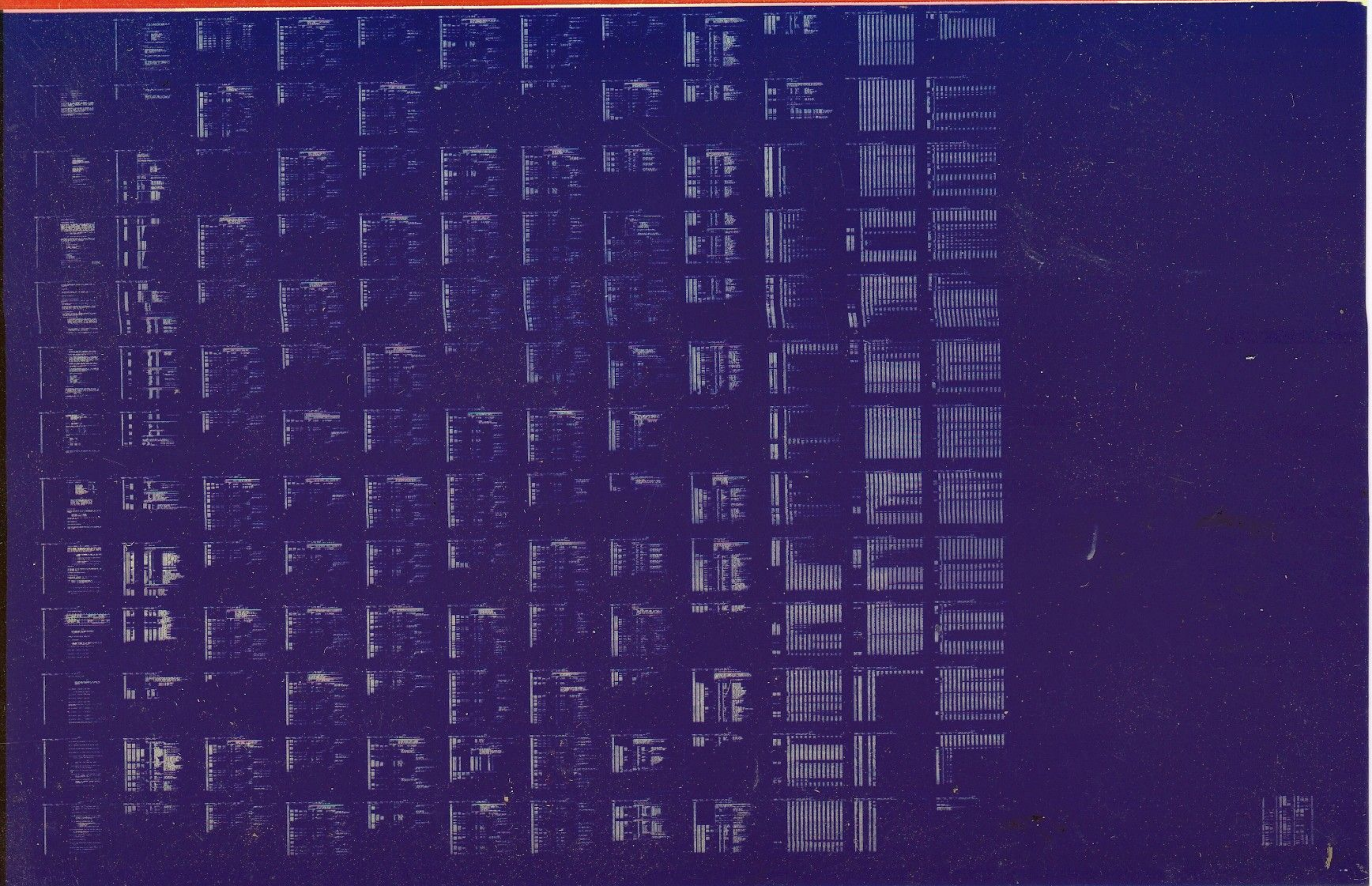


# DLV11-E

OFFLINE TEST  
MD-11-DVDVA-A

EP-DVDVA-A-DL-A  
COPYRIGHT © 1977  
FICHE 1 OF 1

JUN 1977  
**digital**  
MADE IN USA





801

A:NDCC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 3  
JVCVAA.P11 06-MAY-77 15:29

SEQ 0001

1  
2  
3

.REM 2

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  
34  
35  
36

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVDVA-A-D  
PRODUCT NAME: DLV11-E OFFLINE TEST  
PRODUCT DATE: APRIL, 1977  
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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1977 DIGITAL EQUIPMENT CORPORATION

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  
70  
71  
72  
73  
74

TABLE 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.
5.0	SUMMARY OF TESTS AND SPECIAL SUBROUTINES

75 1.0 GENERAL PROGRAM INFORMATION.  
 76 -----  
 77

78 1.1 PROGRAM PURPOSE (ABSTRACT).  
 79

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

89  
 90 175610 -FIRST SERIAL LINE ADDRESS OF 16 CONSECUTIVE  
 91 SERIAL LINE DEVICES.  
 92

93  
 94 300 - VECTOR FOR FIRST OF 16 DEVICES.  
 95

96 THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 4K OF  
 97 MEMORY AND A DLV11-E (LSI-BUS) MODULE. IT CAN RUN UNDER XXDP,  
 98 APT AND ACT MONITORS, AND ON PROCESSORS WITH NO HARDWARE  
 99 SWITCH REGISTER. POWER FAIL IS SUPPORTED.  
 100

101 1.2 SYSTEM REQUIREMENTS.  
 102

103 1. HARDWARE REQUIREMENTS:

104 ANY PDP-11 FAMILY PROCESSOR  
 105 4K MEMORY - MINIMUM  
 106 H315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)  
 107

108 SOFTWARE REQUIREMENTS:  
 109

110 THIS DIAGNOSTIC IS DESIGNED TO RUN IN ANY OF THE  
 111 FOLLOWING WAYS:

112 STAND ALONE  
 113 WITH APT MONITOR  
 114 WITH ACT MONITOR  
 115 WITH XXDP MONITOR (CHAINABLE)  
 116

117 1.3 RELATED DOCUMENTS AND STANDARDS.  
 118

119 DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS  
 120 APT  
 121 ACT  
 122 SYSMAC  
 123

124 175-003-009-02  
 125 MD-11-DZZMA  
 126 AUTOCAT-11-QZAUB  
 127 MD-11-DZQAC  
 128

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES.

129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183

NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT THE PROCESSOR, MEMORY, AND BUS ARE ASSUMED TO BE FULLY OPERATIONAL.

### 1.5 ASSUMPTIONS.

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED LOCATION 'SUSWR' AND 'SDEVN' TO THE PROPER VALUES.

### 2.0 OPERATING INSTRUCTIONS.

-----

#### 2.1 LOADING AND STARTING PROCEDURES.

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED MEDIA.

THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING ADDRESS. 200 FOR START AND RESTART.

THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING SWITCH 8 IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL) IN THE LOWER BYTE. (NOTE: ALL TESTS PREVIOUS TO THE SELECTED ONE ARE EXECUTED IN QUICK VERIFY MODE.)

#### 2.2 SPECIAL ENVIRONMENTS.

THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING UDER APT,ACT,XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE PROCEDURES MANUAL AND SYSMAC PACKAGE.

#### 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

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

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) A RUBOUT WILL DELETE THE LAST INPUT VALUE AND WILL DELIMIT ALL DELETED CHARACTERS BETWEEN BACK SLASHES.
  - C) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.
  - D) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED AND WILL WAIT FOR THE OPERATOR TO ENTER THE "SWREG" DATA AGAIN USING VALID CHARACTERS.

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)

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.

HO1

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  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280

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! ! 15 !	!UNIT! ! #1 !	!UNIT! ! #0 !

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

OPTIONS

LOCATION \$USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION.



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

THE DEFAULT VALUE OF SUSWR 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 (H3FS)	1 = YES
14	FR AND FD JUMPERS IN	1 = YES
15	(NOT DEFINED)	

NOTE

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.

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- 2 MINUTE  
 ADDITIONAL PASSES 2 MINUTES  
 ADDITIONAL DEVICES 2 MINUTES

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

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROLEDURE.

SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4-K OF MEMORY THE ERROR TYPEOUT IS VERY BRIEF. THE FORMAT OF THE ERROR TYPEOUT IS AS FOLLOWS:

336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388

TEST#+++++,ERROR#+++++,PC=+++++,ADDRESS=+++++,VECTOR=+++++

WHERE ALL VALUES TYPED ARE OCTAL EXCEPT THE ERROR NUMBER, WHICH IS DECIMAL. THE ADDRESS AND VECTOR REFER TO THE FAILING SLU'S FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED. BITS 15,13,10 AND 9 OF THE SWITCH REGISTER CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 - CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTINE. CONTINUEING THE PROGRAM CAUSES IT TO PROCEED.

BIT 13 - DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 - CAUSES THE BELL TO RING.

BIT 9 - CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING OF TEST TO ERROR.

THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.

3.2 ERROR HALTS.  
-----

THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE, AND IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER IS A ONE WHEN AN ERROR OCCURS.

4.0 PERFORMANCE AND PROGRESS REPORTS.  
-----

4.1 PERFORMANCE REPORTS.

AS EACH DEVICE COMPLETES ONE PASS OF THE DIAGNOSTIC THE FOLLOWING WILL BE TYPED:

CSR:+++++,VECTOR:+++++,ERRORS:+++++

WHERE: 'CSR:+++++' IS THE DEVICE CSR UNDER TEST  
'VECTOR:++' IS THE ASSOCIATED VECTOR  
AND 'ERRORS:++' IS THE TOTAL NUMBER OF ERRORS ON THIS DEVICE ON THIS PASS.

NOTE

THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.

AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS TYPED:

"ENDPASS#+++++."

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												
	R	ERR	ERR	ERR												
TCSR:	PROGRAMMABLE BAUD				PBR				XMIT	XMIT				MAIN	BREA	
	RATE SELECT				ENAB				RDY	IE				T	K	
TBUF:																

NOTE

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

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

389  
390  
391  
392  
393  
394  
395  
396  
397  
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  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495

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.

TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET  
-----

TEST 10 DATAIE - RCSR5 SET, CLEAR, RESET  
-----

TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET  
-----

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.

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  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547

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 H315.

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

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

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 CHAR 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  
---- --

NO1

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  
593  
594  
595  
596  
597  
598  
599  
600

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

TEST 36 TEST DATA THROUGH CABLE

TEST 37 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE  
MODE.

TEST 40 TEST BREAK GENERATION LOGIC TRANSMIT KNOWN CHAR  
WITH BREAK SET AND COMPARE RECEIVED WITH 0.

TEST 41 NOT A TEST - SEND BACK TO LOOP



601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654

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  
 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.

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 MICRO SEC.

SERVICE ROUTINE: INTSRV  
-----

THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT

655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668

'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.

```

669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724

```

```

TITLE MAINDEC-11-DVDVA-A
*COPYRIGHT (C) 1977
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM BY ODES CHOATE
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
*
.SBTTL OPERATIONAL SWITCH SETTINGS
*
*      SWITCH      USE
*      -----      -----
*      15          HALT ON ERROR
*      14          LOOP ON TEST
*      13          INHIBIT ERROR TYPEOUTS
*      11          INHIBIT ITERATIONS
*      10          BELL ON ERROR
*      9           LOOP ON ERROR
*      8           LOOP ON TEST IN SWR<7:0>
*
.SBTTL BASIC DEFINITIONS
*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE     ;;BASIC DEFINITION OF SCOPE CALL
*
*MISCELLANEOUS DEFINITIONS
HT= 11              ;;CODE FOR HORIZONTAL TAB
LF= 12              ;;CODE FOR LINE FEED
CR= 15              ;;CODE FOR CARRIAGE RETURN
CRLF= 200           ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776         ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774     ;;STACK LIMIT REGISTER
PIRQ= 177772       ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570       ;;HARDWARE SWITCH REGISTER
DDISP= 177570      ;;HARDWARE DISPLAY REGISTER
*
*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0             ;;GENERAL REGISTER
R1= %1             ;;GENERAL REGISTER
R2= %2             ;;GENERAL REGISTER
R3= %3             ;;GENERAL REGISTER
R4= %4             ;;GENERAL REGISTER
R5= %5             ;;GENERAL REGISTER
R6= %6             ;;GENERAL REGISTER
R7= %7             ;;GENERAL REGISTER
SP= %6             ;;STACK POINTER
PC= %7             ;;PROGRAM COUNTER
*
*PRIORITY LEVEL DEFINITIONS
PRO= 0             ;;PRIORITY LEVEL 0

```

```

00110C
000011
000012
000015
000200
177776
177774
177772
177570
177570
000000
000001
000002
000003
000004
000005
000006
000007
000006
000007
000000

```



725	000040	PR1=	40	::	PRIORITY LEVEL	1
726	000100	PR2=	100	::	PRIORITY LEVEL	2
727	000140	PR3=	140	::	PRIORITY LEVEL	3
728	000200	PR4=	200	::	PRIORITY LEVEL	4
729	000240	PR5=	240	::	PRIORITY LEVEL	5
730	000300	PR6=	300	::	PRIORITY LEVEL	6
731	000340	PR7=	340	::	PRIORITY LEVEL	7

.\*"SWITCH REGISTER" SWITCH DEFINITIONS

733		SW15=	100000
734	100000	SW14=	40000
735	040000	SW13=	20000
736	020000	SW12=	10000
737	010000	SW11=	4000
738	004000	SW10=	2000
739	002000	SW09=	1000
740	001000	SW08=	400
741	000400	SW07=	200
742	000200	SW06=	100
743	000100	SW05=	40
744	000040	SW04=	20
745	000020	SW03=	10
746	000010	SW02=	4
747	000004	SW01=	2
748	000002	SW00=	1
749	000001	.EQUIV	SW09, SW9
750		.EQUIV	SW08, SW8
751		.EQUIV	SW07, SW7
752		.EQUIV	SW06, SW6
753		.EQUIV	SW05, SW5
754		.EQUIV	SW04, SW4
755		.EQUIV	SW03, SW3
756		.EQUIV	SW02, SW2
757		.EQUIV	SW01, SW1
758		.EQUIV	SW00, SW0

