593	1594	1609	1610	1626	1627	1645	1646	1669	1670	1685
		1686	1702	1703	1721	1722	1745	1746	1761	1762	1778	1779	1797	1798
		1821	1822	1837	1838	1854	1855	1873	1874	1912	1913	1945	1946	1979
		1980	2003	2004	2026	2027	2059	2060	2083	2084	2103	2104	2123	2124
		2148	2149	2172	2173	2192	2193	2212	2213	2238	2239	2262	2263	2282
		2283	2302	2303	2326	2327	2350	2351	2370	2371	2390	2391	2418	2419
		2449	2450	2467	2468	2476	2477	2493	2494	2518	2519	2550	2551	2574
		2575	2597	2598	2621	2622	2657	2658	2679	2680	2703	2704	2733	2734
		2755	2756	2774	2775	2817	2818	2835	2836	2875	2876	2892	2893	2912
		2913	2926	2927	2934	2935	2938	2939	2944	2945	2958	2959	2966	2967
		2985	2986	2987	2988	2990	2991	3001	3002	3017	3018	3077	3078	3097
		3098	3115	3116	3145	3146	3155	3156	3180	3181	3189	3190	3201	3202
		3203	3204	3223	3224	3234	3235	3241	3242	3245	3246	3253	3254	3263
		3264	3271	3272	3273	3274	3282	3283	3397	3398	3403	3404	3405	3406
		3434	3435	3534	3535	3539	3540	3541	3542	3567	3568	3608	3609	3613
		3614	3615	3616	3666	3667	3668	3669	3722	3723	3725	3726	3749	3750
		3770	3771	3772	3773	3822	3823	3826	3827	3853	3854	3923	3924	3929
		3930	3939	3940	3981	3982	4024	4025	4029	4030	4040	4041	4070	4071
		4100	4101	4109	4110	4150	4151	4185	4186	4194	4195	4198	4199	4208
		4209	4227	4228	4239	4240	4250	4251	4252	4253	4257	4258	4282	4283
		4284	4285	4293	4294	4300	4301	4302	4306	4307	4322	4323	4324	4325
		4330	4331	4332	4333	4337	4338	4340	4341	4345	4346	4347	4367	4368
		4401	4402	4403	4405	4406	4415	4416	4426	4427	4434	4435	4445	4446
		4455	4456	4472	4473	4474								
\$MADR1	001226	10450												
\$MADR2	001232	10490												
\$MADR3	001236	10520												
\$MADR4	001242	10550												
\$MAIL	001174	965	969	10180	1127	1142	1194	1257	1341	1418	1505	1572	1657	1733
		1809	1892	1925	1958	1992	2047	2138	2227	2316	2407	2507	2611	2693
		2785	2844	2902	3032	3172	3343	3467	3642	3740	3842	4062	4584	4867
		4945												
\$MAMS1	001224	10390												
\$MAMS2	001230	10470												
\$MAMS3	001234	10500												
\$MAMS4	001240	10530												
\$MBADR	001002	9650												
\$MCALL =	000000	10												
\$MFLG	014170	4783*	4789	4824*	48280									
\$MNEW	013715	4665	47770											
\$MSGAD	001210	10250	4799*	4802										
\$MSGLG	001212	10260	4804*											
\$MSGTY	001174	10190	4797	4805*	4817	4821*								
\$MSMR	013704	4662	47750											
\$MTYP1	001225	10400												
\$MTYP2	001231	10480												
\$MTYP3	001235	10510												
\$MTYP4	001241	10540												



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 143  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

\$PASTM	001006	9670												
\$POWER	012660	4551	45580											
\$PWRAD	012646	45530												
\$PWRDN	012506	1103	45200	4548										
\$PWRMG	012642	45510												
\$PWRUP	012560	4530	45360											
\$QUES	001170	10100	4640	4708	4757	4773	4890							
\$RDCHR	013434	47210	5138											
\$RDDEC=	***** U	5140												
\$RDLIN	013554	47490	5139											
\$RDOCT=	***** U	5140												
\$RDSZ =	000010	47420												
\$RTNAD	012464	45110												
\$R2A =	***** U	5140												
\$SAVLE=	177777	10	29890	29900	32020	32060	32440	32450	36670	36710	37710	37750	42260	42270
		42830	42870	43230	43270	43310	43350							
\$SAVRE=	***** U	5140												
\$SAVR6	012656	45290	4537	45380	45390	45570								
\$SCOPE	014374	1097	49040											
\$SETUP=	000137	10880	1096	1097	1099	1101	1103	1105	1106	1107	1109	1136	1139	4488
		4645	4779	4851	4877	4885	4905							
\$SSKO =	000236	29890	2990	32020	3206	32440	3245	36670	3671	37710	3775	42260	4227	42830
		4287	43230	4327	43310	4335								
\$STUP =	177777	10880												
\$SVLAD	014602	4915	49440											
\$SVPC =	000204	9410	946											
\$SMR =	167400	10	689	693	694	695	696	697	698	699	1007	1008	1009	1106
		1107	1109	1110	1193	1256	1340	1417	1504	1571	1656	1732	1808	1891
		1924	1957	1991	2046	2137	2226	2315	2406	2506	2610	2692	2784	2843
		2901	3031	3171	3342	3466	3641	3739	3841	4061	4125	4483	4489	4504
		4510	4512	4554	4842	4843	4844	4845	4846	4855	4862	4874	4878	4890
		4896	4897	4898	4899	4900	4906	4918	4920	4921	4924	4925	4926	4933
		4934	4935	4947	4950	4953								
\$SMREG	001216	10300	1130											
\$SMRPK=	000000	699	700	4900	4901	4922								
\$TAGLE=	177777	10	12080	12190	12230	12310	12620	12660	12740	12780	12900	12940	13070	13110
		13260	13300	13500	13540	13660	13700	13830	13870	14020	14060	14320	14360	14480
		14520	14650	14690	14840	14880	15090	15130	15230	15270	15390	15430	15550	15590
		15760	15800	15890	15930	16050	16090	16220	16260	16410	16450	16650	16690	16810
		16850	16980	17020	17170	17210	17410	17450	17570	17610	17740	17780	17930	17970
		18170	18210	18330	18370	18500	18540	18690	18730	19030	19120	19360	19450	19680
		19790	19980	20030	20130	20260	20520	20590	20770	20830	20970	21030	21170	21230
		21430	21480	21660	21720	21860	21920	22060	22120	22320	22380	22560	22620	22760
		22820	22960	23020	23210	23260	23440	23500	23640	23700	23840	23900	24120	24180
		24450	24490	24630	24670	24720	24760	24890	24930	25120	25180	25460	25500	25700
		25740	25930	25970	26160	26210	26530	26570	26750	26790	26990	27030	27290	27330
		27510	27550	27700	27740	28130	28170	28310	28350	28710	28750	28880	28920	29080
		29120	29270	29300	29340	29350	29380	29400	29480	29580	29670	29690	29730	29760
		29850	29870	29890	29960	30010	30130	30170	30680	30770	30860	30970	31070	31150
		31350	31450	31500	31550	31760	31800	31850	31890	32010	32030	32060	32240	32260
		32290	32340	32350	32410	32440	32500	32530	32540	32600	32630	32640	32710	32730
		32810	32820	33880	33920	33970	33980	34030	34050	34300	34340	35240	35280	35340
		35350	35390	35410	35620	35670	35980	36020	36080	36090	36130	36150	36660	36680
		36710	37150	37220	37240	37250	37440	37490	37700	37720	37750	38150	38220	38250
		38260	38490	38530	39240	39250	39350	39390	39770	39810	40140	40180	40240	40290
		40350	40400	40660	40700	40960	41000	41050	41090	41860	41900	41940	41950	41980

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 144  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

\$TAGNU= 000251

42040	42080	4211	42260	42360	42390	42820	42840	42870	42920	42930	43220	43240
43270	43300	43320	43350	43360	43370	43390	43400	44160	44190	44220	44260	44270
44340	44450	44460	44550	44570								
10	1207	12080	1218	12190	1261	12620	1273	12740	1289	12900	1306	13070
1325	13260	1349	13500	1365	13660	1382	13830	1401	14020	1431	14320	1447
14480	1464	14650	1483	14840	1508	15090	1522	15230	1538	15390	1554	15550
1575	15760	1588	15890	1604	16050	1621	16220	1640	16410	1664	16650	1680
16810	1697	16980	1716	17170	1740	17410	1756	17570	1773	17740	1792	17930
1816	18170	1832	18330	1849	18500	1868	13690	1902	19030	1935	19360	1967
19680	1997	19980	2012	20130	2051	20520	2076	20770	2096	20970	2116	21170
2142	21430	2165	21660	2185	21860	2205	22060	2231	22320	2255	22560	2275
22760	2295	22960	2320	23210	2343	23440	2363	23640	2383	23840	2411	24120
2444	24450	2462	24630	2471	24720	2488	24890	2511	25120	2545	25460	2569
25700	2592	25930	2615	26160	2652	26530	2674	26750	2698	26990	2728	27290
2750	27510	2769	27700	2812	28130	2830	28310	2870	28710	2887	28880	2907
29080	2926	29270	2929	29300	2933	29350	2941	2944	29450	2947	29480	2966
29670	2968	29690	2972	29730	2975	29760	2995	29960	3012	30130	3067	30680
3085	30860	3106	31070	3134	31350	3149	31500	3175	31760	3184	31850	3200
32010	32020	3205	32060	3223	32240	3225	32260	3228	32290	3233	32350	3249
32500	3252	32540	3257	3259	32600	3262	32640	3387	33880	3391	33920	3396
33980	3429	34300	3523	35240	3527	35280	3533	35350	3561	35620	3597	35980
3601	36020	3607	36090	3665	36660	36670	3670	36710	3714	37150	3743	37440
3769	37700	37710	3774	37750	3814	38150	3848	38490	3923	39240	3926	3929
39300	3934	39350	3976	39770	4013	40140	4017	40180	4034	40350	4065	40660
4095	40960	4104	41050	41510	4185	41860	4189	41900	4193	41950	4203	42040
4235	42360	42580	4281	42820	42830	4286	42870	43070	4321	43220	43230	4326
43270	4329	43300	43310	4334	43350	43680	44060	4415	44160	4418	44190	4421
44220	4425	44270	4444	44460								
11540	11550	11610	11620	11640	11650	11660	11670	11690	11700	11710	11730	11740
11760	11770	11790	11800	11820	11830	11840	11960	11970	11990	12000	12010	12020
12030	12040	12050	12060	12090	12100	12120	12130	12230	12260	12270	12280	12300
12310	1232	12360	12370	12390	12400	12410	12630	12640	12660	12690	12700	12780
12830	12840	12850	12860	12940	12990	13000	13020	13030	13110	13160	13170	13190
13200	13300	13450	13460	13540	13590	13600	13610	13620	13700	13750	13760	13780
13790	13870	13920	13930	13950	13960	14060	14270	14280	14360	14410	14420	14430
14440	14520	14570	14580	14600	14610	14690	14740	14750	14770	14780	14880	15100
15110	15130	15160	15170	15180	15190	15270	15320	15330	15350	15360	15430	15480
15490	15500	15510	15590	15770	15780	15800	15840	15850	15930	15980	15990	16000
16010	16090	16140	16150	16170	16180	16260	16310	16320	16340	16350	16450	16600
16610	16690	16740	16750	16760	16770	16850	16900	16910	16930	16940	17020	17060
17070	17100	17110	17210	17360	17370	17450	17500	17510	17520	17530	17610	17660
17670	17690	17700	17780	17830	17840	17860	17870	17970	18120	18130	18210	18260
18270	18280	18290	18370	18420	18430	18450	18460	18540	18590	18600	18620	18630
18730	18990	19000	19120	19310	19320	19450	19640	19650	19790	20000	20010	20030
20080	20090	20260	20560	20570	20590	20680	20690	20720	20730	20830	20880	20890
20920	20930	21030	21080	21090	21120	21130	21230	21450	21460	21480	21570	21580
21610	21620	21720	21770	21780	21810	21820	21920	21970	21980	22010	22020	22120
22350	22360	22380	22470	22480	22510	22520	22620	22670	22680	22710	22720	22820
22870	22880	22910	22920	23020	23230	23240	23260	23350	23360	23390	23400	23500
23550	23560	23590	23600	23700	23750	23760	23790	23800	23900	24150	24160	24180
24280	24290	24310	24320	24400	24410	24490	24550	24560	24590	24600	24670	24760
24810	24820	24850	24860	24930	25150	25160	25180	25280	25290	25320	25330	25410
25420	25500	25550	25560	25600	25610	25740	25790	25800	25830	25840	25970	26180
26190	26210	26330	26340	26350	26360	26390	26400	26430	26440	26570	26620	26630
26650	26660	26790	27000	27010	27030	27050	27060	27130	27140	27330	27370	27380
27440	27450	27550	27740	27890	27900	27920	27930	27960	27970	28170	28220	28230

\$TEMP = 000300



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 146  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

\$TPFLG	001157	1006#	4578	4640										
\$TPS	001150	1001#	4627	4640										
\$TRAP	015326	1101	5106#											
\$TRAP2	015350	5117#	5128											
\$TRP =	000012	5121#	5130#	5131#	5132#	5133#	5134#	5135	5136#	5137	5138#	5139#	5140#	
\$TRPAD	015362	5111	5128#											
\$TSKO =	000244	1208#	1231	1262#	1266	1274#	1278	1290#	1294	1307#	1311	1326#	1330	1350#
		1354	1366#	1370	1383#	1387	1402#	1406	1432#	1436	1448#	1452	1465#	1469
		1484#	1488	1509#	1513	1523#	1527	1539#	1543	1555#	1559	1576#	1580	1589#
		1593	1605#	1609	1622#	1626	1641#	1645	1665#	1669	1681#	1685	1698#	1702
		1717#	1721	1741#	1745	1757#	1761	1774#	1778	1793#	1797	1817#	1821	1833#
		1837	1850#	1854	1869#	1873	1903#	1912	1936#	1945	1968#	1979	1998#	2003
		2013#	2026	2052#	2059	2077#	2083	2097#	2103	2117#	2123	2143#	2148	2166#
		2172	2186#	2192	2206#	2212	2232#	2238	2256#	2262	2276#	2282	2296#	2302
		2321#	2326	2344#	2350	2364#	2370	2384#	2390	2412#	2418	2445#	2449	2463#
		2467	2472#	2476	2489#	2493	2512#	2518	2546#	2550	2570#	2574	2593#	2597
		2616#	2621	2653#	2657	2675#	2679	2699#	2703	2729#	2733	2751#	2755	2770#
		2774	2813#	2817	2831#	2835	2871#	2875	2888#	2892	2908#	2912	2927#	2940
		2948#	2958	2967#	2989	2996#	3001	3013#	3017	3068#	3077	3086#	3097	3107#
		3115	3135#	3145	3150#	3155	3176#	3180	3185#	3189	3201#	3203	3206#	3282
		3388#	3405	3430#	3434	3524#	3541	3562#	3567	3598#	3615	3666#	3668	3671#
		3725	3744#	3749	3770#	3772	3775#	3826	3849#	3853	3924#	3925	3935#	3939
		3977#	3981	4014#	4029	4035#	4040	4066#	4070	4096#	4100	4105#	4109	4186#
		4226	4236#	4239	4282#	4284	4287#	4293	4322#	4324	4327#	4340	4416#	4457
\$TSK1 =	000250	1219#	1223	2930#	2934	2935#	2938	2969#	2989	3206#	3281	3392#	3397	3398#
		3403	3528#	3534	3535#	3539	3602#	3608	3609#	3613	3671#	3724	3775#	3825
		4018#	4024	4186#	4211	4226	4287#	4292	4327#	4339	4419#	4445	4446#	4455
\$TSK2 =	000247	2973#	2987	3224#	3244	3250#	3253	3254#	3273	3715#	3722	3815#	3822	4190#
		4194	4195#	4198	4204#	4208	4330#	4332	4335#	4337	4422#	4426	4427#	4434
\$TSK3 =	000236	2976#	2985	3226#	3244	3260#	3263	3264#	3271	4335#	4336			
\$TSK4 =	000147	3229#	3234	3235#	3241									
\$TSTM	001004	966#												
\$TSTNM	001102	979#	4488#	4854	4890	4895	4922	4944#	4945	4950	4954			
\$TTYIN	013662	4750	4751	4768	4772#									
\$TYPBN=	***** U	5134												
\$TYPDS	014654	4966#	5133											
\$TYPE	012670	4578#	4810	5121	5129									
\$TYPEC	013102	4608	4615	4622	4627#	4628	4694							
\$TYPEX	013150	4633	4635	4638#										
\$TYPOC	015124	5051#	5130											
\$TYPON	015140	5050	5053#	5132										
\$TYPOS	015100	5046#	5131											
\$UNIT	001206	1024#												
\$UNITM	001010	968#												
\$USWR	001220	1031#	1260	1507	1574	1996	2050	2141	2230	2319	2410	2510	2614	3174
		3284	3560	3742	4064	4274								
		1056#	4442											
\$VECT1	001244	1057#												
\$VECT2	001246	4909#												
\$XTSTR	014406	1172#	1173#	1175#	1176#	1178#	1179#	1181#	1182#	1226#	1227#	1229#	1230#	1238#
\$YESNO=	000001	1239#	1285#	1286#	1302#	1303#	1319#	1320#	1361#	1362#	1378#	1379#	1395#	1396#
		1443#	1444#	1460#	1461#	1477#	1478#	1518#	1519#	1535#	1536#	1550#	1551#	1600#
		1601#	1617#	1618#	1634#	1635#	1676#	1677#	1693#	1694#	1710#	1711#	1752#	1753#
		1769#	1770#	1786#	1787#	1828#	1829#	1845#	1846#	1862#	1863#	2072#	2073#	2092#
		2093#	2112#	2113#	2161#	2162#	2181#	2182#	2201#	2202#	2251#	2252#	2271#	2272#
		2291#	2292#	2339#	2340#	2359#	2360#	2379#	2380#	2431#	2432#	2459#	2460#	2485#