.\*DATA BIT DEFINITIONS (BIT00 TO BIT15)

760		BIT15=	100000
761		BIT14=	40000
762	100000	BIT13=	20000
763	040000	BIT12=	10000
764	020000	BIT11=	4000
765	010000	BIT10=	2000
766	004000	BIT09=	1000
767	002000	BIT08=	400
768	001000	BIT07=	200
769	000400	BIT06=	100
770	000200	BIT05=	40
771	000100	BIT04=	20
772	000040	BIT03=	10
773	000020	BIT02=	4
774	000010	BIT01=	2
775	000004	BIT00=	1
776	000002	.EQUIV	BIT09, BIT9
777	000001	.EQUIV	BIT08, BIT8
778		.EQUIV	BIT07, BIT7
779			
780			

```

781 .EQUIV BIT06,BIT6
782 .EQUIV BIT05,BIT5
783 .EQUIV BIT04,BIT4
784 .EQUIV BIT03,BIT3
785 .EQUIV BIT02,BIT2
786 .EQUIV BIT01,BIT1
787 .EQUIV BIT00,BIT0

```

```

789 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
790 ERRVEC= 4 ; TIME OUT AND OTHER ERRORS
791 RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS
792 TBITVEC=14 ; "T" BIT
793 TRTVEC= 14 ; TRACE TRAP
794 BPTVEC= 14 ; BREAKPOINT TRAP (BPT)
795 IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
796 PWRVEC= 24 ; POWER FAIL
797 EMTVEC= 30 ; EMULATOR TRAP (EMT) **ERROR**
798 TRAPVEC=34 ; "TRAP" TRAP
799 TKVEC= 60 ; TTY KEYBOARD VECTOR
800 TPVEC= 64 ; TTY PRINTER VECTOR
801 PIRQVEC=240 ; PROGRAM INTERRUPT REQUEST VECTOR

```

```

802 ILLMEM= 4
803 ADRS= R1
804 GOOD= R2
805 BAD= R3
806 REGISTER=R1
807 BIT= R2
808 FUNCT= R3
809 LEAD= R2
810 FOLLOW= R4
811 DLADDR= 175610

```

```

813 ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
814 SET= -1
815 CLR= 0

```

```

818 ;*****
819 ; RCSR REGISTER BIT NAMES
820 ;*****
821 DATAINT= BIT15 ; DATASET INTERRUPT
822 RING= BIT14 ; RINGING SIGNAL INDICATOR
823 CLRSEND= BIT13 ; CLEAR TO SEND FROM DATASET
824 CARDET= BIT12 ; CARRIER DETECT
825 RCVRACT= BIT11 ; RECEIVER ACTIVE INDICATOR
826 SECREC= BIT10 ; SECONDARY RECEIVE
827 ; UNUSED BIT09
828 ; UNUSED BIT08
829 RCVRDONE= BIT07 ; RECEIVER DONE
830 RCVRIE= BIT06 ; RECEIVER INTERRUPT ENABLE
831 DATAIE= BIT05 ; DATASET INTERRUPT ENABLE
832 ; UNUSED BIT04
833 SECXMIT= BIT03 ; SECONDARY TRANSMIT DATA
834 REQSEND= BIT02 ; REQUEST TO SEND
835 DTR= BIT01 ; DATA TERMINAL READY
836 ; UNUSED BIT00

```





893	000040	TDATA5=	BIT05	:	!	\ TRANSMITTER DATA BUFFER
894	000020	TDATA4=	BIT04	:	:	
895	00001C	TDATA3=	BIT03	:	:	
896	000004	TDATA2=	BIT02	:	:	
897	000002	TDATA1=	BIT01	:	:	
898	000001	TDATA0=	BIT00	:	:	

```

;*****
; FLAG BITS TO BE USE OR CLEARED IN SUSWR.

```

904	000017	DATA =	17
905	000020	PARITY =	20
906	000040	EVENODD =	40
907	000100	COMSPD =	100
908	000200	PBR =	200

```

; BAUDE MUST BE ON THE UPPER
; BYTE BOUNDARY OF SUSWR.--4 BITS

```

912	007400	BAUD =	7400
913	010000	BRK =	10000
914	020000	CABLE =	20000
915	040000	FRFD =	40000

```

;*****
.SBTTL TRAP CATCHER

```

```

;*=0
;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

```

923	000174	. =174	
924	000174	DISPREG: .WORD 0	::SOFTWARE DISPLAY REGISTER
925	000176	SWREG: .WORD 0	::SOFTWARE SWITCH REGISTER
926		.SBTTL STARTING ADDRESS(ES)	
927	00020C	000137	0013J6

```

JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

```

928  
929  
930  
931  
932  
933  
934 000046  
935  
936 000052  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946 000024  
947  
948 000044  
949  
950  
951  
952  
953  
954 001000  
955 001000 000000  
956 001002 001174  
957 001004 000005  
958 001006 000055  
959 001010 000036  
960 001012 000030

```
.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
    $SVPC=.          ;SAVE PC
    =46
    $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOF
    =52
    .WORD 0         ;;2)SET LOC.52 TO ZERO
    =$SVPC          ;; RESTORE PC
.=1000
.SBTTL APT PARAMETER BLOCK
;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
    .SX=.           ;SAVE CURRENT LOCATION
    =24            ;SET POWER FAIL TO POINT TO START OF PROGRAM
    200            ;FOR APT START UP
    =44            ;POINT TO APT INDIRECT ADDRESS PNTR.
    $APTHDR        ;POINT TO APT HEADER BLOCK
    =.SX           ;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.
$APTHD:
$SHIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;;AD. ESS OF APT MAILBOX (BITS 0-15)
$STMT: .WORD 5 ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 45. ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 30. ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

961  
962  
963  
964  
965  
966  
967 001100  
968 001100  
969 001100 000000  
970 001102 000  
971 001103 000  
972 001104 000000  
973 001106 000000  
974 001110 000000  
975 001112 000000  
976 001114 000  
977 001115 001  
978 001116 000000  
979 001120 000000  
980 001122 000000  
981 001124 000000  
982 001126 000000  
983 001130 000000  
984 001132 000000  
985 001134 000  
986 001135 000  
987 001136 000000  
988 001140 177570  
989 001142 177570  
990 001144 177560  
991 001146 177562  
992 001150 177564  
993 001152 177566  
994 001154 000  
995 001155 002  
996 001156 012  
997 001157 000  
998 001160 000000  
999 001162 000000  
1000 001164 177607 000377  
1001 001170 077  
1002 001171 015  
1003 001172 000012  
1004  
1005  
1006  
1007  
1008  
1009 001174  
1010 001174 000000  
1011 001176 000000  
1012 001200 000000  
1013 001202 000000  
1014 001204 000000  
1015 001206 000000  
1016 001210 000000

.SBTTL COMMON TAGS

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

SCMTAG: . =1100

;; START OF COMMON TAGS

.WORD 0  
\$STNM: .BYTE 0  
\$ERFLG: .BYTE 0  
\$ICNT: .WORD 0  
\$LPAOR: .WORD 0  
\$LPERR: .WORD 0  
\$ERTTL: .WORD 0  
\$ITEMB: .BYTE 0  
\$ERMAX: .BYTE 1  
\$ERRPC: .WORD 0  
\$GDADR: .WORD 0  
\$BODR: .WORD 0  
\$GDDAT: .WORD 0  
\$BDDAT: .WORD 0  
\$AUTOB: .BYTE 0  
\$INTAG: .BYTE 0  
\$SWR: .WORD DSWR  
DISPLAY: .WORD DDISP  
\$TKS: 177560  
\$TKB: 177562  
\$TPS: 177564  
\$TPB: 177566  
\$NULL: .BYTE 0  
\$FILLS: .BYTE 2  
\$FILLC: .BYTE 12  
\$TPFLG: .BYTE 0  
\$TIMES: 0  
\$ESCAPE: 0  
\$BELL: .ASCIZ <207><377><377>  
\$QUES: .ASCII /?/  
\$CRLF: .ASCII <15>  
\$LF: .ASCIZ <12>

;; CONTAINS THE TEST NUMBER  
;; CONTAINS ERROR FLAG  
;; CONTAINS SUBTEST ITERATION COUNT  
;; CONTAINS SCOPE LOOP ADDRESS  
;; CONTAINS SCOPE RETURN FOR ERRORS  
;; CONTAINS TOTAL ERRORS DETECTED  
;; CONTAINS ITEM CONTROL BYTE  
;; CONTAINS MAX. ERRORS PER TEST  
;; CONTAINS PC OF LAST ERROR INSTRUCTION  
;; CONTAINS ADDRESS OF 'GOOD' DATA  
;; CONTAINS ADDRESS OF 'BAD' DATA  
;; CONTAINS 'GOOD' DATA  
;; CONTAINS 'BAD' DATA  
;; RESERVED--NOT TO BE USED  
;; AUTOMATIC MODE INDICATOR  
;; INTERRUPT MODE INDICATOR  
;; ADDRESS OF SWITCH REGISTER  
;; ADDRESS OF DISPLAY REGISTER  
;; TTY KBD STATUS  
;; TTY KBD BUFFER  
;; TTY PRINTER STATUS REG. ADDRESS  
;; TTY PRINTER BUFFER REG. ADDRESS  
;; CONTAINS NULL CHARACTER FOR FILLS  
;; CONTAINS # OF FILLER CHARACTERS REQUIRED  
;; INSERT FILL CHARS. AFTER A "LINE FEED"  
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)  
;; MAX. NUMBER OF ITERATIONS  
;; ESCAPE ON ERROR ADDRESS  
;; CODE FOR BELL  
;; QUESTION MARK  
;; CARRIAGE RETURN  
;; LINE FEED

\*\*\*\*\*  
.SBTTL APT MAILBOX-ETABLE

\*\*\*\*\*  
.EVEN  
\$MAIL: .WORD  
\$MSGTY: .WORD M.MSGTY  
\$FATAL: .WORD AFATAL  
\$TESTN: .WORD ATESTN  
\$PASS: .WORD APASS  
\$DEVCT: .WORD ADEVCT  
\$UNIT: .WORD AUNIT  
\$MSGAD: .WORD AMSGAD

;; APT MAILBOX  
;; MESSAGE TYPE CODE  
;; FATAL ERROR NUMBER  
;; TEST NUMBER  
;; PASS COUNT  
;; DEVICE COUNT  
;; I/O UNIT NUMBER  
;; MESSAGE ADDRESS

K02

1017	001212	000000	\$MSGLG: .WORD	AMSQLG	:: MESSAGE LENGTH
1018	001214		\$ETABLE:		:: APT ENVIRONMENT TABLE
1019	001214	000	\$ENV: .BYTE	AENV	:: ENVIRONMENT BYTE
1020	001215	000	\$ENVM: .BYTE	AENVM	:: ENVIRONMENT MODE BITS
1021	001216	000000	\$SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
1022	001220	071110	\$USWR: .WORD	AUSWR	:: USER SWITCHES
1023	001222	000000	\$CPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
1024			*		BITS 15-11=CPU TYPE
1025			*		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1026			*		11/70=06, PDQ=07, Q=10
1027			*		BIT 10=REAL TIME CLOCK
1028			*		BIT 9=FLOATING POINT PROCESSOR
1029			*		BIT 8=MEMORY MANAGEMENT
1030	001224	000	\$MAMS1: .BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
1031	001225	000	\$MTYP1: .BYTE	AMTYP1	:: MEM. TYPE, BLK#1
1032			*		MEM. TYPE BYTE -- (HIGH BYTE)
1033			*		900 NSEC CORE=001
1034			*		300 NSEC BIPOLAR=002
1035			*		500 NSEC MOS=003
1036	001226	000000	\$MAOR1: .WORD	AMADR1	:: HIGH ADDRESS, BLK#1
1037			*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1038	001230	000	\$MAMS2: .BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
1039	001231	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
1040	001232	000000	\$MAOR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
1041	001234	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
1042	001235	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
1043	001236	000000	\$MAOR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
1044	001240	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
1045	001241	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
1046	001242	000000	\$MAOR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
1047	001244	000300	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
1048	001246	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
1049	001250	175610	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
1050	001252	000001	\$DEVN: .WORD	ADEVN	:: DEVICE MAP
1051	001254		\$ETEND:		
1052			.MEXIT		



1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067 001254  
1068  
1069 001254 175610  
1070 001256 000300  
1071 001260 175610  
1072 001262 175612  
1073 001264 175614  
1074 001266 175615  
1075 001270 175616  
1076 001272 000000  
1077 001274 000020  
1078 001334 000000

.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  
;\* DH ;;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

M02

MAINDEC-11-DVDVA-A MACY11 27(1006)  
 DVDVA.P11 06-MAY-77 15:29

16-MAY-77 10:36 PAGE 27  
 ERROR POINTER TABLE

SEQ 0025

1079	001336					START:	
1080						.SBTTL	INITIALIZE THE COMMON TAGS
1081						;;CLEAR	THE COMMON TAGS (\$CMTAG) AREA
1082	001336	012706	001100			MOV	\$CMTAG, R6 ;: FIRST LOCATION TO BE CLEARED
1083	001342	005026				CLR	(R6)+ ;: CLEAR MEMORY LOCATION
1084	001344	022706	001140			CMP	\$SWR, R6 ;: DONE?
1085	001350	001374				BNE	-6 ;: LOOP BACK IF NO
1086	001352	012706	001100			MOV	\$STACK, SP ;: SETUP THE STACK POINTER
1087						;;INITIALIZE	A FEW VECTORS
1088	001356	012737	014132	000020		MOV	\$SCOPE, \$IOTVEC ;: IOT VECTOR FOR SCOPE ROUTINE
1089	001364	012737	000340	000022		MOV	\$340, \$IOTVEC+2 ;: LEVEL 7
1090	001372	012737	013732	000030		MOV	\$ERROR, \$EMTVEC ;: EMT VECTOR FOR ERROR ROUTINE
1091	001400	012737	000340	000032		MOV	\$340, \$EMTVEC+2 ;: LEVEL 7
1092	001406	012737	015064	000034		MOV	\$TRAP, \$TRAPVEC ;: TRAP VECTOR FOR TRAP CALLS
1093	001414	012737	000340	000036		MOV	\$340, \$TRAPVEC+2 ;: LEVEL 7
1094	001422	012737	012250	000024		MOV	\$SPWRON, \$PWRVEC ;: POWER FAILURE VECTOR
1095	001430	012737	000340	000026		MOV	\$340, \$PWRVEC+2 ;: LEVEL 7
1096	001436	016767	010520	010510		MOV	\$ENDCT, \$EOPCT ;: SETUP END-OF-PROGRAM COUNTER
1097	001444	005067	177510			CLR	\$TIMES ;: INITIALIZE NUMBER OF ITERATIONS
1098	001450	005067	177506			CLR	\$ESCAPE ;: CLEAR THE ESCAPE ON ERROR ADDRESS
1099	001454	112767	000001	177433		MOVB	\$1, \$ERMAX ;: ALLOW ONE ERROR PER TEST
1100	001462	012767	001462	177416		MOV	\$., \$LPADR ;: INITIALIZE THE LOOP ADDRESS FOR SCOPE
1101	001470	012767	001470	177412		MOV	\$., \$LPERR ;: SETUP THE ERROR LOOP ADDRESS
1102						;;SIZE FOR A	HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1103						;;EQUAL	TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1104	001476	013746	000004			MOV	\$ERRVEC, -(SP) ;: SAVE ERROR VECTOR
1105	001502	012737	001536	000004		MOV	\$64\$, \$ERRVEC ;: SET UP ERROR VECTOR
1106	001510	012767	177570	177422		MOV	\$DSWR, \$SWR ;: SETUP FOR A HARDWARE SWICH REGISTER
1107	001516	012767	177570	177416		MOV	\$DDISP, \$DISPLAY ;: AND A HARDWARE DISPLAY REGISTER
1108	001524	022777	177777	177406		CMP	\$-1, \$SWR ;: TRY TO REFERENCE HARDWARE SWR
1109	001532	001012				BNE	\$65\$ ;: BRANCH IF NO TIMEOUT TRAP OCCURRED
1110							AND THE HARDWARE SWR IS NOT = -1
1111	001534	000403				BR	\$65\$ ;: BRANCH IF NO TIMEOUT
1112	001536	012716	001544		64\$:	MOV	\$65\$, (SP) ;: SET UP FOR TRAP RETURN
1113	001542	000002				RTI	
1114	001544	012767	000176	177366	65\$:	MOV	\$SWREG, \$SWR ;: POINT TO SOFTWARE SWR
1115	001552	012767	000174	177362		MOV	\$DISPREG, \$DISPLAY
1116	001560	012637	000004		66\$:	MOV	(SP)+, \$ERRVEC ;: RESTORE ERROR VECTOR
1117							
1118	001564	005067	177412			CLR	\$PASS ;: CLEAR PASS COUNT
1119	001570	132767	000200	177417		BITB	\$APTSIZE, \$ENVM ;: TEST USER SIZE UNDER APT
1120	001576	001403				BEQ	\$67\$ ;: YES, USE NON-APT SWITCH
1121	001600	012767	001216	177332		MOV	\$SSWREG, \$SWR ;: NO, USE APT SWITCH REGISTER
1122	001606				67\$:		
1123						.SBTTL	TYPE PROGRAM NAME
1124						;;TYPE	THE NAME OF THE PROGRAM IF FIRST PASS
1125	001606	005227	177777			INC	\$-1 ;: FIRST TIME?
1126	001612	001037				BNE	\$68\$ ;: BRANCH IF NO
1127	001614	022737	012214	000042		CMP	\$SENDAD, \$42 ;: ACT-11?
1128	001622	001433				BEQ	\$68\$ ;: BRANCH IF YES
1129	001624	104401	001672			TYPE	\$69\$ ;: TYPE ASCIZ STRING
1130						.SBTTL	GET VALUE FOR SOFTWARE SWITCH REGISTER
1131	001630	005737	000042			TST	\$42 ;: ARE WE RUNNING UNDER XXDP/ACT?
1132	001634	001012				BNE	\$70\$ ;: BRANCH IF YES
1133	001636	126727	177352	000001		CMPB	\$ENV, \$1 ;: ARE WE RUNNING UNDER APT?
1134	001644	001406				BEQ	\$70\$ ;: BRANCH IF YES

N02

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 28  
DVDVAA.P11 06-MAY-77 15:29 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 0026

1135	001646	026727	177266	000176		CMP	SWR, #SWREG	:: SOFTWARE SWITCH REG SELECTED?
1136	001654	001005				BNE	71\$	:: BRANCH IF NO
1137	001656	104406				GTSWR		:: GET SOFT-SWR SETTINGS
1138	001660	000403				BR	71\$	
1139	001662	112767	000001	177244	70\$:	MOVB	#1, \$AUTOB	:: SET AUTO-MODE INDICATOR
1140	001670				71\$:			
1141	001670	000410				BR	68\$	:: GET OVER THE ASCIZ
1142					:: 69\$:	.ASCIZ	<CRLF>*MD-11-DVDVA-A*<CRLF>	
1143	001712				68\$:			

## B03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 29  
 DVDVA.P11 06-MAY-77 15:29 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0027

```

1144 001712          LET INITFLAG := #1
1145 001712 012767 000001 010176  LOOP:  MOV      #1,INITFLAG
1146 001720          ; NO ARGUMENTS--ADRS -> NEXT ADDRESS
1147 001720          ;
1148 001720 004767 010042          ; ADRS+2 -> NEXT VECTOR
1149          ; GET UNIT ADDRESS
1150          DLADD := (ADRS)+
1151 001724          MOV      (ADRS)+,DLADD
1152 001724 012167 177324          ; GET UNIT VECTOR
1153          DLVEC := (ADRS)
1154 001730          MOV      (ADRS),DLVEC
1155 001730 011167 177322          LET      ADRS := DLADD
1156 001734          MOV      DLADD,ADRS
1157 001734 016701 177314          ; RCSR = DLADD + 0
1158          RCSR := DLADD
1159 001740          MOV      DLADD,RCSR
1160 001740 016767 177310 177312  LET      RBUF := DLADD + #2
1161 001746          MOV      DLADD,RBUF
1162 001746 016767 177302 177306  ADD      #2,RBUF
1163 001754 062767 000002 177300  LET      TCSR := DLADD + #4
1164 001762          MOV      DLADD,TCSR
1165 001762 016767 177266 177274  ADD      #4,TCSR
1166 001770 062767 000004 177266  LET      TCSRHI := DLADD + #5
1167 001776          MOV      DLADD,TCSRHI
1168 001776 016767 177252 177262  ADD      #5,TCSRHI
1169 002004 062767 000005 177254  LET      TBUF := DLADD + #6
1170 002012          MOV      DLADD,TBUF
1171 002012 016767 177236 177250  ADD      #6,TBUF
1172 002020 062767 000006 177242  LET      RS := #RSTACK
1173 002026          MOV      #RSTACK,RS
1174 002026 012705 001334          ;;BRESET
1175          RESET
1176 002032 000005

```

```

1177 .....
1178 .....
1179 .....
1180 .....
1181 .....
1182 .....
1183 002034 000004 .....
1184 002036 012767 000002 177114 .....
1185 002044 012767 000001 177126 .....
1186 002052 .....
1187 002052 016701 177176 .....
1188 .....
1189 002056 .....
1190 002056 010146 .....
1191 002060 012701 000004 .....
1192 002064 012721 011574 .....
1193 002070 012711 000340 .....
1194 002074 012601 .....
1195 002076 .....
1196 002076 005067 177170 .....
1197 002102 .....
1198 002102 .....
1199 002102 .....
1200 002102 012767 002110 177000 .....
1201 .....
1202 002110 .....
1203 002110 005067 007466 .....
1204 .....
1205 .....
1206 002114 005711 .....
1207 002116 .....
1208 002116 005767 007460 .....
1209 002122 001401 .....
1210 .....
1211 002124 .....
1212 002124 104001 .....
1213 002126 .....
1214 002126 .....
1215 002126 .....
1216 002126 .....
1217 002126 062767 000002 177136 .....
1218 002134 .....
1219 002134 016701 177114 .....
1220 002140 066701 177126 .....
1221 002144 .....
1222 002144 026727 177122 000010 .....
1223 002152 001353 .....
1224 002154 .....
1225 002154 010146 .....
1226 002156 010246 .....
1227 002160 012701 000004 .....
1228 002164 010102 .....
1229 002166 062702 000002 .....
1230 002172 010221 .....
1231 002174 005011 .....
1232 002176 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
CLR I
REPEAT
S1: BGNSUB
MOV #645, $LPERR ; CLEAR FLAG
LET INTFLAG := #0
CLR INTFLAG
; READ FLAG
TST @ADRS IF INTFLAG NE #0 THEN
TST INTFLAG
BEQ $2 ; FATAL ERROR
; ERRDF 1,,NODL
ERROR I
ENDIF
S2: ENDSUB
LET I := I + #2
LET ADRS := DLADD + I
UNTIL I EQ #8.
CLRVEC ILLMEM
MOV R1, -(SP) ;; PUSH R1 ON STACK
MOV R2, -(SP) ;; PUSH R2 ON STACK
MOV #ILLMEM, R1
MOV R1, R2
ADD #2, R2
MOV R2, (R1)+
CLR (R1)
MOV (SP)+, R2 ;; POP STACK INTO R2

```



003

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 31  
DVDVAA.P11 06-MAY-77 15:29 T1 ADDRESSABILITY

SEQ 0029

1233 002200 012601  
1234  
1235 002202

MOV (SP)+,R1

::POP STACK INTO R1  
;END OF TEST

ENDTST

```

1236 ; *****
1237 * THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
1238 ; *****
1239 ; *****
1240 *TEST 2 BREAK - TCSR0 SET, CLEAR, RESET
1241 * THIS BIT IS THE ONLY ONE IN THIS POSITION
1242 * THAT IS READ AND WRITE.
1243 ; *****
1244 †ST2: SCOPE
1245 002202 000004 MOV #10,$TIMES ;;DO 10 ITERATIONS
1246 002204 012767 000010 176746 MOV #2,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1247 002212 012767 000002 176760
1248 ; SEE IF IT IS CLEAR
1249 002220 ; BGNSUB
1250 002220 012767 002226 176662 MOV #64,$SLPERR
1251
1252 002226 IF #BREAK SETIN @TCSR THEN
1253 002226 032777 000001 177030 BIT #BREAK,@TCSR
1254 002234 001401 BEQ $3
1255 ; BREAK DID NOT RESET IN TCSR
1256 002236 ERRHRD 2,,DIDNOT
1257 002236 104002 ERROR 2
1258 002240 ENDF
1259 002240 $3:
1260 002240 ENDSUB
1261
1262 ; TRY TO SET BREAK BIT
1263 002240 ; BGNSUB
1264 002240 012767 002246 176642 MOV #64,$SLPERR
1265 002246 LET @TCSR := @TCSR SET.BY #BREAK
1266 002246 052777 000001 177010 BIS #BREAK,@TCSR
1267 ; STUCK TO 0
1268 002254 IF #BREAK NOTSETIN @TCSR THEN
1269 002254 032777 000001 177002 BIT #BREAK,@TCSR
1270 002262 001001 BNE $4
1271 ; BREAK DID NOT SET IN TCSR
1272 002264 ERRHRD 3,,DIDNOT
1273 002264 104003 ERROR 3
1274 002266 ENDF
1275 002266 $4:
1276 002266 ENDSUB
1277
1278 ; TRY TO CLEAR A SET BIT
1279 002266 ; BGNSUB
1280 002266 012767 002274 176614 MOV #64,$SLPERR
1281
1282 002274 LET @TCSR := @TCSR CLR.BY #BREAK
1283 002274 042777 000001 176762 BIC #BREAK,@TCSR
1284 ; SHOULD HAVE CLEARED
1285 002302 IF #BREAK SETIN @TCSR THEN
1286 002302 032777 000001 176754 BIT #BREAK,@TCSR
1287 002310 001401 BEQ $5
1288 ; BREAK DID NOT CLEAR IN TCSR
1289 002312 ERRHRD 4,,DIDNOT
1290 002312 104004 ERROR 4
1291 002314 ENDF

```

F03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 33  
DVDVAA.P11 06-MAY-77 15:29 T2 BREAK - TCSR0 SET, CLEAR, RESET

SEQ 0031

```

1292 002314          $5:
1293 002314
1294
1295          ; NOW SEE IF RESET CLEARS IT
1296 002314          BGN SUB
1297 002314 012767 002322 176566      MOV      #64$, $LPERR
1298
1299 002322          LET      @TCSR := @TCSR SET.BY #BREAK
1300 002322 052777 000001 176734      BIS      #BREAK, @TCSR
1301          ; ISSUE BUS RESET
1302          BRESÉT
1303 002330 000005      RESET
1304 002332
1305 002332 032777 000001 176724      BIT      #BREAK, @TCSR
1306 002340 001401      BEQ      $6
1307          ; BREAK DID NOT RESET IN TCSR
1308 002342          ERRHRD 5,, DIDNOT
1309 002342 104005      ERROR  5
1310 002344          ENDIF
1311 002344          $6:
1312 002344          ENDSUB
1313 002344          ENDTST
1314
1315

```

G03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 34  
DVDVA.P11 06-MAY-77 15:29

T2 BREAK - TCSR0 SET, CLEAR, RESET

SEQ 0032