	24860	25320	25330	25600	25610	25830	25840	26330	26340	26430	26440	26650	26660
	27890	27900	28480	28490	29140	29150	29360	29370	31960	31970	32020	32030	32370
	32380	32390	32400	32850	32880	33530	33540	33650	33660	33750	33760	34150	34160
	34460	34470	34830	34840	34860	34870	35100	35110	35440	35450	35470	35480	35730
	35740	35810	35820	35840	35850	35860	35870	36170	36180	36190	36200	36290	36300
	36450	36460	36670	36680	37070	37080	37090	37100	37290	37300	37520	37530	37710
	37720	38070	38080	38090	38100	39110	39120	39150	39160	39180	39190	39420	39430
	39440	39450	39650	39660	39690	39700	39790	39800	40000	40010	40040	40050	40080
	40090	40270	40280	40370	40380	40730	40740	40820	40830	42240	42250	42750	42780
	42830	42840	42880	42890	42900	42910	42950	42960	43230	43240	43310	43320	43610
	43620	44490	44500	44510	44520	44530	44540	44630	44640				
##ARGC = 000000	41510	42580	43070	43680	44060								
##BYTE = 000403	12170	12600	12720	12880	13050	13240	13480	13640	13810	14000	14300	14460	14630
	14820	15070	15210	15370	15530	15740	15870	16030	16200	16390	16630	16790	16960
	17150	17390	17550	17720	17910	18150	18310	18480	18670	19010	19340	19660	19960
	20110	20500	20750	20950	21150	21410	21640	21840	22040	22300	22540	22740	22940
	23190	23420	23620	23820	24100	24430	24610	24700	24870	25100	25440	25680	25910
	26140	26510	26730	26970	27490	28290	28860	29060	29280	29460	29670	29710	29740
	29940	30110	30660	30840	31050	31330	31480	31740	31830	32240	32270	32480	32560
	32580	33860	33900	34280	35220	35260	35600	35960	36000	37130	37420	38130	38470
##DST = 000067	39330	39750	40120	40160	40330	40640	41030	41880	42020	42340	44170	44200	
##FLAG = 000001	32850	42750											
	12170	1219	12230	12600	1262	12660	12720	1274	12780	12880	1290	12940	13050
	1307	13110	13240	1326	13300	13480	1350	13540	13640	1366	13700	13810	1383
	13870	14000	1402	14060	14300	1432	14360	14460	1448	14520	14630	1465	14690
	14820	1484	14880	15070	1509	15130	15210	1523	15270	15370	1539	15430	15530
	1555	15590	15740	1576	15800	15870	1589	15930	16030	1605	16090	16200	1622
	16260	16390	1641	16450	16630	1665	16690	16790	1681	16850	16960	1698	17020
	17150	1717	17210	17390	1741	17450	17550	1757	17610	17720	1774	17780	17910
	1793	17970	18150	1817	18210	18310	1833	18370	18480	1850	18540	18670	1869
	18730	19010	1903	19120	19340	1936	19450	19660	1968	19790	19960	1998	20030
	20110	2013	20260	20500	2052	20590	20750	2077	20830	20950	2097	21030	21150
	2117	21230	21410	2143	21480	21640	2166	21720	21840	2186	21920	22040	2206
	22120	22300	2232	22380	22540	2256	22620	22740	2276	22820	22940	2296	23020
	23190	2321	23260	23420	2344	23500	23620	2364	23700	23820	2384	23900	24100
	2412	24180	24430	2445	24490	24610	2463	24670	24700	2472	24760	24870	2489
	24930	25100	2512	25180	25440	2546	25500	25680	2570	25740	25910	2593	25970
	26140	2616	26210	26510	2653	26570	26730	2675	26790	26970	2699	27030	27280
	27330	27490	2751	27550	27690	27740	28120	28170	28290	2831	28350	28700	28750
	28860	2888	28920	29060	2908	29120	29280	2930	29380	29460	2948	29580	29660
	29670	2969	29710	2973	29740	2976	29850	29870	29940	2996	30010	30110	3013
	30170	30660	3068	30770	30840	3086	30970	31050	3107	31150	31330	3135	31450
	31480	3150	31550	31740	3176	31800	31830	3185	31890	32230	32240	3226	32270
	3229	32410	32480	3250	32560	32580	3260	32710	32730	33860	3388	33900	3392
	34030	34050	34280	3430	34340	35220	3524	35260	3528	35390	35410	35600	3562
	35670	35960	3598	36000	3602	36130	36150	37130	3715	37220	37420	3744	37490
	38130	3815	38220	38470	3849	38530	39330	3935	39390	39750	3977	39810	40120
	4014	40160	4018	40240	40290	40330	4035	40400	40640	4066	40700	40950	41000
	41030	4105	41090	41880	4190	41980	42020	4204	42080	42340	4236	42390	44170
	4419	44200	4422	44340	44550								
##FROM = 000000	11570	24340	25350	25630	25860	26460	26680	27180	27610	28020	28610	30450	30550
	31270	33790	34230	35010	35130	35890	36550	36760	36890	37610	37800	37920	38560
	40870	42190	44280										
##GET4 = 000000	45040												
##LOC = 012336	12180	1219	12320	1233	12610	1262	12730	1274	12890	1290	13060	1307	13250
	1326	13490	1350	13650	1366	13820	1383	14010	1402	14310	1432	14470	1448

1464	1465	1483	1484	1508	1509	1522	1523	1538	1539	1554	1555	1575
1576	1588	1589	1604	1605	1621	1622	1640	1641	1664	1665	1680	1681
1697	1698	1716	1717	1740	1741	1756	1757	1773	1774	1792	1793	1816
1817	1832	1833	1849	1850	1868	1869	1902	1903	1935	1936	1967	1968
1997	1998	2012	2013	2051	2052	2076	2077	2096	2097	2116	2117	2142
2143	2165	2166	2185	2186	2205	2206	2231	2232	2255	2256	2275	2276
2295	2296	2320	2321	2343	2344	2363	2364	2383	2384	2411	2412	2444
2445	2462	2463	2471	2472	2488	2489	2511	2512	2545	2546	2569	2570
2592	2593	2615	2616	2652	2653	2674	2675	2698	2699	2728	2729	2750
2751	2769	2770	2812	2813	2830	2831	2870	2871	2887	2888	2907	2908
2929	2930	2943	2944	2947	2948	2968	2969	2972	2973	2975	2976	2995
2996	3012	3013	3067	3068	3085	3086	3106	3107	3134	3135	3149	3150
3175	3176	3184	3185	3225	3226	3228	3229	3249	3250	3257	3258	3259
3260	3387	3388	3391	3392	3429	3430	3523	3524	3527	3528	3561	3562
3597	3598	3601	3602	3714	3715	3743	3744	3814	3815	3848	3849	3928
3929	3934	3935	3976	3977	4013	4014	4017	4018	4034	4035	4065	4066
4095	4096	4104	4105	4189	4190	4203	4204	4235	4236	4418	4419	4421
4422	4458	4459										
4151	4253	4258	4302	4307	4347	4368	4403	4406	4474			
4151	4237	4241	4251	4258	4301	4307	4346	4368	4402	4406	4473	
4151	4237	4250	4258	4297	4300	4307	4345	4368	4401	4406	4465	4472
4151	4242	4252	4258	4301	4307	4346	4368	4402	4406	4473		
3285	4275											
1157	1158	2434	2437	2535	2538	2563	2566	2586	2589	2646	2649	2668
2671	2718	2724	2761	2767	2802	2808	2861	2867	3045	3048	3055	3058
3127	3130	3379	3382	3423	3426	3501	3507	3513	3516	3589	3592	3655
3656	3657	3676	3682	3689	3695	3761	3762	3763	3780	3786	3792	3798
3856	3857	3858	4087	4093	4219	4222	4428	4429				
5047	5051	5061	5096									

\$LOCN= 000000  
 \$NETU= 000000  
 \$RTN1= 000242  
 \$RTN2= 000243  
 \$SRC = 000027  
 \$TO = 000000

\$OFILL 015323  
 \$1 002126  
 \$10 002446  
 \$100 005260  
 \$101 005322  
 \$102 005364  
 \$103 005424  
 \$104 005512  
 \$105 005554  
 \$106 005612  
 \$107 005660  
 \$11 002474  
 \$110 005706  
 \$111 005742  
 \$112 006034  
 \$113 006056  
 \$114 006150  
 \$115 006166  
 \$116 006224  
 \$117 006244  
 \$12 002522  
 \$120 006262  
 \$121 006264  
 \$122 006300  
 \$123 006324  
 \$124 006324  
 \$125 006370  
 \$126 006366

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 149  
 CVDVAD.F11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

\$127	006366	2975	29850	
\$13	002552	1401	14060	
\$130	006402	2995	30010	
\$131	006420	3012	30170	
\$132	006534	3067	30770	
\$133	006562	3085	30970	
\$134	006604	3106	31150	
\$135	006654	3134	31450	
\$136	006666	3149	31550	
\$137	006726	3175	31800	
\$14	002622	1431	14360	
\$140	006746	3184	31890	
\$141	006776	3200	32030	
\$142	006774	32010	3281	
\$143	007144	3205	32820	
\$144	007034	32230	3244	
\$145	007070	3225	32450	
\$146	007056	3228	32340	
\$147	007066	3233	32410	
\$15	002650	1447	14520	
\$150	007102	3249	32530	
\$151	007126	3252	32730	
\$152	007124	3257	3259	32630
\$153	007126	3262	32710	
\$154	007416	3387	34050	
\$155	007414	3391	33970	
\$156	007416	3396	34030	
\$157	007474	3429	34340	
\$16	002676	1464	14690	
\$160	007734	3523	35410	
\$161	007732	3527	35340	
\$162	007734	3533	35390	
\$163	007776	3561	35670	
\$164	010104	3597	36150	
\$165	010102	3601	36080	
\$166	010104	3607	36130	
\$167	010234	3665	36680	
\$17	002726	1483	14880	
\$170	010232	36660	3724	
\$171	010360	3670	37250	
\$172	010356	3714	37220	
\$173	010424	3743	37490	
\$174	010470	3769	37720	
\$175	010466	37700	3825	
\$176	010614	3774	38260	
\$177	010612	3814	38220	
\$2	002152	1218	12230	
\$20	002764	1508	15130	
\$200	010652	3848	38530	
\$201	010764	39230	3928	
\$202	011000	3926	39290	
\$203	011010	3934	39390	
\$204	011070	3976	39810	
\$205	011150	4013	40290	
\$206	011144	4017	40240	
\$207	011164	4034	40400	

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 150  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