```

1316 .....
1317 .....
1318 .....
1319 .....
1320 002344 000004 .....
1321 002346 012767 000010 176604 .....
1322 002354 012767 000003 176616 .....
1323 .....
1324 ..... ; SEE IF IT IS CLEAR
1325 002362 ..... BGNSUB
1326 002362 012767 002370 176520 MOV #64$,SLPERR
1327 .....
1328 002370 ..... IF #MAINT SETIN @TCSR THEN
1329 002370 032777 000004 176666 BIT #MAINT,@TCSR
1330 002376 001401 BEQ $7
1331 ..... ; MAINT DID NOT RESET IN TCSR
1332 002400 ..... ERRHRD 6,,DIDNOT
1333 002400 104006 ERROR 6
1334 002402 ..... ENDIF
1335 002402 $7: .....
1336 002402 ..... ENDSUB
1337 .....
1338 ..... ; TRY TO SET MAINT BIT
1339 002402 ..... BGNSUB
1340 002402 012767 002410 176500 MOV #64$,SLPERR
1341 002410 ..... LET @TCSR := @TCSR SET.BY #MAINT
1342 002410 052777 000004 176646 BIS #MAINT,@TCSR
1343 ..... ; STUCK TO 0
1344 002416 ..... IF #MAINT NOTSETIN @TCSR THEN
1345 002416 032777 000004 176640 BIT #MAINT,@TCSR
1346 002424 001001 BNE $10
1347 ..... ; MAINT DID NOT SET IN TCSR
1348 002426 ..... ERRHRD 7,,DIDNOT
1349 002426 104007 ERROR 7
1350 002430 ..... ENDIF
1351 002430 $10: .....
1352 002430 ..... ENDSUB
1353 .....
1354 ..... ; TRY TO CLEAR A SET BIT
1355 002430 ..... BGNSUB
1356 002430 012767 002436 176452 MOV #64$,SLPERR
1357 .....
1358 002436 ..... LET @TCSR := @TCSR CLR.BY #MAINT
1359 002436 042777 000004 176620 BIC #MAINT,@TCSR
1360 ..... ; SHOULD HAVE CLEARED
1361 002444 ..... IF #MAINT SETIN @TCSR THEN
1362 002444 032777 000004 176612 BIT #MAINT,@TCSR
1363 002452 001401 BEQ $11
1364 ..... ; MAINT DID NOT CLEAR INTCSR
1365 002454 ..... ERRHRD 8,,DIDNOT
1366 002454 104010 ERROR 8
1367 002456 ..... ENDIF
1368 002456 $11: .....
1369 002456 ..... ENDSUB
1370 .....
1371 ..... ; NOW SEE IF RESET CLEARS IT

```

H03

MAINDEC-11-DVDVA-A MACY11 27(1006)  
DVDVAA.P11 06-MAY-77 15:29

16-MAY-77 10:36 PAGE 35  
T3 MAINT - TCSR2 SET, CLEAR, RESET

SEQ 0033

```
1372 002456                                BGNSUB
1373 002456 012767 002464 176424          MOV    #645,$LPERR
1374                                     LET    @TCSR := @TCSR SET.5/ #MAINT
1375 002464                                     : ISSUE BUS RESET
1376 002464 052777 000004 176572          BIS    #MAINT,@TCSR
1377                                     BRESÉT
1378 002472                                     IF    #MAINT SETIN @TCSR THEN
1379 002472 000005          RESET
1380 002474                                     : MAINT DID NOT RESET IN TCSR
1381 002474 032777 000004 176562          BIT    #MAINT,@TCSR
1382 002502 001401          BEQ    $12
1383                                     ERRHRD 9,,DIDNOT
1384 002504                                     ENDF
1385 002504 104011          ERROR 9
1386 002506
1387 002506          $12:
1388 002506          ENDSUB
1389 002506          ENDTST
1390
1391
1392
```

```

1393 ;*****
1394 ;*****
1395 ;*TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
1396 ;*****
1397 002506 000004 ST4: SCOPE
1398 002510 012767 000010 176442 MOV #10,$TIMES ; DO 10 ITERATIONS
1399 002516 012767 000004 176454 MOV #4,$TESTN ; SET TEST NUMBER IN APT MAIL BOX
1400 ; ; USE PRIORITY OF 7
1401 002524 012746 000340 MOV #PR7,-(SP) ; PUT NEW PS ON STACK
1402 002530 012746 002536 MOV #64$,-(SP) ; PUT NEW PC ON STACK
1403 002534 000002 RTI ; POP NEW PC AND PS
1404 002536 64$:
1405
1406 ; SEE IF IT IS CLEAR
1407 002536 ; BGNSUB
1408 002536 012767 002544 176344 MOV #65$,$LPERR
1409
1410 ; IF #XMITIE SET IN @TCSR THEN
1411 002544 032777 000100 176512 BIT #XMITIE,@TCSR
1412 002552 001401 BEQ $13
1413 ; XMITIE DID NOT RESET IN TCSR
1414 002554 ; ERRHRD 10,,DIDNOT
1415 002554 104012 ERROR 10
1416 002556 ENDIF
1417 002556 $13:
1418 002556 ENDSUB
1419
1420 ; TRY TO SET XMITIE BIT
1421 002556 ; BGNSUB
1422 002556 012767 002564 176324 MOV #64$,$LPERR
1423 002564 LET @TCSR := @TCSR SET.BY #XMITIE
1424 002564 052777 000100 176472 BIS #XMITIE,@TCSR
1425 ; STUCK TO 0
1426 002572 IF #XMITIE NOTSET IN @TCSR THEN
1427 002572 032777 000100 176464 BIT #XMITIE,@TCSR
1428 002600 001001 BNE $14
1429 ; XMIT DID NOT RESET IN TCSR
1430 002602 ; ERRHRD 11,,DIDNOT
1431 002602 104013 ERROR 11
1432 002604 ENDIF
1433 002604 $14:
1434 002604 ENDSUB
1435
1436 ; TRY TO CLEAR A SET BIT
1437 002604 ; BGNSUB
1438 002604 012767 002612 176276 MOV #64$,$LPERR
1439
1440 LET @TCSR := @TCSR CLR.BY #XMITIE
1441 002612 042777 000100 176444 BIC #XMITIE,@TCSR
1442 ; SHOULD HAVE CLEARED
1443 002620 IF #XMITIE SET IN @TCSR THEN
1444 002620 032777 000100 176436 BIT #XMITIE,@TCSR
1445 002626 001401 BEQ $15
1446 ; XMIT DID NOT CLEAR IN TCSR
1447 002630 ; ERRHRD 12,,DIDNOT
1448 002630 104014 ERROR 12

```



J03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 37  
DVDVA.A.P11 06-MAY-77 15:29 T4 XMITIE - TCSR6 SET, CLEAR, RESET

SEQ 0035

```

1449 002632
1450 002632
1451 002632
1452
1453
1454 002632
1455 002632 012767 002640 176250      MOV      #64$, $LPERR
1456
1457 002640
1458 002640 052777 000100 176416      BIS      #XMITIE, @TCSR
1459
1460 002646
1461 002646 000005      RESET
1462 002650
1463 002650 032777 000100 176406      BIT      #XMITIE, @TCSR
1464 002656 001401      BEQ      $16
1465
1466 002660
1467 002660 104015      ERROR   13
1468 002662
1469 002662
1470 002662
1471 002662
1472
1473
1474

```

```

ENDIF
$15:
ENDSUB
; NOW SEE IF RESET CLEARS IT
BGNSUB
LET @TCSR := @TCSR SET.BY #XMITIE
; ISSUE BUS RESET
BRESÉ
IF #XMITIE SETIN @TCSR THEN
; XMIT DID NOT RESET IN TCSR
ERRHRD 13,,DIDNOT
ENDIF
$16:
ENDSUB
ENDTST

```

# K03

MAINDEC-11-DVDVA-A      MACY11 27(1006)      16-MAY-77 10:36 PAGE 38  
 DVDVAA.P11      06-MAY-77 15:29

T4      XMITIE - TCSR6 SET, CLEAR, RESET

SEQ 0036

```

1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485 002662 000004
1486 002664 012767 000010 176266
1487 002672 012767 000005 176300
1488
1489 002700
1490 002700 012767 002706 176202
1491 002706
1492 002706 042777 000002 176344
1493
1494 002714
1495 002714 032777 000002 176336
1496 002722 001401
1497
1498 002724
1499 002724 104016
1500 002726
1501 002726
1502 002726
1503
1504
1505 002726
1506 002726 012767 002734 176154
1507
1508 002734
1509 002734 052777 000002 176316
1510 002742
1511 002742 032777 000002 176310
1512 002750 001001
1513
1514 002752
1515 002752 104017
1516 002754
1517 002754
1518 002754
1519
1520
1521 002754
1522 002754 012767 002762 176126
1523 002762
1524 002762 042777 000002 176270
1525
1526 002770
1527 002770 032777 000002 176262
1528 002776 001401
1529
1530 003000
  
```

```

*****
*****
*TEST 5            DTR - RCSR1    SET, CLEAR
*                    NOTE: RESET DOES NOT CLEAR THIS BIT
*                    THIS BIT IS THE ONLY ONE IN THIS POSITION
*                    THAT IS READ AND WRITE.
*                    WE CANNOT TEST FOR AN INITIAL CONDITION
*                    AS THIS BIT IS UNDEFINED UPON POWER UP AND
*                    INIT DOESN'T AFFECT IT.
*****
↑STS:    SCOPE
         MOV        #10,$TIMES        ;;DO 10 ITERATIONS
         MOV        #5,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
                            ; TRY TO CLEAR DTR BIT
                                       BGNSUB
                                       LET     @RCSR := @RCSR CLR.BY #DTR
                                       ; STUCK TO 0
                                       IF     #DTR SETIN @RCSR THEN
                                                  ; DTR DID NOT CLEAR IN RCSR
                                                  ERRHRD 14,,DIDNOT
                                                  ENDIF
$17:                               ENDSUB
                                       ; TRY TO SET DTR
                                                  BGNSUB
                                       LET     @RCSR := @RCSR SET.BY #DTR
                                       IF     #DTR NOTSETIN @RCSR THEN
                                                  ; DTR DID NOT SET IN RCSR
                                                  ERRHRD 15,,DIDNOT
                                                  ENDIF
$20:                               ENDSUB
                                       ; TRY TO CLEAR IT AGAIN
                                                  BGNSUB
                                       MOV     #64$, $LPERR
                                       LET     @RCSR := @RCSR CLR.BY #DTR
                                       BIC     #DTR, @RCSR
                                       ; SHOULD HAVE CLEARED IT
                                       IF     #DTR SETIN @RCSR THEN
                                                  ; DTR DID NOT CLEAR IN RCSR
                                                  ERRHRD 16,,DIDNOT
  
```

L03

MAINDEC-11-DVDVA-A      MACY11 27(1006)    16-MAY-77 10:36    PAGE 39  
DVDVAA.P11      06-MAY-77 15:29    T5      DTR - RCSR1    SET, CLEAR

SEQ 0037

1531 003000 104020  
1532 003002  
1533 003002  
1534 003002  
1535 003002  
1536  
1537  
1538

ERROR 16

\$21:

ENDIF

ENDSUB  
ENDTST

M03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 40  
DVDVAR.P11 06-MAY-77 15:29 TS

DTR - RCSR1 SET, CLEAR

SEQ 0038

```

1539
1540
1541
1542
1543
1544 003002 000004
1545 003004 012767 000010 176146
1546 003012 012767 000006 176160
1547 003020
1548 003020 032767 040000 176172
1549 003026 001004
1550 003030
1551 003030 012767 000001 176122
1552 003036 000452
1553 003040
1554 003040
1555
1556
1557 003040
1558 003040 012767 003046 176042
1559
1560 003046
1561 003046 032777 000004 176294
1562 003054 001401
1563
1564 003056
1565 003056 104021
1566 003060
1567 003060
1568 003060
1569
1570
1571 003060
1572 003060 012767 003066 176022
1573 003066
1574 003066 052777 000004 176164
1575
1576 003074
1577 003074 032777 000004 176156
1578 003102 001001
1579
1580 003104
1581 003104 104022
1582 003106
1583 003106
1584 003106
1585
1586
1587 003106
1588 003106 012767 003114 175774
1589
1590 003114
1591 003114 042777 000004 176136
1592
1593 003122
1594 003122 032777 000004 176130

```

```

*****
*****
*TEST 6      REQSEND - RCSR2      SET, CLEAR, RESET
*          THIS TEST ASSUMES THAT *MPER FR IS IN
*****
TST6:  SCOPE
      MOV      #10,$TIMES      ;;DO 10 ITERATIONS
      MOV      #6,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                   IF #FRFD NOTSETIN $USWR THEN
      BIT      #FRFD,$USWR
      BNE      $22
                                   EXIT TST
      MOV      #1,$TIMES
      BR       TST7            ;;EXIT THIS TEST
                                   ENDF
$22:
                                   ; SEE IF IT IS CLEAR
                                   BGNSUB
      MOV      #64,$$LPERR
                                   IF      #REQSEND SETIN @RCSR THEN
      BIT      #REQSEND,@RCSR
      BEQ      $23
                                   ; REQSEND DID NOT RESET IN RCSR
                                   ERRHRD 17,,DIDNOT
      ERROR    17
                                   ENDF
$23:
      ENDSUB
                                   ; TRY TO SET REQSEND BIT
                                   BGNSUB
      MOV      #64,$$LPERR
      LET      @RCSR := @RCSR SET.BY #REQSEND
      BIS      #REQSEND,@RCSR
                                   IF ; STUCK TO 0
                                   #REQSEND NOTSETIN @RCSR THEN
      BIT      #REQSEND,@RCSR
      BNE      $24
                                   ; REQSEND DID NOT SET IN RCSR
                                   ERRHRD 18,,DIDNOT
      ERROR    18
                                   ENDF
$24:
      ENDSUB
                                   ; TRY TO CLEAR A SET BIT
                                   BGNSUB
      MOV      #64,$$LPERR
      LET      @RCSR := @RCSR CLR.BY #REQSEND
      BIC      #REQSEND,@RCSR
                                   IF ; SHOULD HAVE CLEARED
                                   #REQSEND SETIN @RCSR THEN
      BIT      #REQSEND,@RCSR

```



1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680

003164 000004  
003166 012767 000010 175764  
003174 012767 000007 175776  
  
003202  
003202 012767 003210 175700  
  
003210  
003210 032777 000010 176042  
003216 001401  
  
003220  
003220 104025  
003222  
003222  
003222  
  
003222  
003222 012767 003230 175660  
003230  
003230 052777 000010 176022  
  
003236  
003236 032777 000010 176014  
003244 001001  
  
003246  
003246 104026  
003250  
003250  
003250  
  
003250  
003250 012767 003256 175632  
  
003256  
003256 042777 000010 175774  
  
003264  
003264 032777 000010 175766  
003272 001401  
  
003274  
003274 104027  
003276  
003276  
003276

```
*****
*****
;TEST 7          SECXMIT - RCSR3          SET, CLEAR, RESET
;              THIS BIT IS THE ONLY ONE IN THIS POSITION
;              THAT IS READ AND WRITE.
*****
↑ST7:  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     $27
      ; SECXMIT DID NOT RESET IN RCSR
      ERRHRD 21,,DIDNOT
      ERROR   21
      ENDIF
$27:  ENDSUB
      ; TRY TO SET SECXMIT BIT
      BGNSUB
      MOV      #64,$LPERR
      LET     @RCSR := @RCSR SET.BY #SECXMIT
      BIS     #SECXMIT,@RCSR
      IF      ; STUCK TO 0
      #SECXMIT NOTSETIN @RCSR THEN
      ; SECXMIT DID NOT SET IN RCSR
      ERRHRD 22,,DIDNOT
      ERROR   22
      ENDIF
$30:  ENDSUB
      ; TRY TO CLEAR A SET BIT
      BGNSUB
      MOV      #64,$LPERR
      LET     @RCSR := @RCSR CLR.BY #SECXMIT
      BIC     #SECXMIT,@RCSR
      IF      ; SHOULD HAVE CLEARED
      #SECXMIT SETIN @RCSR THEN
      ; SECXMIT DID NOT CLEAR IN RCSR
      ERRHRD 23,,DIDNOT
      ERROR   23
      ENDIF
$31:  ENDSUB
```



C04

```

1681 003276                                BGNSUB
1682 003276 012767 003304 175604          MOV    #64$, $LPERR
1683                                     ; NOW SEE IF RESET CLEARS IT
1684
1685 003304                                LET    @RCSR := @RCSR SET.BY #SECXMIT
1686 003304 052777 000010 175746          BIS    #SECXMIT, @RCSR
1687                                     ; ISSUE BUS RESET
1688 003312                                BRESÉT
1689 003312 000005                          RESET
1690 003314                                IF    #SECXMIT SETIN @RCSR THEN
1691 003314 032777 000010 175736          BIT    #SECXMIT, @RCSR
1692 003322 001401                          BEQ    $32
1693                                     ; SECXMIT DID NOT RESET IN RCSR
1694 003324                                ERRHRD 24, , DIDNOT
1695 003324 104030                          ERROR  24
1696 003326
1697                                     $32:
1698 003326
1699 003326                                ENDIF
1700
1701                                     ENDSUB
1702                                     ENDTST

```

```

1703 .....*****
1704 .....*****
1705 *TEST 10      DATAIE - RCSR3 SET, CLEAR, RESET
1706 *           THIS BIT IS THE ONLY ONE IN THIS POSITION
1707 *           THAT IS READ AND WRITE.
1708 .....*****
1709 003326 000004  ST10: SCOPE
1710 003330 012767 000010 175622  MOV #10,$TIMES ;;DO 10 ITERATIONS
1711 003336 012767 000010 175634  MOV #10,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1712 ; SEE IF IT IS CLEAR
1713 003344  BGNSUB
1714 003344 012767 003352 175536  MOV #64,$SLPERR
1715
1716 003352  IF #DATAIE SETIN @RCSR THEN
1717 003352 032777 000040 175700  BIT #DATAIE,@RCSR
1718 003360 001401  BEQ $33
1719 ; DATAIE DID NOT RESET IN RCSR
1720 003362  ERRHRD 25,,DIDNOT
1721 003362 104031  ERROR 25
1722 003364  ENDF
1723 003364  $33:
1724 003364  ENDSUB
1725
1726 ; TRY TO SET DATAIE BIT
1727 003364  BGNSUB
1728 003364 012767 003372 175516  MOV #64,$SLPERR
1729 003372  LET @RCSR := @RCSR SET.BY #DATAIE
1730 003372 052777 000040 175660  BIS #DATAIE,@RCSR
1731 ; STUCK TO 0
1732 003400  IF #DATAIE NOTSETIN @RCSR THEN
1733 003400 032777 000040 175652  BIT #DATAIE,@RCSR
1734 003406 001001  BNE $34
1735 ; DATAIE DID NOT SET IN RCSR
1736 003410  ERRHRD 26,,DIDNOT
1737 003410 104032  ERROR 26
1738 003412  ENDF
1739 003412  $34:
1740 003412  ENDSUB
1741
1742 ; TRY TO CLEAR A SET BIT
1743 003412  BGNSUB
1744 003412 012767 003420 175470  MOV #64,$SLPERR
1745
1746 003420  LET @RCSR := @RCSR CLR.BY #DATAIE
1747 003420 042777 000040 175632  BIC #DATAIE,@RCSR
1748 ; SHOULD HAVE CLEARED
1749 003426  IF #DATAIE SETIN @RCSR THEN
1750 003426 032777 000040 175624  BIT #DATAIE,@RCSR
1751 003434 001401  BEQ $35
1752 ; DATAIE DID NOT CLEAR IN RCSR
1753 003436  ERRHRD 27,,DIDNOT
1754 003436 104033  ERROR 27
1755 003440  ENDF
1756 003440  $35:
1757 003440  ENDSUB
1758

```

E04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 45  
DVDVA.P11 06-MAY-77 15:29

T10 DATAIE - RCSR5 SET, CLEAR, RESET

SEQ 0043

```

1759 ; NOW SEE IF RESET CLEARS IT
1760 003440 ; BGNSUB
1761 003440 012767 003446 175442 MOV #645,$LPERR
1762
1763 003446 LET @RCSR := @RCSR SET.BY #DATAIE
1764 003446 052777 000040 175604 BIS #DATAIE,@RCSR
1765 ; ISSUE BUS RESET
1766 003454 BRESÉT
1767 003454 000005 RESET
1768 003456 IF #DATAIE SETIN @RCSR THEN
1769 003456 032777 000040 175574 BIT #DATAIE,@RCSR
1770 003464 001401 BEQ $36
1771 ; DATAIE DID NOT RESET IN RCSR
1772 003466 ERRHRD 28,,DIDNOT
1773 003466 104034 ERROR 28
1774 003470
1775 003470 $36:
1776 003470
1777 003470
1778
1779
1780

```

ENDSUB  
ENDTST

F04

MAINDEC-11-DVDVA-A  
DVDVAA.P11 06-MAY-77

MACY11 27(1006)  
15:29

16-MAY-77 10:36 PAGE 46  
T10 DATAIE - RCSR5 SET, CLEAR, RESET

SEQ 0044

```

1781 ;*****
1782 ;*****
1783 *TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET
1784 * THIS BIT IS THE ONLY ONE IN THIS POSITION
1785 * THAT IS READ AND WRITE.
1786 ;*****
1787 ST11: SCOPE
1788 MOV #10,STIMES ;:DO 10 ITERATIONS
1789 MOV #11,STESTN ;:SET TEST NUMBER IN APT MAIL BOX
1790 ; SEE IF IT IS CLEAR
1791 ; BGNSUB
1792 MOV #645,$LPERR
1793
1794 IF #RCVRIE SETIN @RCSR THEN
1795 BIT #RCVRIE,@RCSR
1796 BEQ $37
1797 ; RCVRIE DID NOT RESET IN RCSR
1798 ; ERRHRD 29,,DIDNOT
1799 ERROR 29
1800 ; ENDF
1801 $37:
1802 ; ENDSUB
1803
1804 ; TRY TO SET RCVRIE BIT
1805 ; BGNSUB
1806 MOV #645,$LPERR
1807 LET @RCSR := @RCSR SET.BY #RCVRIE
1808 BIS #RCVRIE,@RCSR
1809 ; STUCK TO 0
1810 IF #RCVRIE NOTSETIN @RCSR THEN
1811 BIT #RCVRIE,@RCSR
1812 BNE $40
1813 ; RCVRIE DID NOT SET IN RCSR
1814 ; ERRHRD 30,,DIDNOT
1815 ERROR 30
1816 ; ENDF
1817 $40:
1818 ; ENDSUB
1819
1820 ; TRY TO CLEAR A SET BIT
1821 ; BGNSUB
1822 MOV #645,$LPERR
1823 LET @RCSR := @RCSR CLR.BY #RCVRIE
1824 BIC #RCVRIE,@RCSR
1825 ; SHOULD HAVE CLEARED
1826 IF #RCVRIE SETIN @RCSR THEN
1827 BIT #RCVRIE,@RCSR
1828 BEQ $41
1829 ; RCVRIE DID NOT CLEAR IN RCSR
1830 ; ERRHRD 31,,DIDNOT
1831 ERROR 31
1832 ; ENDF
1833 $41:
1834 ; ENDSUB
1835
1836

```

G04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 47  
DVDVAA.P11 06-MAY-77 15:29 T11 RCVRIE - RCSR6 SET, CLEAR, RESET

SEQ 0045

```
1837 ; NOW SEE IF RESET CLEARS IT
1838 003602 BGNSUB
1839 003602 012767 003610 175300 MOV #64$, $LPERR
1840
1841 003610 LET @RCSR := @RCSR SET.BY #RCVRIE
1842 003610 052777 000100 175442 BIS #RCVRIE, @RCSR
1843 ; ISSUE BUS RESET
1844 003616 BRESÉT
1845 003616 000005 RESET
1846 003620 IF #RCVRIE SETIN @RCSR THEN
1847 003620 032777 000100 175432 BIT #RCVRIE, @RCSR
1848 003626 001401 BEQ $42
1849 ; RCVRIE DID NOT RESET IN RCSR
1850 003630 ERRHRD 32,, DIDNOT
1851 003630 104040 ERROR 32
1852 003632
1853 003632 $42:
1854 003632
1855 003632 CKLOOP
1856 003632 ENDSUB
1857
1858
1859
1860 ENDTST
```

H04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 48  
DVDVAA.P11 06-MAY-77 15:29 T11

RCVRIE - RCSR6 SET, CLEAR, RESET

SEG 0046

1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897

003632 000004  
003634 012767 000010 175316  
003642 012767 000012 175330  
  
003650  
003650 012767 003656 175232  
003656  
003656 032777 000200 175374  
003664 001402  
  
003666  
003666 104041  
  
003670  
003670 000005  
003672  
003672  
003672  
003672

```
*****  
* THE FOLLOWING 4 TESTS VERIFY  
* THAT RESET (INIT) INITIALIZES READ ONLY BITS.  
*****  
*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 SET IN RCSR THEN  
BIT #RCVRDONE, RCSR  
BEQ $43  
  
;RCVRDONE SHOULD HAVE CLEARED BY INIT  
; RCVRDONE DID NOT CLEAR IN RCSR  
ERRHRD 33,HRESET, DIDNOT  
  
;REISSUE RESET  
BRESET  
  
ENDIF  
  
;ALLOW LOOPING AFTER ERROR  
CKLOOP  
ENDSUB  
ENDTST
```

\$43:

```

1898
1899
1900
1901
1902 003672 000004
1903 003674 012767 000010 175256
1904 003702 012767 000013 175270
1905
1906
1907
1908
1909 003710
1910 003710 012767 003716 175172
1911
1912 003716
1913 003716 032777 000200 175340
1914 003724 001002
1915
1916
1917
1918 003726
1919 003726 104042
1920
1921 003730
1922 003730 000005
1923 003732
1924 003732
1925
1926 003732
1927 003732
1928 003732
1929
1930
1931

```

```

*****
*****
*TEST 13 TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
*****
†ST13: 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 $44
;RESET SHOULD HAVE SET BIT.
;XMITRDY DID NOT SET IN TCSR (AFTER RESE
ERRHRD 34,HRESET,DIDNOT
;ISSUE ANOTHER RESET
BRESET
ENDIF
;ALLOW LOOPING ON ERROR
CKLOOP
ENDSUB
ENDTST
$44:

```



J04

MAINDEC-11-DVDVA-A MACY11 27(1006)  
DVDVA.A.P11 06-MAY-77 15:29

16-MAY-77 10:36 PAGE 50  
T13 TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT

SEQ 0048

1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965

003732 000004  
003734 012767 000010 175216  
003742 012767 000014 175230  
  
003750  
003750 012767 003756 175132  
003756  
003756 032777 100000 175274  
003764 001402  
  
003766  
003766 104043  
  
003770  
003770 000005  
003772  
003772  
003772  
003772  
003772

\*\*\*\*\*  
\*\*\*\*\*  
\*TEST 14 TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.  
\*\*\*\*\*

TST14: SCOPE  
MOV #10,\$TIMES ;;DO 10 ITERATIONS  
MOV #14,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX

BGNSUB  
MOV #64,\$SLPERR  
IF #DATAINT SETIN @RCSR THEN  
BIT #DATAINT,@RCSR  
BEQ \$45  
  
ERRHRD 35, HRESET, DIDNOT  
ERROR 35  
  
;TESTING EFFECT OF RESET ON BIT  
;DATAINT DID NOT CLEAR IN RCSR  
;ALLOW A FRESH START  
BRESET  
  
RESET  
ENDIF  
CKLOOP  
ENDSUB  
ENDTST

\$45:

K04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 51  
DVDVAA.P11 06-MAY-77 15:29 T14

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

SEQ 0049

1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011

003772 000004  
003774 012767 000010 175156  
004002 012767 000015 175170  
  
004010  
004010 032767 020000 175202  
004016 001004  
  
004020  
004020 012767 000001 175132  
004026 000411  
004030  
004030  
  
004030  
004030 012767 004036 175052  
  
004036  
004036 032777 004000 175214  
004044 001402  
  
004046  
004046 104044  
  
004050  
004050 000005  
004052  
004052  
004052  
004052  
004052