\$21	003012	1522	1527#	
\$210	011230	4065	4070#	
\$211	011312	4095	4100#	
\$212	011322	4104	4109#	
\$213	011646	4237	4250#	
\$214	011650	4242	4252#	
\$215	011516	4185#	4226	
\$216	011622	4212	4214	4227#
\$217	011536	4189	4194#	
\$22	003040	1538	1543#	
\$220	011544	4193	4198#	
\$221	011562	4203	4208#	
\$222	011634	4235	4239#	
\$223	011750	4297	4300#	
\$224	011750	4301#		
\$225	011714	4281	4284#	
\$226	011710	4282#	4292	
\$227	011740	4286	4293#	
\$23	003066	1554	1559#	
\$230	012030	4345#		
\$231	012030	4346#		
\$232	011776	4321	4324#	
\$233	011772	4322#	4339	
\$234	012022	4326	4340#	
\$235	012010	4329	4332#	
\$236	012006	4330#	4336	
\$237	012020	4334	4337#	
\$24	003124	1575	1580#	
\$240	012222	4401#		
\$241	012222	4402#		
\$242	012362	4465	4472#	
\$243	012362	4473#		
\$244	012224	4415#	4458	
\$245	012310	4418	4445#	
\$246	012250	4421	4426#	
\$247	012256	4425	4434#	
\$25	003144	1588	1593#	
\$250	012330	4444	4455#	
\$26	003172	1604	1609#	
\$27	003220	1621	1626#	
\$3	002264	1261	1266#	
\$30	003250	1640	1645#	
\$31	003306	1664	1669#	
\$32	003334	1680	1685#	
\$33	003362	1697	1702#	
\$34	003412	1716	1721#	
\$35	003450	1740	1745#	
\$36	003476	1756	1761#	
\$37	003524	1773	1778#	
\$4	002304	1273	1278#	
\$40	003554	1792	1797#	
\$40CAT=	***** U	4864	4906	
\$41	003612	1816	1821#	
\$42	003640	1832	1837#	
\$43	003666	1849	1854#	
\$44	003716	1868	1873#	



BEGIN	10														
BGNHRD	10														
BGNHW	10														
BGNINI	10														
BGNMOD	10	4147													
BGNMSG	10														
BGNSFT	10														
BGNSRV	10	3959	3994	4351											
BGNSUB	10	1208	1268	1282	1298	1315	1344	1358	1374	1391	1426	1440	1456	1473	1515
	1531	1547	1583	1597	1613	1630	1659	1673	1689	1705	1735	1749	1765	1782	1811
	1825	1841	1858	1898	1930	1963	2007	2067	2087	2107	2156	2176	2196	2246	2266
	2286	2334	2354	2374	2427	2454	2480	2527	2554	2578	2638	2661	2704	2736	2791
	2821	2849	3034	3081	3100	3118	3361	3408	3477	3557					
BGNSW	10														
BRESET	10	1184	1321	1397	1479	1636	1712	1788	1864	1909	1942	1976	2023	2825	3435
	3907	4110													
CALL	10	1156	2434	2535	2563	2586	2646	2668	2717	2760	2801	2860	3045	3055	3127
	3379	3423	3500	3513	3589	3654	3675	3688	3760	3779	3791	3855	4086	4219	4427
CASE	10														
CKLOOP	10	1874	1914	1947	1980	2028									
CLRVEC	10	1233	3441	3624											
COMMEN	10	8110													
DECR	10														
DECRU	10														
DEFAULT	10														
DEVREG	10														
DEVTYP	10														
DISPAT	10														
ELSE	10	2932	3232	3251	3261	3395	3532	3606	4192	4424	4443				
END	10														
ENDCLN	10														
ENDCOM	10	8110													
ENDDC	10														
ENDDO	10	2988	3243												
ENDHRD	10														
ENDHW	10														
ENDIF	10	1222	1265	1277	1293	1310	1329	1353	1369	1386	1405	1435	1451	1468	1487
	1512	1526	1542	1558	1579	1592	1608	1625	1644	1668	1684	1701	1720	1744	1760
	1777	1796	1820	1836	1853	1872	1911	1944	1978	2002	2025	2058	2082	2102	2122
	2147	2171	2191	2211	2237	2261	2281	2301	2325	2349	2369	2389	2417	2448	2466
	2475	2492	2517	2549	2573	2596	2620	2656	2678	2702	2732	2754	2773	2816	2834
	2874	2891	2911	2937	2957	2984	2986	3000	3016	3076	3096	3114	3144	3154	3179
	3188	3240	3270	3272	3402	3404	3433	3538	3540	3566	3612	3614	3721	3748	3821
	3852	3938	3980	4023	4028	4039	4069	4099	4108	4197	4207	4238	4433	4454	
ENDINC	10	3280	3723	3824	4291	4335	4338								
ENDINI	10														
ENDLOO	10	4225													
ENDMOD	10														
ENDMSG	10														
ENDRTN	10	4249	4299	4344	4400	4471									
ENDSEL	10														
ENDSFT	10														
ENDSRV	10	3987	4047	4362											
ENDSUB	10	1224	1279	1295	1312	1331	1355	1371	1388	1407	1437	1453	1470	1489	1528
	1544	1560	1594	1610	1627	1646	1670	1686	1703	1722	1746	1762	1779	1798	1822
	1838	1855	1875	1915	1948	1981	2029	2084	2104	2124	2173	2193	2213	2263	2283

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 154  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

	2303	2351	2371	2391	2451	2477	2494	2551	2575	2598	2658	2680	2734	2775	2819
	2836	2876	3078	3098	3116	3156	3406	3437	3548	3620					
ENDSW	10														
ENDTST	10	1244	1332	1408	1490	1561	1647	1723	1799	1876	1916	1949	1982	2030	2125
	2214	2304	2392	2495	2601	2681	2776	2837	2893	3023	3160	3328	3451	3634	3730
	3830	4050	4119												
EQUALS	10														
ERRDF	10	1220													
ERRHRD	10	1275	1291	1308	1327	1351	1367	1384	1403	1433	1449	1466	1485	1524	1540
	1556	1590	1606	1623	1642	1666	1682	1699	1718	1742	1758	1775	1794	1818	1834
	1851	1870	1906	1939	1969	2015	2078	2098	2118	2167	2187	2207	2257	2277	2297
	2345	2365	2385	2446	2464	2473	2490	2547	2571	2594	2654	2676	2730	2752	2771
	2814	2832	2872	2889	2950	2978	2997	3014	3069	3088	3109	3136	3151	3268	3393
	3400	3431	3529	3536	3603	3610	3716	3816	3936	4097	4106				
ERROR	7050	1221	1276	1292	1309	1328	1352	1368	1385	1404	1434	1450	1467	1486	1525
	1541	1557	1591	1607	1624	1643	1667	1683	1700	1719	1743	1759	1776	1795	1819
	1835	1852	1871	1907	1940	1970	2016	2079	2099	2119	2168	2188	2208	2258	2278
	2298	2346	2366	2386	2447	2465	2474	2491	2548	2572	2595	2655	2677	2731	2753
	2772	2815	2833	2873	2890	2951	2979	2998	3015	3070	3089	3110	3137	3152	3269
	3394	3401	3432	3530	3537	3604	3611	3717	3817	3937	4098	4107			
ESCAPE	10	8110													
EXIF	10														
EXIFB	10	4210													
EXIT	10	1262	1509	1576	1999	2055	2144	2234	2322	2414	2514	2599	2617	2699	2908
	2954	2981	3019	3073	3093	3112	3141	3157	3176	3185	3291	3563	3718	3745	3818
	3849	3946	4066	4112											
GETPRI	10	8110													
GETSMR	10	8110	11390												
GPHARD	10														
GPRMA	10														
GPRMD	10														
GPRML	10														
HEADER	10														
IF	10	1216	1259	1271	1287	1304	1323	1347	1363	1380	1399	1429	1445	1462	1481
	1506	1520	1536	1552	1573	1586	1602	1619	1638	1662	1678	1695	1714	1738	1754
	1771	1790	1814	1830	1847	1866	1900	1933	1965	1995	2010	2049	2074	2094	2114
	2140	2163	2183	2203	2229	2253	2273	2293	2318	2341	2361	2381	2409	2442	2460
	2469	2486	2509	2543	2567	2590	2613	2650	2672	2696	2748	2828	2885	2905	2927
	2945	2970	2973	2993	3010	3065	3083	3104	3132	3147	3173	3182	3226	3247	3255
	3385	3389	3427	3521	3525	3559	3595	3599	3712	3741	3812	3846	3932	3974	4011
	4015	4032	4063	4187	4416	4419									
IFB	10	4102	4201	4233											
IFCOND	10														
IF.ERR	10	2727	2768	2811	2869	4094									
IF.NO.	10														
INCR	10	3198	3663	3767	4279	4327									
INCRU	10	4319													
INLINE	10														
LASTAD	10														
LEAVE	10														
LET	10	1153	1160	1163	1165	1168	1170	1173	1176	1179	1182	1195	1199	1200	1201
	1202	1203	1204	1209	1211	1225	1227	1236	1237	1239	1240	1263	1269	1283	1284
	1299	1301	1316	1318	1345	1359	1360	1375	1377	1392	1394	1427	1441	1442	1457
	1459	1474	1476	1510	1516	1517	1532	1534	1548	1549	1577	1584	1598	1599	1614
	1616	1631	1633	1660	1674	1675	1690	1692	1706	1709	1736	1750	1751	1766	1768
	1783	1785	1812	1826	1827	1842	1844	1859	1861	1899	1931	1964	2000	2008	2056

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 155  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