```
*****  
*****  
*TEST 15 TEST THAT RCVRACT - RCSR 11 - IS 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 $46 ; CAN'T TEST WITHOUT BERG OR H315.  
EXIT TST  
MOV #1,$TIMES  
BR TST16 ;;EXIT THIS TEST  
ENDIF  
$46:  
BGNSUB  
MOV #64,$SLPERR  
IF #RCVRACT SETIN @RCSR THEN  
BIT #RCVRACT,@RCSR  
BEQ $47  
:RESET SHOULD HAVE CLEARED RCVRACT  
ERRHRD 36, HRESET, DIDNOT  
ERROR 36  
;TESTING EFFECT OF RESET ON BIT  
;RCVRACT DID NOT CLEAR IN RCSR  
:ALLOW ANOTHER TRY  
BRESET  
ENDIF  
$47:  
;ALLOW LOOPING ON ERROR  
CKLOOP  
ENDSUB  
ENDTST
```

L04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 52  
DVDVAA.P11 06-MAY-77 15:29 T15

TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT

SEQ 0050

2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021 004052 000004  
2022 004054 012767 000010 175076  
2023 004062 012767 000016 175110  
2024  
2025 004070  
2026 004070 032767 020000 175122  
2027 004076 001004  
2028  
2029 004100  
2030 004100 012767 000001 175052  
2031 004106 000441  
2032 004110  
2033 004110  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041 004110  
2042 004110 012767 004116 174772  
2043  
2044  
2045 004116  
2046 004116 042777 000002 175134  
2047  
2048 004124  
2049 004124 032777 010000 175126  
2050 004132 001401  
2051  
2052 004134  
2053 004134 104045  
2054  
2055  
2056 004136  
2057 004136  
2058 004136  
2059  
2060  
2061 004136  
2062 004136 012767 004144 174744  
2063  
2064  
2065 004144  
2066 004144 052777 000002 175106  
2067

```
*****
* THE FOLLOWING 4 TESTS VERIFY
* THAT THE EIA SIGNALS CAN BE TRANSMITTED
* AND RECEIVED THROUGH THE CABLE
*****
*****
*TEST 16 TEST THAT CARDET SETS AND CLEARS
* AS DTR SETS AND CLEARS
*****
TST16: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #16,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
;;CAN WE USE THE WRAPAROUND??
IF #CABLE NOTSETIN $USWR THEN
; CAN'T TEST WITHOUT BERG OR H315.
EXIT TST
MOV #1,$TIMES
BR TST17 ;;EXIT THIS TEST
ENDIF
$S0:
; DTR AND
; CARDET ARE CONNECTED
; BY THE H315 OR EQUIV.
; CLEAR
BGNSUB
MOV #64,$SLPERR
BIC #DTR,$RCSR
; CLEAR DTR
LET $RCSR := $RCSR CLR.BY #DTR
; CARDET SHOULD FOLLOW
IF #CARDET SETIN $RCSR THEN
; CARDET DID NOT
ERRHRD 37,,FORCE
; CLEAR WITH DTR
ENDIF
$S1:
ENDSUB
; SET
BGNSUB
MOV #64,$SLPERR
BIS #DTR,$RCSR
LET $RCSR := $RCSR SET.BY #DTR
; CARDET SHOULD FOLLOW
```



NO4

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 54  
DVDVA.A.P11 06-MAY-77 15:29

T16 TEST THAT CARDET SETS AND CLEARS

SEQ 0052

```

2104 .....
2105 .....
2106 .....
2107 *TEST 17 TEST THAT CLREND SETS AND CLEARS
2108 * AS DTR SETS AND CLEARS
2109 .....
2110 004212 000004 TST17: SCOPE
2111 004214 012767 000010 174736 MOV #10,STIMES ;;DO 10 ITERATIONS
2112 004222 012767 000017 174750 MOV #17,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
2113 004230 IF ; CAN WE USE THE WRAPAROUND??
2114 004230 032767 020000 174762 BIT #CABLE,SUSWR ;CABLE NOTSETIN SUSWR THEN
2115 004236 001004 BNE $54
2116 ; CAN'T TEST WITHOUT BERG OR H315.
2117 004240 EXIT TST
2118 004240 012767 000001 174712 MOV #1,STIMES
2119 004246 000441 BR TST20 ;;EXIT THIS TEST
2120 004250 ENDIF
2121 004250 $54:
2122
2123 ; DTR AND
2124 ; CLREND ARE CONNECTED
2125
2126 ; BY THE H315 OR EQUIV.
2127
2128 ; CLEAR
2129 004250 BGNSUB
2130 004250 012767 004256 174632 MOV #64$,SLPERR
2131
2132
2133 004256 LET ; CLEAR DTR
2134 004256 042777 000002 174774 BIC #DTR,@RCSR @RCSR := @RCSR CLR.BY #DTR
2135
2136 IF ; CLREND SHOULD FOLLOW
2137 004264 032777 020000 174766 BIT #CLREND,@RCSR #CLREND SETIN @RCSR THEN
2138 004272 001401 BEQ $55
2139
2140 ; CLREND DID NOT
2141 004274 104050 ERROR 40 ERRHRD 40,.FORCE
2142
2143 ; CLEAR WITH DTR
2144 004276 ENDIF
2145 004276 $55:
2146 004276 ENDSUB
2147
2148 ; SET
2149 004276 BGNSUB
2150 004276 012767 004304 174604 MOV #64$,SLPERR
2151
2152
2153 LET ; SET DTR
2154 004304 052777 000002 174746 BIS #DTR,@RCSR @RCSR := @RCSR SET.BY #DTR
2155
2156 IF ; CLREND SHOULD FOLLOW
2157 004312 032777 020000 174740 BIT #CLREND,@RCSR #CLREND NOTSETIN @RCSR THEN
2158 004320 001001 BNE $56
2159 ; CLREND DID NOT SET

```



```

2192 ;*****
2193 ;*****
2194 ;TEST 20 TEST THAT RING SETS AND CLEARS
2195 ;* AS REQSEND SETS AND CLEARS
2196 ;*****
2197 004352 000004 †ST20: SCOPE
2198 004354 012767 000010 174576 MOV #10,$TIMES ;;DO 10 ITERATIONS
2199 004362 012767 000020 174610 MOV #20,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2200 ; CAN WE USE THE WRAPAROUND??
2201 004370 IF #CABLE,$USWR ;CABLE NOTSETIN $USWR THEN
2202 004370 032767 020000 174622 BIT #CABLE,$USWR
2203 004376 001004 BNE $60
2204 ; CAN'T TEST WITHOUT BERG OR H315.
2205 004400 EXIT TST
2206 004400 012767 000001 174552 MOV #1,$TIMES
2207 004406 000441 BR TST21 ;;EXIT THIS TEST
2208 004410
2209 004410 $60: ENDF
2210
2211 ; REQSEND AND
2212 ; RING ARE CONNECTED
2213
2214 ; BY THE H315 OR EQUIV.
2215
2216 ; CLEAR
2217 004410 BGNSUB
2218 004410 012767 004416 174472 MOV #64,$SLPERR
2219
2220 ; CLEAR REQSEND
2221 004416 LET @RCSR := @RCSR CLR.BY #REQSEND
2222 004416 042777 000004 174634 BIC #REQSEND,@RCSR
2223 ; RING SHOULD FOLLOW
2224 004424 IF #RING SETIN @RCSR THEN
2225 004424 032777 040000 174626 BIT #RING,@RCSR
2226 004432 001401 BEQ $61
2227 ; RING DID NOT
2228 004434 ERRHRD 43,,FORCE
2229 004434 104053 ERROR 43
2230
2231 ; CLEAR WITH REQSEND
2232 004436 ENDF
2233 004436 $61:
2234 004436 ENDSUB
2235
2236 ; SET
2237 004436 BGNSUB
2238 004436 012767 004444 174444 MOV #64,$SLPERR
2239
2240 ; SET REQSEND
2241 004444 LET @RCSR := @RCSR SET.BY #REQSEND
2242 004444 052777 000004 174606 BIS #REQSEND,@RCSR
2243 ; RING SHOULD FOLLOW
2244 004452 IF #RING NOTSETIN @RCSR THEN
2245 004452 032777 040000 174600 BIT #RING,@RCSR
2246 004460 001001 BNE $62
2247 ; RING DID NOT SET

```



```

2248 004462                                ERRHRD 44,,FORCE
2249 004462 104054                        ERROR 44
2250
2251
2252 004464                                ; CLEAR
2253 004464                                ; CLEAR
2254 004464                                ; CLEAR
2255
2256
2257 004464                                ; CLEAR
2258 004464 012767 004472 174416          MOV  #645,$LPERR
2259
2260
2261 004472                                ; CLEAR REQSEND
2262 004472 042777 000004 174560          BIC  #REQSEND,@RCSR
2263
2264 004500                                ; RING SHOULD FOLLOW
2265 004500 032777 040000 174552          BIT  #RING,@RCSR
2266 004506 001401                        BEQ  $63
2267
2268 004510                                ; RING DID NOT
2269 004510 104055                        ERROR 45
2270
2271
2272 004512                                ; CLEAR WITH REQSEND
2273 004512                                ; CLEAR WITH REQSEND
2274 004512                                ; CLEAR WITH REQSEND
2275 004512                                ; CLEAR WITH REQSEND
2276
2277
2278
2279

```

\$62:

ENDSUB  
ENDTST

\$63:

E05

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 58  
DVDVA.P11 06-MAY-77 15:29

T20 TEST THAT RING SETS AND CLEARS

SEQ 0056

```

2280 ;*****
2281 ;*****
2282 ;TEST 21 TEST THAT SECREC SETS AND CLEARS
2283 ;* AS SECXMIT SETS AND CLEARS
2284 ;*****
2285 004512 000004 TST21: SCOPE
2286 004514 012767 000010 174436 MOV #10,$TIMES ;;DO 10 ITERATIONS
2287 004522 012767 000021 174450 MOV #21,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2288 ; CAN WE USE THE WRAPAROUND??
2289 004530 IF #CABLE,$USWR ;CABLE NOTSETIN $USWR THEN
2290 004530 032767 020000 174462 BIT #CABLE,$USWR
2291 004536 001004 BNE $64
2292 ; CAN'T TEST WITHOUT BERG OR H315.
2293 004540 EXIT TST
2294 004540 012767 000001 174412 MOV #1,$TIMES
2295 004546 000441 BR TST22 ;;EXIT THIS TEST
2296 004550 ENDF
2297 004550 $64:
2298
2299 ; SECXMIT AND
2300 ; SECRC ARE CONNECTED
2301
2302 ; BY THE H315 OR EQUIV.
2303
2304 ; CLEAR
2305 004550 BGNSUB
2306 004550 012767 004556 174332 MOV #64,$LPERR
2307
2308 ; CLEAR SECXMIT
2309 004556 LET @RCSR := @RCSR CLR.BY #SECXMIT
2310 004556 042777 000010 174474 BIC #SECXMIT,@RCSR
2311 ; SECRC SHOULD FOLLOW
2312 004564 IF #SECRC SETIN @RCSR THEN
2313 004564 032777 002000 174466 BIT #SECRC,@RCSR
2314 004572 001401 BEQ $65
2315 ; SECRC DID NOT
2316 004574 ERRHRD 46,,FORCE
2317 004574 104056 ERROR 46
2318
2319 ; CLEAR WITH SECXMIT
2320 004576 ENDF
2321 004576 $65:
2322 004576 ENDSUB
2323
2324 ; SET
2325 004576 BGNSUB
2326 004576 012767 004604 174304 MOV #64,$LPERR
2327
2328 ; SET SECXMIT
2329 004604 LET @RCSR := @RCSR SET.BY #SECXMIT
2330 004604 052777 000010 174446 BIS #SECXMIT,@RCSR
2331 ; SECRC SHOULD FOLLOW
2332 004612 IF #SECRC NOTSETIN @RCSR THEN
2333 004612 032777 002000 174440 BIT #SECRC,@RCSR
2334 004620 001001 BNE $66
2335 ; SECRC DID NOT SET

```

```

2336 004622                                ERRHRD 47,,FORCE
2337 004622 104057                        ERROR 47
2338
2339
2340 004624                                ; WITH SECXMIT
2341 004624                                ENDIF
2342 004624                                $66:
2343
2344
2345 004624                                ; CLEAR
2346 004624 012767 004632 174256          MOV #64$, $LPERR
2347
2348
2349 004632                                ; CLEAR SECXMIT
2350 004632 042777 000010 174420          BIC #SECXMIT, @RCSR
2351
2352 004640                                ; SECRC SHOULD FOLLOW
2353 004640 032777 002000 174412          BIT #SECRC, @RCSR
2354 004646 001401                          BEQ $67
2355
2356 004650                                ; SECRC DID NOT
2357 004650 104060                        ERROR 48
2358
2359
2360 004652                                ; CLEAR WITH SECXMIT
2361 004652                                ENDIF
2362 004652                                $67:
2363 004652                                ENDSUB
2364
2365
2366
2367
    
```

ENDTST

```

2368
2369
2370
2371
2372
2373
2374
2375 004652 000004
2376 004654 012767 000010 174276
2377 004662 012767 000022 174310
2378
2379 004670
2380 004670 032767 020000 174322
2381 004676 001004
2382
2383 004700
2384 004700 012767 000001 174252
2385 004706 000463
2386 004710
2387 004710
2388
2389
2390 004710 012746 000340
2391 004714 012746 004722
2392 004720 000002
2393 004722
2394
2395
2396 004722
2397 004722 012767 004730 174160
2398
2399 004730
2400 004730 042777 000002 174322
2401
2402 004736
2403 004736 010546
2404 004740 012745 000010
2405 004744 004767 004544
2406 004750 012605
2407
2408 004752
2409 004752 017703 174302
2410
2411 004756
2412 004756 032777 100000 174274
2413 004764 001401
2414
2415 004766
2416 004766 104061
2417 004770
2418 004770
2419
2420 004770
2421
2422
2423 00477C

```