	2068	2071	2088	2091	2108	2111	2145	2157	2160	2177	2180	2197	2200	2235	2247
	2250	2267	2270	2287	2290	2323	2335	2338	2355	2358	2375	2378	2415	2428	2430
	2439	2455	2458	2481	2484	2515	2528	2531	2540	2555	2559	2579	2582	2618	2632
	2634	2639	2642	2662	2664	2700	2705	2712	2737	2743	2788	2792	2795	2822	2847
	2850	2855	2882	2909	2913	2915	2917	2923	2930	2935	2952	2955	2982	3007	3035
	3041	3051	3061	3074	3082	3094	3101	3119	3123	3142	3177	3186	3191	3193	3195
	3206	3209	3212	3215	3218	3220	3230	3236	3238	3275	3277	3283	3288	3345	3349
	3352	3356	3357	3358	3359	3360	3362	3364	3374	3409	3411	3414	3439	3444	3445
	3447	3448	3471	3472	3473	3474	3475	3478	3479	3482	3485	3495	3509	3517	3543
	3546	3558	3564	3569	3572	3580	3583	3585	3616	3618	3622	3627	3628	3630	3631
	3644	3659	3685	3697	3705	3708	3719	3728	3746	3751	3764	3789	3801	3805	3808
	3819	3850	3879	3882	3884	3888	3890	3894	3897	3900	3904	3910	3914	3917	3941
	3943	3964	3967	3971	3978	3999	4002	4006	4019	4021	4026	4036	4067	4072	4076
	4081	4084	4174	4176	4178	4190	4195	4205	4223	4271	4273	4287	4289	4294	4317
	4360	4422	4431	4435	4437	4439	4441	4446	4448	4450	4452	4460	4462		
LOCAL	10														
LOOP	10	4184													
MSG	11860	1188	12500	1252	14950	1497	15660	1568	20400	2042	21310	2133	22200	2222	23100
	2312	23980	2400	25000	2502	26860	2688	27780	2780	28950	2897	31640	3166	33330	3335
	34590	3461	36360	3638	38360	3838	40550	4057							
MULT	10	8110													
NEWST	10	8110	1186	1250	1336	1413	1495	1566	1652	1728	1804	1887	1920	1953	1987
	2040	2131	2220	2310	2398	2500	2606	2686	2778	2839	2895	3027	3164	3333	3459
	3636	3735	3836	4055	4121										
NOLOCA	10														
POINTE	10														
POP	10	8110	1241	3449	3632	4341	4541	4542	4825	4826	5007				
PRINTB	10														
PUSH	10	8110	1234	3442	3625	4314	4522	4528	4786	4788	4809	4966			
REPEAT	10	1206	2925	3922	4414										
REPORT	10	8110													
RETURN	10	4236	4240	4296	4464										
ROUTIN	10	4149	4256	4305	4366	4404									
SAVR14	10														
SCOPE	7060	1192	1255	1339	1416	1503	1570	1655	1731	1807	1890	1923	1956	1990	2045
	2136	2225	2314	2405	2505	2609	2691	2783	2842	2900	3030	3170	3341	3465	3640
	3738	3840	4060	4124	4487										
SELECT	10														
SETPRI	10	8110	1420	2421	2521	2624	3368	3417	3489	3576	3649	3755	3874		
SETTRA	51210	5130	5131	5132	5133	5135	5137	5138	5139						
SETUP	10	8110	1088												
SETVEC	10	1198	3355	3470											
SKIP	10	8110	1264	1511	1578	2001	2057	2146	2236	2324	2416	2516	2600	2619	2701
	2910	2956	2983	3020	3075	3095	3113	3143	3158	3178	3187	3292	3565	3720	3747
	3820	3851	3947	4068	4113										
SLASH	10	8110													
SPACE	8110														
STARS	10	8110	827	829	847	849	867	869	889	891	910	925	939	950	952
	959	972	1013	1016	1186	1191	1245	1247	1250	1254	1335	1336	1338	1412	1413
	1415	1494	1495	1502	1565	1566	1569	1651	1652	1654	1727	1728	1730	1803	1804
	1806	1881	1884	1987	1889	1920	1922	1953	1955	1986	1987	1989	2032	2036	2040
	2044	2130	2131	2135	2219	2220	2224	2309	2310	2313	2397	2398	2404	2499	2500
	2504	2605	2606	2608	2685	2686	2690	2777	2778	2782	2838	2839	2841	2894	2895
	2899	3026	3027	3029	3163	3164	3169	3332	3333	3340	3458	3459	3464	3635	3636
	3639	3734	3735	3737	3835	3836	3839	4054	4055	4059	4121	4123	4148	4169	4255
	4269	4304	4313	4350	4357	4368	4406	4413	4479	4518	4534	4563	4642	4645	4713



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 156  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

	4742	4781	4838	4892	4956	5023	5100										
STRUCT	10																
SWRSU	10	8110	11110														
TRMTRP	51210																
TYPBIN	10	8110															
TYPDEC	10	8110	4142	4499													
TYPNAM	10	8110	1132														
TYPNUM	10	8110															
TYPOCS	10	8110															
TYPOCT	10	8110	4130	4136	4373	4380	4386	4392	4398	4663							
TYPTXT	10	8110	4126	4132	4138	4369	4375	4382	4388	4394							
UNTIL	10	1230	2939	3924	4456												
UNTILB	10																
WAITMS	10	2433	2534	2562	2585	2645	2667	3044	3054	3126	3378	3422	3512	3588	4218		
WHILE	10	2965	3222														
WHILEB	10																
%ADDON	10	1207	1208	1217	1219	1260	1262	1272	1274	1288	1290	1305	1307	1324	1326		
	1348	1350	1364	1366	1381	1383	1400	1402	1430	1432	1446	1448	1463	1465	1482		
	1484	1507	1509	1521	1523	1537	1539	1553	1555	1574	1576	1587	1589	1603	1605		
	1620	1622	1639	1641	1663	1665	1679	1681	1696	1698	1715	1717	1739	1741	1755		
	1757	1772	1774	1791	1793	1815	1817	1831	1833	1848	1850	1867	1869	1901	1903		
	1934	1936	1966	1968	1996	1998	2011	2013	2050	2052	2075	2077	2095	2097	2115		
	2117	2141	2143	2164	2166	2184	2186	2204	2206	2230	2232	2254	2256	2274	2276		
	2294	2296	2319	2321	2342	2344	2362	2364	2382	2384	2410	2412	2443	2445	2461		
	2463	2470	2472	2487	2489	2510	2512	2544	2546	2568	2570	2591	2593	2614	2616		
	2651	2653	2673	2675	2697	2699	2728	2729	2749	2751	2769	2770	2812	2813	2829		
	2831	2870	2871	2886	2888	2906	2908	2926	2927	2928	2930	2935	2945	2946	2948		
	2966	2967	2969	2971	2973	2974	2976	2989	2994	2996	3011	3013	3066	3068	3084		
	3086	3105	3107	3133	3135	3148	3150	3174	3176	3183	3185	3199	3201	3202	3206		
	3223	3224	3226	3227	3229	3235	3244	3248	3250	3254	3256	3260	3264	3386	3388		
	3390	3392	3398	3428	3430	3522	3524	3526	3528	3535	3560	3562	3596	3598	3600		
	3602	3609	3664	3666	3667	3671	3713	3715	3742	3744	3768	3770	3771	3775	3813		
	3815	3847	3849	3923	3924	3930	3933	3935	3975	3977	4012	4014	4016	4018	4033		
	4035	4064	4066	4095	4096	4103	4105	4151	4185	4186	4188	4190	4195	4202	4204		
	4226	4234	4236	4258	4280	4282	4283	4287	4307	4320	4322	4323	4327	4328	4330		
	4331	4335	4368	4406	4415	4416	4417	4419	4420	4422	4427	4446					
%AND	10	3256															
%BRANC	10	1218	1232	1261	1273	1289	1306	1325	1349	1365	1382	1401	1431	1447	1464		
	1483	1508	1522	1538	1554	1575	1588	1604	1621	1640	1664	1680	1697	1716	1740		
	1756	1773	1792	1816	1832	1849	1868	1902	1935	1967	1997	2012	2051	2076	2096		
	2116	2142	2165	2185	2205	2231	2255	2275	2295	2320	2343	2363	2383	2411	2444		
	2462	2471	2488	2511	2545	2569	2592	2615	2652	2674	2698	2728	2750	2769	2812		
	2830	2870	2887	2907	2929	2933	2941	2943	2947	2968	2972	2975	2989	2995	3012		
	3067	3085	3106	3134	3149	3175	3184	3200	3205	3225	3228	3233	3244	3249	3252		
	3257	3259	3262	3281	3387	3391	3396	3429	3523	3527	3533	3561	3597	3601	3607		
	3665	3670	3714	3724	3743	3769	3774	3814	3825	3848	3926	3928	3934	3976	4013		
	4017	4034	4065	4095	4104	4189	4193	4203	4212	4214	4226	4235	4237	4242	4281		
	4286	4292	4297	4321	4326	4329	4334	4336	4339	4418	4421	4425	4444	4458	4465		
%BRCOD	10	2940	3204	3669	3773	3925	4211	4213	4285	4325	4333						
%CALL	10	1157	2434	2535	2563	2586	2646	2668	2718	2761	2802	2861	3045	3055	3127		
	3379	3423	3501	3513	3589	3655	3676	3689	3761	3780	3792	3856	4087	4219	4428		
%CHECK	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482		
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755		
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115		
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461		
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906	2928		

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 157  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

	2946	2967	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183	3224	3227
	3248	3256	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742	3813	3847	3933
\$CHK1	3975	4012	4016	4033	4064	4103	4188	4202	4234	4417	4420				
	10	1154	1161	1164	1166	1169	1183	1196	1199	1200	1201	1202	1203	1205	1209
	1212	1236	1239	1240	1263	1269	1283	1299	1316	1345	1359	1375	1392	1427	1441
	1457	1474	1510	1516	1532	1548	1577	1584	1598	1614	1631	1660	1674	1690	1706
	1736	1750	1766	1783	1812	1826	1842	1859	1899	1931	1964	2000	2008	2056	2068
	2088	2108	2145	2157	2177	2197	2235	2247	2267	2287	2323	2335	2355	2375	2415
	2428	2440	2455	2481	2515	2528	2541	2555	2579	2618	2635	2639	2662	2700	2705
	2713	2737	2744	2792	2796	2822	2850	2856	2883	2909	2916	2918	2924	2931	2953
	2955	2982	3008	3035	3042	3052	3062	3074	3082	3094	3101	3119	3124	3142	3177
	3186	3192	3194	3199	3202	3207	3210	3213	3216	3219	3221	3231	3276	3278	3289
	3346	3350	3356	3357	3358	3359	3360	3362	3409	3412	3440	3444	3447	3448	3471
	3472	3473	3474	3475	3478	3480	3496	3518	3558	3564	3570	3623	3627	3630	3631
	3660	3664	3667	3686	3698	3719	3746	3765	3768	3771	3790	3802	3819	3850	3880
	3883	3885	3889	3891	3895	3898	3901	3905	3972	4020	4022	4067	4077	4085	4175
	4177	4179	4191	4196	4206	4272	4280	4283	4295	4318	4320	4323	4328	4331	4423
\$CKOP2	4432	4436	4438	4440	4442	4447	4461								
	10	1171	1174	1177	1180	1226	1228	1237	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3237	3239
	3284	3353	3365	3375	3415	3445	3483	3486	3510	3544	3547	3573	3581	3584	3586
	3617	3619	3628	3645	3706	3709	3729	3752	3806	3809	3911	3915	3918	3942	3944
	3965	3968	3979	4000	4003	4007	4027	4037	4073	4082	4224	4274	4288	4290	4361
	4449	4451	4453	4463											
\$CKR6	10	3285	4275												
\$CMND	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906	2928
	2946	2967	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183	3224	3227
	3248	3256	3258	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742	3813	3847
\$COMPA	3933	3975	4012	4016	4033	4064	4103	4188	4202	4234	4417	4420			
	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2728	2749	2769	2812	2829
	2870	2886	2906	2928	2946	2967	2971	2974	2994	3011	3066	3084	3105	3133	3148
	3174	3183	3199	3224	3227	3248	3256	3386	3390	3428	3522	3526	3560	3596	3600
	3664	3713	3742	3768	3813	3847	3933	3975	4012	4016	4033	4064	4095	4103	4188
	4202	4234	4280	4320	4328	4417	4420								
\$COUNT	10	1157	2434	2535	2563	2586	2646	2668	2718	2761	2802	2861	3045	3055	3127
	3379	3423	3501	3513	3589	3655	3676	3689	3761	3780	3792	3856	4087	4219	4428
\$DO	10	2967	3224												
\$ELSE	10														
\$ERRMS	10														
\$EXIFA	10														
\$EXIFO	10														
\$EXIF2	10	4211													
\$EXIF3	10														
\$GENBR	10	1218	1232	1261	1273	1289	1306	1325	1349	1365	1382	1401	1431	1447	1464
	1483	1508	1522	1538	1554	1575	1588	1604	1621	1640	1664	1680	1697	1716	1740
	1756	1773	1792	1816	1832	1849	1868	1902	1935	1967	1997	2012	2051	2076	2096

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 158  
 CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

	2116	2142	2165	2185	2205	2231	2255	2275	2295	2320	2343	2363	2383	2411	2444
	2462	2471	2488	2511	2545	2569	2592	2615	2652	2674	2698	2728	2750	2769	2812
	2830	2870	2887	2907	2929	2933	2941	2943	2947	2968	2972	2975	2989	2995	3012
	3067	3085	3106	3134	3149	3175	3184	3200	3205	3225	3228	3233	3244	3249	3252
	3257	3259	3262	3281	3387	3391	3396	3429	3523	3527	3533	3561	3597	3601	3607
	3665	3670	3714	3724	3743	3769	3774	3814	3825	3848	3926	3928	3934	3976	4013
	4017	4034	4065	4095	4104	4189	4193	4203	4212	4214	4226	4235	4237	4242	4281
	4286	4292	4297	4321	4326	4329	4334	4336	4339	4418	4421	4425	4444	4458	4465
\$GENTA	10	1207	1223	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2926	2934	2938	2944	2958	2966	2985	2987	2990	3001	3017
	3077	3097	3115	3145	3155	3180	3189	3201	3203	3223	3234	3241	3245	3253	3263
	3271	3273	3282	3397	3403	3405	3434	3534	3539	3541	3567	3608	3613	3615	3666
	3668	3722	3725	3749	3770	3772	3822	3826	3853	3923	3929	3939	3981	4024	4029
	4040	4070	4100	4109	4185	4194	4198	4208	4227	4239	4250	4252	4282	4284	4293
	4300	4301	4322	4324	4330	4332	4337	4340	4345	4346	4401	4402	4415	4426	4434
	4445	4455	4472	4473											
\$IF	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906	2928
	2946	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183	3227	3248	3256
	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742	3813	3847	3933	3975	4012
	4016	4033	4064	4103	4188	4202	4234	4417	4420						
\$IFCOD	10	1217	1231	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463
	1482	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739
	1755	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095
	2115	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443
	2461	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906
	2928	2942	2946	2967	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183
	3224	3227	3248	3256	3258	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742
	3813	3847	3927	3933	3975	4012	4016	4033	4064	4103	4188	4202	4234	4417	4420
	4457														
\$IFCON	10	2728	2769	2812	2870	4095									
\$IFOPR	10	1218	1232	1261	1273	1289	1306	1325	1349	1365	1382	1401	1431	1447	1464
	1483	1508	1522	1538	1554	1575	1588	1604	1621	1640	1664	1680	1697	1716	1740
	1756	1773	1792	1816	1832	1849	1868	1902	1935	1967	1997	2012	2051	2076	2096
	2116	2142	2165	2185	2205	2231	2255	2275	2295	2320	2343	2363	2383	2411	2444
	2462	2471	2488	2511	2545	2569	2592	2615	2652	2674	2698	2728	2750	2769	2812
	2830	2870	2887	2907	2929	2933	2941	2943	2947	2968	2972	2975	2989	2995	3012
	3134	3149	3175	3184	3225	3228	3249	3257	3259	3387	3391	3429	3523	3527	3561
	3597	3601	3714	3743	3814	3848	3928	3934	3976	4013	4017	4034	4065	4095	4104
	4189	4203	4235	4418	4421	4458									
\$LET	10	1154	1161	1164	1166	1169	1171	1174	1177	1180	1183	1196	1199	1200	1201
	1202	1203	1205	1209	1212	1226	1228	1236	1237	1239	1240	1263	1269	1283	1285
	1299	1302	1316	1319	1345	1359	1361	1375	1378	1392	1395	1427	1441	1443	1457
	1460	1474	1477	1510	1516	1518	1532	1535	1548	1550	1577	1584	1598	1600	1614
	1617	1631	1634	1660	1674	1676	1690	1693	1706	1710	1736	1750	1752	1766	1769
	1783	1786	1812	1826	1828	1842	1845	1859	1862	1899	1931	1964	2000	2008	2056
	2068	2072	2088	2092	2108	2112	2145	2157	2161	2177	2181	2197	2201	2235	2247
	2251	2267	2271	2287	2291	2323	2335	2339	2355	2359	2375	2379	2415	2428	2431
	2440	2455	2459	2481	2485	2515	2528	2532	2541	2555	2560	2579	2583	2618	2633

MAINDEC-ZZ-CVDVA-D  
CVDVAD.P11

MACY11  
12-JUL-84 05:04

30A(1052)

12-JUL-84 09:52 PAGE 159  
CROSS REFERENCE TABLE -- MACRO NAMES

	2635	2639	2643	2662	2665	2700	2705	2713	2737	2744	2789	2792	2796	2822	2848
	2850	2856	2883	2909	2914	2916	2918	2924	2931	2936	2953	2955	2982	3008	3035
	3042	3052	3062	3074	3082	3094	3101	3119	3124	3142	3177	3186	3192	3194	3196
	3207	3210	3213	3216	3219	3221	3231	3237	3239	3276	3278	3284	3289	3346	3350
	3353	3356	3357	3358	3359	3360	3362	3365	3375	3409	3412	3415	3440	3444	3445
	3447	3448	3471	3472	3473	3474	3475	3478	3480	3483	3486	3496	3510	3518	3544
	3547	3558	3564	3570	3573	3581	3584	3586	3617	3619	3623	3627	3628	3630	3631
	3645	3660	3686	3698	3706	3709	3719	3729	3746	3752	3765	3790	3802	3806	3809
	3819	3850	3880	3883	3885	3889	3891	3895	3898	3901	3905	3911	3915	3918	3942
	3944	3965	3968	3972	3979	4000	4003	4007	4020	4022	4027	4037	4067	4073	4077
	4082	4085	4175	4177	4179	4191	4196	4206	4224	4272	4274	4288	4290	4295	4318
	4361	4423	4432	4436	4438	4440	4442	4447	4449	4451	4453	4461	4463		
\$LPCNT	10	3199	3664	3768	4280	4320	4328								
\$OPADD	10	1172	1175	1178	1181	1226	1229	1238	2936	3202	3237	3239	3353	3446	3629
	3667	3771	3965	4000	4027	4283	4323	4331	4361	4451	4453	4463			
\$OPAND	10	3285	4275												
\$OPCD1	10	1172	1175	1178	1181	1226	1229	1238	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3202	3237
	3239	3285	3353	3365	3375	3415	3446	3483	3486	3510	3544	3547	3573	3581	3584
	3586	3617	3619	3629	3645	3667	3707	3709	3729	3752	3771	3807	3809	3911	3915
	3918	3942	3944	3965	3969	3979	4000	4004	4008	4027	4037	4073	4082	4224	4275
	4283	4288	4290	4295	4323	4331	4361	4449	4451	4453	4463				
\$OPCD2	10	4288	4295	4449											
\$OPCOD	10	1172	1175	1178	1181	1226	1229	1238	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3202	3237
	3239	3285	3353	3365	3375	3415	3446	3483	3486	3510	3544	3547	3573	3581	3584
	3586	3617	3619	3629	3645	3667	3707	3709	3729	3752	3771	3807	3809	3911	3915
	3918	3942	3944	3965	3969	3979	4000	4004	4008	4027	4037	4073	4082	4224	4275
	4283	4288	4290	4295	4323	4331	4361	4449	4451	4453	4463				
\$OPCOM	10	4295													
\$OPDEF	10	1154	1157	1161	1164	1166	1169	1171	1172	1174	1175	1177	1178	1180	1181
	1183	1196	1199	1200	1201	1202	1203	1205	1209	1212	1217	1218	1226	1228	1229
	1231	1232	1236	1237	1238	1239	1240	1260	1261	1263	1269	1272	1273	1283	1285
	1288	1289	1299	1302	1305	1306	1316	1319	1324	1325	1345	1348	1349	1359	1361
	1364	1365	1375	1378	1381	1382	1392	1395	1400	1401	1427	1430	1431	1441	1443
	1446	1447	1457	1460	1463	1464	1474	1477	1482	1483	1507	1508	1510	1516	1518
	1521	1522	1532	1535	1537	1538	1548	1550	1553	1554	1574	1575	1577	1584	1587
	1588	1598	1600	1603	1604	1614	1617	1620	1621	1631	1634	1639	1640	1660	1663
	1664	1674	1676	1679	1680	1690	1693	1696	1697	1706	1710	1715	1716	1736	1739
	1740	1750	1752	1755	1756	1766	1769	1772	1773	1783	1786	1791	1792	1812	1815
	1816	1826	1828	1831	1832	1842	1845	1848	1849	1859	1862	1867	1868	1899	1901
	1902	1931	1934	1935	1964	1966	1967	1996	1997	2000	2008	2011	2012	2050	2051
	2056	2068	2072	2075	2076	2088	2092	2095	2096	2108	2112	2115	2116	2141	2142
	2145	2157	2161	2164	2165	2177	2181	2184	2185	2197	2201	2204	2205	2230	2231
	2235	2247	2251	2254	2255	2267	2271	2274	2275	2287	2291	2294	2295	2319	2320
	2323	2335	2339	2342	2343	2355	2359	2362	2363	2375	2379	2382	2383	2410	2411
	2415	2428	2431	2434	2435	2436	2437	2440	2443	2444	2455	2459	2461	2462	2470
	2471	2481	2485	2487	2488	2510	2511	2515	2528	2532	2535	2536	2537	2538	2541
	2544	2545	2555	2560	2563	2564	2565	2566	2568	2569	2579	2583	2586	2587	2588
	2589	2591	2592	2614	2615	2618	2633	2635	2639	2643	2646	2647	2648	2649	2651
	2652	2662	2665	2668	2669	2670	2671	2673	2674	2697	2698	2700	2705	2713	2718
	2719	2720	2721	2722	2723	2724	2728	2737	2744	2749	2750	2761	2762	2763	2764