```

: *****
: *****
: *TEST 22 TEST THAT 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 H315
: *****
TST22: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #22,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
;; CAN WE USE THE WRAPAROUND??
IF #CABLE NOTSETIN $USWR THEN
BIT #CABLE,$USWR
BNE $70
EXIT 'TST ; CAN'T TEST WITHOUT BERG OR H315.
MOV #1,$TIMES
BR TST23 ;;EXIT THIS TEST
ENDIF
$70:
MOV #PR7,-($P) ;MAKE SURE NOTHING UNEXPECTED HAPPENS
MOV #645,-($P) ;;PUT NEW PS ON STACK
RTI ;;PUT NEW PC ON STACK
;;POP NEW PC AND PS
;READ TWICE - CLEARS
BGNSUB
MOV #655,$LPERR
LET ; CLEAR DTR
RCSR := RCSR CLR.BY #DTR
;WAIT 10 MICRO-SEC FOR CABLE
WAITUS 10
MOV R5,$($P)
MOV #10,-($R5)
JSR PC,WAIT
MOV ($P)+,R5
LET ; READ RCSR - TO CLEAR DATAINT
R3 := RCSR
; READ RCSR AGAIN
IF #DATAINT SETIN RCSR THEN
; READING RCSR DID NOT CLEAR DATAINT
ERRHRD 49,EDATAINT
ENDIF
$71:
ENDSUB
; DTR SETTING SETS DATAINT
BGNSUB

```



```

2468 ;*****
2469 ;*****
2470 ;TEST 23 TEST THAT DATAINT SETS WHEN RING SETS
2471 ;* AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
2472 ;*****
2473 ST23: SCOPE
2474 005056 000004 MOV #10,STIMES ;;DO 10 ITERATIONS
2475 005050 012767 000010 174072 MOV #23,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
2476 005056 012767 000023 174104 IF ;CAN WE USE THE WRAPAROUND??
2477 005074 BIT #CABLE,SUSWR ;CABLE NOTSETIN SUSWR THEN
2478 005074 032767 020000 174116 BNE $75
2479 005172 001004
2480 ;CAN'T TEST WITHOUT BERG OR H315.
2481 005104 EXIT TST
2482 005104 012767 000001 174046 MOV #1,STIMES
2483 005112 000473 BR TST24 ;;EXIT THIS TEST
2484 005114 ENDF
2485 005114 $75:
2486
2487 ;NO INTERRUPTS
2488 005114 012746 000340 MOV #PR7,-(SP) ;;PUT NEW PS ON STACK
2489 005120 012746 005126 MOV #64$,-(SP) ;;PUT NEW PC ON STACK
2490 005124 000002 RTI ;;POP NEW PC AND PS
2491 005126 $64$:
2492
2493 ;START OFF WITH EVERYTHING CLEAR
2494 005126 BGN SUB
2495 005126 012767 005134 173754 MOV #65$,SLPERR
2496
2497 ;CLEAR RING
2498 005134 LET @RCSR := @RCSR CLR.BY #REQSEND
2499 005134 042777 000004 174116 BIC #REQSEND,@RCSR
2500 ;WAIT 10 MICRO-SEC FOR CABLE
2501 005142 WAITUS 10
2502 005142 010546 MOV R5,-(SP)
2503 005144 012745 000010 MOV #10,-(R5)
2504 005150 004767 004340 JSR PC,WAIT
2505 005154 012605 MOV (SP)+,R5
2506
2507 ;READ ONCE
2508 005156 LET R3 := @RCSR
2509 005156 017703 174076 MOV @RCSR,R3
2510 ;READ TWICE
2511 IF #DATAINT SETIN @RCSR THEN
2512 005162 032777 100000 174070 BIT #DATAINT,@RCSR
2513 005170 001401 BEQ $76
2514 ;READING RCSR DID NOT CLEAR DATAINT
2515 005172 104065 ERROR 53 ERRHRD 53, EDATAINT
2516 005174 ENDF
2517 005174 $76:
2518 005174 ENDSUB
2519
2520 ; SET RING --> SET DATAINT
2521 005174 BGN SUB
2522 005174 012767 005202 173706 MOV #64$,SLPERR
2523
    
```



K05

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 64  
DVDVA.P11 06-MAY-77 15:29

T23 TEST THAT DATAINT SETS WHEN RING SETS

SEQ 0062

2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627

005302 000004  
005304 012767 000010 173646  
005312 012767 000024 173660  
  
005320 032767 020000 173672  
005320 001004  
  
005330 012767 000001 173622  
005336 000454  
005340  
005340  
  
005340 012746 000340  
005344 012746 005352  
005350 000002  
005352  
  
005352 042777 000010 173700  
005360 017703 173674  
  
005364 012767 005372 173516  
  
005372 052777 000010 173660  
  
005400 010546  
005402 012745 000010  
005406 004767 004102  
005412 012605  
005414 032777 100000 173636  
005422 001001  
  
005424 104124  
005426  
005426  
005426

```
*****  
*****  
*TEST 24 TEST THAT DATAINT SETS WHEN SECURE CHANGES STATE  
*****  
↑ST24: SCOPE  
MOV #10, $TIMES ;; DO 10 ITERATIONS  
MOV #24, $TESTN ;; SET TEST NUMBER IN APT MAIL BOX  
;; CAN WE USE THE WRAPAROUND??  
IF #CABLE NOTSETIN $USWR THEN  
#CABLE, $USWR  
BNE $101  
;; CAN'T TEST WITHOUT BERG OR H315.  
;; IT TST  
MOV #1, $TIMES  
BR TST25 ;; EXIT THIS TEST  
ENDIF  
$101:  
MOV #PR7, -(SP) ;; NO INTERRUPTS  
MOV #64$, -(SP) ;; PUT NEW PS ON STACK  
RTI ;; PUT NEW PC ON STACK  
;; POP NEW PC AND PS  
  
; START FRESH  
; CLEAR SECURE  
LET @RCSR := @RCSR CLR. BY #SECXMIT  
LET R3 := @RCSR  
; SET SECURE --> DATAINT SET  
BGNSUB  
MOV #65$, $LPERR  
; SET SECURE  
LET @RCSR := @RCSR SET. BY #SECXMIT  
; WAIT 10 MICRO-SEC FOR CABLE  
WAITUS 10  
MOV R5, -(SP)  
MOV #10, -(R5)  
JSR PC, WAIT  
MOV (SP)+, R5  
IF #DATAINT NOTSETIN @RCSR THEN  
BIT #DATAINT, @RCSR  
BNE $102  
; SETTING SECURE DID NOT SET DATAINT  
ERRHRD 84,, E2DATA  
ERROR 84  
ENDIF  
$102:  
ENDSUB  
; CLEAR SECURE --> DATAINT SET
```





M05

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 66  
 DVDVA.A.P11 06-MAY-77 15:29

T24 TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE

SEG 0064

```

2652 ;*****
2653 ;*****
2654 *TEST 25 TEST THAT XMIT RDY - TCSR 7 - CLEARS
2655 * WHEN TBUF IS LOADED WITH A CHARACTER
2656 * AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
2657 ;*****
2658 †ST25: SCOPE
2659 MOV #10,$TIMES ;;DO 10 ITERATIONS
2660 MOV #25,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2661
2662 BGNSUB
2663 MOV #64,$SLPERR
2664 ; LOAD TBUF WITH ONE CHARACTER
2665 ; WAIT FOR READY TO SET
2666 ; (SHOULD BE VERY SHORT WAIT
2667 ; SINCE UART DOUBLE BUFFERS ITS INPUT)
2668
2669 ;SEND A CHARACTER
2670 LET @TBUF :B= #0
2671
2672 ;WAIT A MAXIMUM
2673 ;OF 50 MSEC FOR
2674 ;XMIT RDY TO SET IN TCSR
2675 CALL TIMER IN (&S,&XMITRDY,TCSR,&SET)
2676 MOV R5,-(SP)
2677 MOV #SET,-(R5)
2678 MOV TCSR,-(R5)
2679 MOV #XMITRDY,-(R5)
2680 MOV #5,-(R5)
2681 JSR PC,TIMER
2682 MOV (SP)+,R5
2683
2684 ;TIMER RETURNS AN ERROR IF BIT DID
2685 ;NOT MEET CONDITION WITHIN TIME LIMIT
2686 IF.ERROR THEN
2687
2688 ;XMIT RDY DID NOT SET IN TCSR
2689 ERRHRD 54,,DIDNOT
2690
2691 ENDIF
2692
2693 $104:
2694 ENDSUB
2695
2696 BGNSUB
2697 MOV #64,$SLPERR
2698 ; LOAD TBUF WITH A SECOND CHARACTER
2699 ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
2700 ; AND THEN WAIT FOR IT TO SET
2701
2702 ;SEND SECOND CHARACTER
2703 LET @TBUF :B= #0
2704
2705 ; GIVE IT TIME TO CLEAR
2706 ; XMITRDY SHOULD HAVE CLEARED UPON
2707 ; RECEIPT OF A CHARACTER
2708 IF #XMITRDY SET IN @TCSR THEN
  
```

N05

MAINDEC-11-DVDVA-A  
DVDVAA.P11

MACY11 27(1006)  
06-MAY-77 15:29

..-MAY-77 10:36 PAGE 67  
T25 TEST THAT XMIT RDY - TCSR 7 - CLEARS

SEQ 0065

```

2708 005576 001401          BEQ      $105
2709
2710 005600
2711 005600 104067          ERROR   55
2712 005602
2713 005602          $105:
2714
2715
2716
2717
2718 005602
2719 005602 010546          MOV     RS, -(SP)
2720 005604 012745 177777          MOV     #SET, -(RS)
2721 005610 016745 173450          MOV     TCSR, -(RS)
2722 005614 012745 000200          MOV     #XMITRDY, -(RS)
2723 005620 012745 000005          MOV     #5, -(RS)
2724 005624 004767 003406          JSR    PC, TIMER
2725 005630 012605          MOV     (SP)+, RS
2726 005632
2727 005632 103001          BCC     $106
2728
2729 005634
2730 005634 104070          ERROR   56
2731 005636
2732 005636          $106:
2733 005636
2734 005636

```

```

: XMITRDY DID NOT CLEAR IN TCSR
ERRHRD 55,,DIDNOT
ENDIF

: WAIT A MAXIMUM
: OF 50 *SEC FOR
: XMIT RDY TO SET IN TCSR
CALL TIMER IN (<#5,#XMITRDY,TCSR,#SET>)

IF.ERROR THEN
: XMIT RDY DID NOT SET IN TCSR
ERRHRD 56,,DIDNOT
ENDIF

ENDSUB
ENDTST

```

```

2735
2736
2737
2738
2739
2740
2741 005636 000004
2742 005640 012767 000010 173312
2743 005646 012767 000026 173324
2744
2745
2746 005654
2747 005654 052777 000004 173402
2748
2749 005662
2750 005662 012767 005670 173220
2751
2752
2753 005670
2754 005670 105077 173374
2755
2756
2757
2758
2759 005674
2760 005674 010546
2761 005676 012745 177777
2762 005702 016745 173352
2763 005706 012745 000200
2764 005712 012745 000005
2765 005716 004767 003314
2766 005722 012605
2767
2768
2769 005724
2770 005724 103001
2771
2772 005726
2773 005726 104071
2774 005730
2775 005730
2776
2777 005730
2778
2779 005730
2780 005730 012767 005736 173152
2781
2782
2783 005736
2784 005736 000005
2785
2786 005740
2787 005740 032777 000200 173312
2788 005746 001401
2789
2790 005750

```

```

*****
*****
*TEST 26      TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
*              AND THAT RESET CLEARS THE BIT.
*****
TST26: SCOPE
        MOV      #10,$TIMES      ;;DO 10 ITERATIONS
        MOV      #26,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

                                ; SET THE MAINTENANCE BIT
                                LET @TCSR := @TCSR SET.BY #MAINT
                                BGNSUB
        MOV      #64,$SLPERR
                                ; SEND A CHARACTER AND LET IT WRAP AROUND
                                LET @TBUF :B= #0
                                ; WAIT A MAXIMUM OF 50 MSEC
                                ; FOR RCVR DONE TO SET IN
                                ; RCSR
                                CALL TIMER IN (<#5,#RCVRDONE,RCSR,#SET>)
        MOV      RS,-(SP)
        MOV      #SET,-(RS)
        MOV      RCSR,-(RS)
        MOV      #RCVRDONE,-(RS)
        MOV      #5,-(RS)
        JSR      PC,TIMER
        MOV      (SP)+,RS

                                ;DIDN'T SET IN TIME
                                IF.ERROR THEN
        BCC      $107
                                ; RCVRDONE DID NOT SET IN RCSR
                                ERRHRD 57,,DIDNOT
        ENDIF
$107:
                                ENDSUB
                                BGNSUB
        MOV      #64,$SLPERR
                                ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
                                ; THIS ALSO WILL CLEAR THE MAINT. BIT
        RESET      BRESET
                                IF #RCVRDONE SETIN @RCSR THEN
        BIT      #RCVRDONE,@RCSR
        BEQ      $110
                                ; RCVRDONE DID NOT RESET IN RCSR.
                                ERRHRD 58,,DIDNOT

```

C06

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 69  
DVDVA.P11 06-MAY-77 15:29 T26 TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)

SEQ 0067

2791 005750 104072  
2792 005752  
2793 005752  
2794 005752  
2795 005752

ERROR 58

\$110:

ENDIF

ENDSUB

ENDTST

```

2796
2797
2798
2799
2800 005752 000004
2801 005754 012767 000010 173176
2802 005762 012767 000027 173210
2803
2804
2805 005770
2806 005770 052777 000004 173266
2807 005776
2808 005776 012767 006004 173104
2809
2810
2811
2812
2813 006004
2814 006004 105077 173260
2815
2816
2817
2818 006010
2819 006010 010546
2820 006012 012745 177777
2821 006016 016745 173236
2822 006022 012745 000200
2823 006026 012745 000050
2824 006032 004767 003200
2825 006036 012605
2826
2827 006040
2828 006040 103001
2829
2830 006042
2831 006042 104073
2832 006044
2833 006044
2834 006044
2835
2836
2837
2838
2839
2840 006044
2841 006044 117700 173212
2842
2843 006050
2844 006050 032777 000200 173202
2845 006056 001401
2846
2847 006060
2848 006060 104074
2849 006062
2850 006062
2851 006062

```

```

*****
*****
*TEST 27 TEST THAT RCVRDONE IS CLEARED BY READING RBUF
*****
↑ST27: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #27,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX

; SET MAINT. BIT
LET @TCSR := @TCSR SET.BY #MAINT
BGNSUB
MOV #64,$SLPERR
; OUTPUT A CHARACTER WITH MAINTENANCE
; SET, AND WAIT FOR XMITRDY TO SET.