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 160  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

2765	2766	2767	2769	2789	2792	2796	2802	2803	2804	2805	2806	2807	2808	2812
2822	2829	2830	2848	2850	2856	2861	2862	2863	2864	2865	2866	2867	2870	2883
2886	2887	2906	2907	2909	2914	2916	2918	2924	2928	2929	2931	2933	2936	2940
2941	2942	2943	2946	2947	2953	2955	2967	2968	2971	2972	2974	2975	2982	2989
2994	2995	3008	3011	3012	3035	3042	3045	3046	3047	3048	3052	3055	3056	3057
3058	3062	3066	3067	3074	3082	3084	3085	3094	3101	3105	3106	3119	3124	3127
3128	3129	3130	3133	3134	3142	3148	3149	3174	3175	3177	3183	3184	3186	3192
3194	3196	3199	3200	3202	3204	3205	3207	3210	3213	3216	3219	3221	3224	3225
3227	3228	3231	3233	3237	3239	3244	3248	3249	3252	3256	3257	3258	3259	3262
3276	3278	3281	3284	3285	3286	3287	3289	3346	3350	3353	3356	3357	3358	3359
3360	3362	3365	3375	3379	3380	3381	3382	3386	3387	3390	3391	3396	3409	3412
3415	3423	3424	3425	3426	3428	3429	3440	3444	3445	3446	3447	3448	3471	3472
3473	3474	3475	3478	3480	3483	3486	3496	3501	3502	3503	3504	3505	3506	3507
3510	3513	3514	3515	3516	3518	3522	3523	3526	3527	3533	3544	3547	3558	3560
3561	3564	3570	3573	3581	3584	3586	3589	3590	3591	3592	3596	3597	3600	3601
3607	3617	3619	3623	3627	3628	3629	3630	3631	3645	3655	3656	3657	3660	3664
3665	3667	3669	3670	3676	3677	3678	3679	3680	3681	3682	3686	3689	3690	3691
3692	3693	3694	3695	3698	3706	3707	3709	3713	3714	3719	3724	3729	3742	3743
3746	3752	3761	3762	3763	3765	3768	3769	3771	3773	3774	3780	3781	3782	3783
3784	3785	3786	3790	3792	3793	3794	3795	3796	3797	3798	3802	3806	3807	3809
3813	3814	3819	3825	3847	3848	3850	3856	3857	3858	3880	3883	3885	3889	3891
3895	3898	3901	3905	3911	3915	3918	3925	3926	3927	3928	3933	3934	3942	3944
3965	3968	3969	3972	3975	3976	3979	4000	4003	4004	4007	4008	4012	4013	4016
4017	4020	4022	4027	4033	4034	4037	4064	4065	4067	4073	4077	4082	4085	4087
4088	4089	4090	4091	4092	4093	4095	4103	4104	4175	4177	4179	4188	4189	4191
4193	4196	4202	4203	4206	4211	4212	4213	4214	4219	4220	4221	4222	4224	4226
4234	4235	4237	4241	4242	4251	4253	4272	4274	4275	4276	4277	4280	4281	4283
4285	4286	4288	4290	4292	4295	4297	4302	4318	4320	4321	4323	4325	4326	4328
4329	4331	4333	4334	4336	4339	4347	4361	4403	4417	4418	4420	4421	4423	4425
4428	4432	4436	4438	4440	4442	4444	4447	4449	4451	4453	4457	4458	4461	4463
4465	4474													

\$OPEQU 1#  
\$OPNAN 1#  
\$OPNEG 1#  
\$OPNOR 1#  
\$OPNOT 1#  
2291 2339 2379 2431 2485 2532 2583 2633 2665 3365 3415 3486 3544 3547 3573  
3581 3617 3619 3707 3709 3729 3752 3807 3809 3942 3944 3969 3979 4004 4008  
4037  
\$OPOR 1# 1285 1319 1361 1395 1443 1477 1535 1600 1634 1676 1710 1752 1786 1828  
1862 2092 2181 2271 2359 2459 2560 2643 2789 2848 2914 3196 3375 3483 3510  
3584 3586 3645 3911 3915 3918 4073 4082 4290  
\$OPROT 1# 4449  
\$OPRO 1# 1154 1161 1164 1166 1169 1183 1196 1199 1200 1201 1202 1203 1205 1209  
1212 1236 1239 1240 1263 1269 1283 1299 1316 1345 1359 1375 1392 1427 1441  
1457 1474 1510 1516 1532 1548 1577 1584 1598 1614 1631 1660 1674 1690 1706  
1736 1750 1766 1783 1812 1826 1842 1859 1899 1931 1964 2000 2008 2056 2068  
2088 2108 2145 2157 2177 2197 2235 2247 2267 2287 2323 2335 2355 2375 2415  
2428 2440 2455 2481 2515 2528 2541 2555 2579 2618 2635 2639 2662 2700 2705  
2713 2737 2744 2792 2796 2822 2850 2856 2883 2909 2916 2918 2924 2931 2953  
2955 2982 3008 3035 3042 3052 3062 3074 3082 3094 3101 3119 3124 3142 3177  
3186 3192 3194 3199 3207 3210 3213 3216 3219 3221 3231 3276 3278 3289 3346  
3350 3356 3357 3358 3359 3360 3362 3409 3412 3440 3444 3447 3448 3471 3472  
3473 3474 3475 3478 3480 3496 3518 3558 3564 3570 3623 3627 3630 3631 3660  
3664 3686 3698 3719 3746 3765 3768 3790 3802 3819 3850 3880 3883 3885 3889  
3891 3895 3898 3901 3905 3972 4020 4022 4067 4077 4085 4175 4177 4179

	4196	4206	4272	4280	4318	4320	4328	4423	4432	4436	4438	4440	4442	4447	4461
\$OPR1	10	3202	3667	3771	4283	4295	4323	4331							
\$OPR2	10	1171	1174	1177	1180	1226	1228	1237	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3237	3239
	3284	3353	3365	3375	3415	3445	3483	3486	3510	3544	3547	3573	3581	3584	3586
	3617	3619	3628	3645	3706	3709	3729	3752	3806	3809	3911	3915	3918	3942	3944
	3965	3968	3979	4000	4003	4007	4027	4037	4073	4082	4224	4274	4288	4290	4361
	4449	4451	4453	4463											
\$OPSHF	10	4288													
\$OPSUB	10	4224													
\$OPSWB	10														
\$OPXOR	10														
\$OR	10														
\$PUT	10	2435	2536	2564	2587	2647	2669	2719	2762	2803	2862	3046	3056	3128	3380
	3424	3502	3514	3590	3677	3690	3781	3793	4088	4220					
\$STRUC	10														
\$SUBON	10	1223	1231	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2934	2938	2940	2958	2985	2987	2989	2990	3001	3017	3077
	3097	3115	3145	3155	3180	3189	3203	3206	3234	3241	3244	3245	3253	3263	3271
	3273	3281	3282	3397	3403	3405	3434	3534	3539	3541	3567	3608	3613	3615	3668
	3671	3722	3724	3725	3749	3772	3775	3822	3825	3826	3853	3925	3939	3981	4024
	4029	4040	4070	4100	4109	4194	4198	4208	4226	4227	4239	4250	4284	4287	4292
	4293	4300	4324	4327	4332	4335	4336	4337	4339	4340	4345	4401	4426	4434	4445
	4455	4457	4472												
\$THEN	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906	2928
	2946	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183	3227	3248	3258
	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742	3813	3847	3933	3975	4012
	4016	4033	4064	4103	4188	4202	4234	4417	4420						
\$TILA	10														
\$TILO	10														
\$UNTL2	10	2940	3925												
\$UNTL3	10														
\$WHILE	10	2966	3223												
\$\$CMRE	9700														
\$\$CMTM	9700														
\$\$DEFA	10														
\$\$ENDS	10														
\$\$ERRO	10														
\$\$ESCA	10	8110													
\$\$GEN	10	1207	1223	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2926	2934	2938	2944	2958	2966	2985	2987	2990	3001	3017
	3077	3097	3115	3145	3155	3180	3189	3201	3203	3223	3234	3241	3245	3253	3263

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 162  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

	3271	3273	3282	3397	3403	3405	3434	3534	3539	3541	3567	3608	3613	3615	3666
	3668	3722	3725	3749	3770	3772	3822	3826	3853	3923	3929	3939	3981	4024	4029
	4040	4070	4100	4109	4150	4185	4194	4198	4208	4227	4239	4250	4252	4257	4282
	4284	4293	4300	4301	4306	4322	4324	4330	4332	4337	4340	4345	4346	4367	4401
	4402	4405	4415	4426	4434	4445	4455	4472	4473						
##GETS	10	1223	1231	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2933	2934	2938	2940	2958	2985	2987	2989	2990	3001	3017
	3077	3097	3115	3145	3155	3180	3189	3203	3206	3233	3234	3241	3244	3245	3252
	3253	3262	3263	3271	3273	3281	3282	3396	3397	3403	3405	3434	3533	3534	3539
	3541	3567	3607	3608	3613	3615	3668	3671	3722	3724	3725	3749	3772	3775	3822
	3825	3826	3853	3925	3939	3981	4024	4029	4040	4070	4100	4109	4193	4194	4198
	4208	4211	4226	4227	4239	4250	4284	4287	4292	4293	4300	4324	4327	4332	4335
	4336	4337	4339	4340	4345	4401	4425	4426	4434	4444	4445	4455	4457	4472	
##GETT	10	2933	3233	3252	3262	3396	3533	3607	4193	4211	4425	4444			
##LPCN	10	3202	3667	3771	4283	4323	4331								
##NEWT	10	8110	1186	1250	1336	1413	1495	1566	1652	1728	1804	1887	1920	1953	1987
	2040	2131	2220	2310	2398	2500	2606	2686	2778	2839	2895	3027	3164	3333	3459
	3636	3735	3836	4055	4121										
##POP	10	1223	1231	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2934	2938	2940	2958	2985	2987	2989	2990	3001	3017	3077
	3097	3115	3145	3155	3180	3189	3203	3206	3234	3241	3244	3245	3253	3263	3271
	3273	3281	3282	3397	3403	3405	3434	3534	3539	3541	3567	3608	3613	3615	3668
	3671	3722	3724	3725	3749	3772	3775	3822	3825	3826	3853	3925	3939	3981	4024
	4029	4040	4070	4100	4109	4194	4198	4208	4226	4227	4239	4250	4284	4287	4292
	4293	4300	4324	4327	4332	4335	4336	4337	4339	4340	4345	4401	4426	4434	4445
	4455	4457	4472												
##PUSH	10	1207	1208	1217	1219	1260	1262	1272	1274	1288	1290	1305	1307	1324	1326
	1348	1350	1364	1366	1381	1383	1400	1402	1430	1432	1446	1448	1463	1465	1482
	1484	1507	1509	1521	1523	1537	1539	1553	1555	1574	1576	1587	1589	1603	1605
	1620	1622	1639	1641	1663	1665	1679	1681	1696	1698	1715	1717	1739	1741	1755
	1757	1772	1774	1791	1793	1815	1817	1831	1833	1848	1850	1867	1869	1901	1903
	1934	1936	1966	1968	1996	1998	2011	2013	2050	2052	2075	2077	2095	2097	2115
	2117	2141	2143	2164	2166	2184	2186	2204	2206	2230	2232	2254	2256	2274	2276
	2294	2296	2319	2321	2342	2344	2362	2364	2382	2384	2410	2412	2443	2445	2461
	2463	2470	2472	2487	2489	2510	2512	2544	2546	2568	2570	2591	2593	2614	2616
	2651	2653	2673	2675	2697	2699	2728	2729	2749	2751	2769	2770	2812	2813	2829
	2831	2870	2871	2886	2888	2906	2908	2926	2927	2928	2930	2935	2946	2948	2966
	2967	2969	2971	2973	2974	2976	2989	2994	2996	3011	3013	3066	3068	3084	3086
	3105	3107	3133	3135	3148	3150	3174	3176	3183	3185	3199	3201	3202	3206	3223
	3224	3226	3227	3229	3235	3244	3248	3250	3254	3256	3260	3264	3386	3388	3390
	3392	3398	3428	3430	3522	3524	3526	3528	3535	3560	3562	3596	3598	3600	3602
	3609	3664	3666	3667	3671	3713	3715	3742	3744	3768	3770	3771	3775	3813	3815
	3847	3849	3923	3924	3933	3935	3975	3977	4012	4014	4016	4018	4033	4035	4064
	4066	4095	4096	4103	4105	4151	4185	4186	4188	4190	4195	4202	4204	4226	4234
	4236	4258	4280	4282	4283	4287	4307	4320	4322	4323	4327	4328	4330	4331	4335
	4368	4406	4415	4416	4417	4419	4420	4422	4427	4446					
##SELE	10														
##SET	51210	5130	5131	5132	5133	5135	5137	5138	5139						

##SETH	11270														
##SETS	10	1207	1208	1217	1219	1260	1262	1272	1274	1288	1290	1305	1307	1324	1326
	1348	1350	1364	1366	1381	1383	1400	1402	1430	1432	1446	1448	1463	1465	1482
	1484	1507	1509	1521	1523	1537	1539	1553	1555	1574	1576	1587	1589	1603	1605
	1620	1622	1639	1641	1663	1665	1679	1681	1696	1698	1715	1717	1739	1741	1755
	1757	1772	1774	1791	1793	1815	1817	1831	1833	1848	1850	1867	1869	1901	1903
	1934	1936	1966	1968	1996	1998	2011	2013	2050	2052	2075	2077	2095	2097	2115
	2117	2141	2143	2164	2166	2184	2186	2204	2206	2230	2232	2254	2256	2274	2276
	2294	2296	2319	2321	2342	2344	2362	2364	2382	2384	2410	2412	2443	2445	2461
	2463	2470	2472	2487	2489	2510	2512	2544	2546	2568	2570	2591	2593	2614	2616
	2651	2653	2673	2675	2697	2699	2728	2729	2749	2751	2769	2770	2812	2813	2829
	2831	2870	2871	2886	2888	2906	2908	2926	2927	2928	2930	2935	2946	2948	2966
	2967	2969	2971	2973	2974	2976	2989	2994	2996	3011	3013	3066	3068	3084	3086
	3105	3107	3133	3135	3148	3150	3174	3176	3183	3185	3197	3201	3202	3206	3223
	3224	3226	3227	3229	3235	3244	3248	3250	3254	3256	3260	3264	3386	3388	3390
	3392	3398	3428	3430	3522	3524	3526	3528	3535	3560	3562	3596	3598	3600	3602
	3609	3664	3666	3667	3671	3713	3715	3742	3744	3768	3770	3771	3775	3813	3815
	3847	3849	3923	3924	3933	3935	3975	3977	4012	4014	4016	4018	4033	4035	4064
	4066	4095	4096	4103	4105	4151	4185	4186	4188	4190	4195	4202	4204	4226	4234
	4236	4258	4280	4282	4283	4287	4307	4320	4322	4323	4327	4328	4330	4331	4335
	4368	4406	4415	4416	4417	4419	4420	4422	4427	4446					
##SETT	10														
##SKIP	10	8110	1264	1511	1578	2001	2057	2146	2236	2324	2416	2516	2600	2619	2701
	2910	2956	2983	3020	3075	3095	3113	3143	3158	3178	3187	3292	3565	3720	3747
	3820	3851	3947	4068	4113										
.EQUAT	10	701													
.HEADE	10	679													
.KT11	10														
.SETUP	10	1088													
.SWRHI	10	689													
.SWRLO	7000														
.\$ACT1	10	937													
.\$APT8	10	10140													
.\$APTH	10	948													
.\$APTY	10	4779													
.\$ASTA	10														
.\$CATC	10	926													
.\$CMTA	10	970													
.\$DB2D	10														
.\$DB20	10														
.\$DIV	10														
.\$EOP	10	4477													
.\$ERRO	10	4836													
.\$ERRT	10														
.\$MULT	10														
.\$POME	10	4516													
.\$RAND	10														
.\$RDDE	10														
.\$RDOC	10														
.\$READ	10	4640													
.\$R2AZ	10														
.\$SAVE	10														
.\$SB2D	10														
.\$SB20	10														
.\$SCOP	10	4890													
.\$SIZE	10														



MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 164  
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- MACRO NAMES

.\$SUPR	10	
.\$TRAP	10	5098
.\$TYPB	10	
.\$TYPD	10	4954
.\$TYPE	10	4561
.\$TYPO	10	5021
.\$4OCA	10	
.1170	10	

. ABS. 015406 000

ERRORS DETECTED: 0

CVDVAD.CVDVAD/SOL/CRF=CVDVAD.SML,CVDVAD.MAC,CVDVAD.P11  
RUN-TIME: 84 90 6 SECONDS  
RUN-TIME RATIO: 287/181=1.5  
CORE USED: 43K (85 PAGES)