; OUTPUT A CHARACTER
LET @TBUF :B= #0
; WAIT MAXIMUM OF 500 MSEC
; FOR RCVRDONE TO SET IN
; RCSR
CALL TIMER IN (<#50,#RCVRDONE,RCSR,#SET>)

MOV R5,-(SP)
MOV #SET,-(R5)
MOV RCSR,-(R5)
MOV #RCVRDONE,-(R5)
MOV #50,-(R5)
JSR PC,TIMER
MOV (SP)+,R5

; DID IT BECAME READY?
IF.ERROR THEN
;RCVRDONE DID NOT SET IN RCSR
ERRHRD 59,, DIDNOT
ENDIF
ENDSUB

; NOW THAT IT IS SET LETS SEE IF READING THE
; BUFFER CLEARS RCVRDONE.

; READ BUFFER
LET RO :B= @RBUF
IF #RCVRDONE SETIN @RCSR THEN
;RCVRDONE DID NOT CLEAR IN RCSR
ERRHRD 60,, DIDNOT
ENDIF

$111:
$112:
ENDTST

```



```

2908                                     ;CHECK FOR TIMING OF RCVRACT. CLEARING
2909                                     ;VS RCVRDONE SETTING
2910
2911
2912                                     WHILE #RCVRACT SETIN @RCSR DO
2913 006212                                     $120:
2914 006212 032777 004000 173040          BIT    #RCVRACT,@RCSR
2915 006220 001416                                     BEQ    $121
2916
2917                                     IF #RCVRDONE SETIN @RCSR THEN
2918 006222 032777 000200 173030          BIT    #RCVRDONE,@RCSR
2919 006230 001411                                     BEQ    $122
2920                                     IF #RCVRACT SETIN @RCSR THEN
2921 006232 032777 004000 173020          BIT    #RCVRACT,@RCSR
2922 006240 001405                                     BEQ    $123
2923                                     ;RCVRDONE AND RCVRACT
2924                                     ;BOTH SET
2925 006242                                     ERRHRD 62, DONEACT
2926 006242 104076          ERROR    62
2927                                     ;NO USE CONTINUING
2928 006244                                     EXIT TST
2929 006244 012767 000001 172706          MOV    #1,$TIMES
2930 006252 000420          BR      TST$1          ;;EXIT THIS TEST
2931                                     ENOIF
2932 006254                                     $123:
2933 006254                                     ENOIF
2934 006254                                     $122:
2935 006254                                     ENDDO
2936 006254 000756          BR      $120
2937 006256                                     $121:
2938
2939                                     ;RCVRACT = 0 NOW.
2940                                     IF #RCVRDONE NOTSETIN @RCSR THEN
2941 006256 032777 000200 172774          BIT    #RCVRDONE,@RCSR
2942 006264 001001                                     BNE    $124
2943                                     ;RCVRDONE DID NOT SET IN RCSR
2944                                     ERRHRD 63,.DIDNOT
2945 006266 104077          ERROR    63
2946                                     ;SET IT BACK.
2947 006270                                     ENOIF
2948 006270                                     $124:
2949                                     ;TEST THAT READING THE RECEIVER
2950                                     ;BUFFER CLEARS RCVRDONE
2951
2952
2953                                     ;READ CHAR.
2954 006270          LET  R0 := @RBUF
2955 006270 017700 172766          MOV    @RBUF,R0
2956
2957                                     IF #RCVRDONE SETIN @RCSR THEN
2958 006274 032777 000200 172756          BIT    #RCVRDONE,@RCSR
2959 006302 001401                                     BEQ    $125
2960                                     ;RCVRDONE DID NOT CLEAR IN RCSR
2961 006304 104100          ERROR    64,.DIDNOT
2962 006304
2963 006306          ENOIF
    
```



G06

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 73  
DVDVAA.P11 06-MAY-77 15:29 T30 TEST THAT RCVRACT - RCSR :1 - SETS

SEQ 0071

2964 006306  
2965  
2966 006306  
2967 006306 000402  
2968 006310 070000  
2969 006312 000000  
2970  
2971 006314  
2972

\$125:

BR TST31  
MAX:70000  
COUNT: 0

EXIT  
;;;EXIT THIS TEST

ENDTST

H06

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 74  
DVDVAA.P11 06-MAY-77 15:29

T30 TEST THAT RCVRCT - RCSR 11 - SETS

SEQ 0072

2973  
 2974  
 2975  
 2976  
 2977  
 2978 006314 000004  
 2979 006316 012767 000010 172634  
 2980 006324 012767 000031 172646  
 2981  
 2982 006332  
 2983 006332 012767 006340 172550  
 2984  
 2985  
 2986  
 2987  
 2988  
 2989 006340  
 2990 006340 105077 172724  
 2991  
 2992 006344  
 2993 006344 010546  
 2994 006346 012745 000050  
 2995 006352 004767 003136  
 2996 006356 012605  
 2997  
 2998  
 2999 006360  
 3000 006360 105077 172704  
 3001  
 3002 006364  
 3003 006364 010546  
 3004 006366 012745 000050  
 3005 006372 004767 003116  
 3006 006376 012605  
 3007  
 3008  
 3009 006400  
 3010 006400 017704 172656  
 3011  
 3012  
 3013 006404  
 3014 006404 032704 040000  
 3015 006410 001005  
 3016  
 3017 006412  
 3018 006412 104101  
 3019  
 3020  
 3021 006414  
 3022 006414 012767 000001 172536  
 3023 006422 000500  
 3024 006424  
 3025 006424  
 3026 006424  
 3027  
 3028

```

*****
*****
*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
                                ;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 500 M SEC
                                WAITUS 50
        MOV     R5,-(SP)
        MOV     #50,-(R5)
        JSR     PC,WAIT
        MOV     (SP)+,R5
                                ;OUTPUT 2ND CHARACTER
                                LET  @TBUF :B= #0
                                ;LET OVERRUN HAPPEN
                                WAITUS 50
        MOV     R5,-(SP)
        MOV     #50,-(R5)
        JSR     PC,WAIT
        MOV     (SP)+,R5
                                ;READ BUFFER AND ERROR BITS
                                LET  R4 := @RBUF
                                ;IT DIDN'T SET
                                IF  #ORERR NOTSETIN R4 THEN
                                ;ORERR DID NOT SET IN RBUF
                                ERRHRD 65,,DIDNOT
                                ;NO USE COMPOUNDING ERRORS
                                EXIT TST
        MOV     #1,$TIMES
        BR      TST32          ;;EXIT THIS TEST
                                ENDIF
$126:
                                ENDSUB
                                ;NOW SEE IF ERROR BIT SET WITH OVERRUN ERROR:

```

```

3029 006424                                BGNSUB
3030 006424 012767 006432 172456          MOV    #64$,SLPERR
3031 006432                                IF #ERROR NOTSETIN R4 THEN
3032 006432 032704 100000                  BIT    #ERROR,R4
3033 006436 001005                          BNE    $127
3034
3035
3036 006440                                ;ERROR DID NOT SET IN RBUF
3037 006440 104102                          ERROR  66
3038
3039
3040
3041 006442                                ;--WHEN ORERR SET.
3042 006442 012767 000001 172510          MOV    #1,$TIMES
3043 006450 000465                          BR     TST32
3044 006452                                ;;;EXIT THIS TEST
3045 006452                                ENDIF
3046 006452                                $127:
3047
3048 006452                                ENDSUB
3049 006452 012767 006460 172430          MOV    #64$,SLPERR
3050
3051
3052
3053 006460                                ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
3054 006460 032777 040000 172574          BIT    #ORERR,RBUF
3055 006466 001002                          BNE    $130
3056
3057 006470                                IF #ORERR NOTSETIN RBUF THEN
3058 006470 104103                          ERROR  67
3059
3060
3061 006472                                ;READING RBUF CLEARED ORERR.
3062 006472 000454                          BR     TST32
3063 006474                                ;ERRHRD 67,ITCLRED
3064 006474                                ;SKIP REST OF TEST
3065
3066
3067 006474                                EXIT
3068 006474 012767 006502 172406          MOV    #64$,SLPERR
3069
3070
3071 006502                                ;;;EXIT THIS TEST
3072 006502 105077 172562                  CLR    RBUF
3073
3074 006506                                ENDIF
3075 006506 010546                                ENDSUB
3076 006510 012745 000050                                BGNSUB
3077 006514 004767 002774          MOV    R5, -(SP)
3078 006520 012605          MOV    #50, -(R5)
3079
3080
3081
3082
3083
3084 006532          JSR    PC, WAIT
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
531
```

# JOB

MAINDEC-11-DVDVA-A    MACY11 27(1006)    16-MAY-77 10:36 PAGE 76  
 DVDVA.A.P11    06-MAY-77 15:29    T31    TEST THE OVERRUN BIT - RBUF 14

SEQ 0074

```

3085 006532 104104                    ERROR 68
3086
3087                                    :-AFTER RECEIVING ANOTHER CHAR
3088                                    :SKIP AROUND REST
3089 006534                            EXIT TST
3090 006534 012767 000001 172416    MOV    #1,STIMES
3091 006542 000430                    BR     TST32                    ;;;EXIT THIS TEST
3092 006544                                                           ENDIF
3093 006544                            $I31:
3094
3095 006544                            IF #ERROR SETIN @RBUF THEN
3096 006544 032777 100000 172510    BIT    #ERROR,@RBUF
3097 006552 001401                    BEQ    $I32
3098                                    :ERROR DID NOT CLEAR IN RBUF
3099 006554                            ERRHRD 69,,DIDNOT
3100 006554 104105                    ERROR 69
3101
3102                                    ENDIF
3103 006556                            $I32:
3104 006556                                                           ENDSUB
3105 006556                                                           EXIT
3106 006556 000422                    BR     TST32                    ;;;EXIT THIS TEST
3107 006560 042522 042101 047111                                   ITCLRED:                    .ASCIZ /READING RBUF CLEARED CV
3108 006566 020107 041122 043125
3109 006574 041440 042514 051101
3110 006602 042105 047440 042526
3111 006610 051122 047125 042440
3112 006616 051122 051117 000056
3113                                    .EVEN
3114 006624                            ENDTST
3115
  
```

K06

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 77  
 DVDVAA.P11 06-MAY-77 15:29 T31 TEST THE OVERRUN BIT - RBUF 14

SEQ 0075

```

3116
3117
3118
3119
3120
3121
3122
3123
3124 006624 000004
3125 006626 012767 000010 172324
3126 006634 012767 000032 172336
3127 006642
3128 006642 032767 000200 172350
3129 006650 001004
3130 006652
3131 006652 012767 000001 172300
3132 006660 000542
3133 006662
3134 006662
3135
3136 006662
3137 006662 012767 177777 000272
3138 006670
3139 006670 012767 177777 000266
3140 006676
3141 006676 052777 000004 172360
3142
3143 006704
3144 006704 005003
3145 006706 000401
3146 006710
3147 006710 005203
3148 006712
3149 006712 020327 000017
3150 006716 003060
3151 006720
3152 006720 017700 172336
3153
3154 006724
3155 006724 116377 007104 172334
3156
3157 006732
3158 006732 005002
3159
3160 006734
3161 006734 005077 172330
3162
3163 006740
3164 006740 005067 000212
3165 006744
3166 006744 005067 000210
3167 006750
3168 006750
3169 006750 005702
3170 006752 001014
3171 006754

```

```

*****
*****
*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 $USWR THEN
BIT #PBR,$USWR
BNE $133
EXIT TST
MOV #1,$TIMES
BR TST33 ;;EXIT THIS TEST
ENDIF
$133:
LET OLD := #-1
MOV #-1,OLD
LET OLD+2 := #-1
MOV #-1,OLD+2
LET @TCSR := @TCSR SET.BY #MAINT
;EACH BAUD RATE
INCR R3 FROM #0 TO #15. BY #1
$135:
INC R3
$134:
CMP R3,#15.
BGT $136
LET R0 := @RBUF
;CHANGE BAUDE RATE
LET @TCSRHI := @RATES(R3)
;FLAG
LET BIT := #0
;OUTPUT THE CHARACTER
LET @TBUF := #0
;INITIALIZE COUNTER
LET NEW := #0
LET NEW+2 := #0
WHILE BIT EQ #0 DO
$137:
TST BIT
BNE $140
IF #RCVRDONE SETIN @RCSR THEN

```

```

3172 006754 022777 000200 172276 BIT #RCVRDONE,RCCSR
3173 006762 001403 BEQ $141
3174 ;DONE - ITS READY
3175 006764 LET BIT := #1
3176 006764 012702 000001 MOV #1,BIT
3177 006770 ELSE
3178 006770 000404 BR $142
3179 006772 $141:
3180 ;OTHERWISE-INCREMENT TIME
3181 006772 LET NEW := NEW + #1
3182 006772 005267 000160 INC NEW
3183 006776 LET NEW+2 := NEW+2 + CARRY
3184 006776 005567 000156 ADC NEW+2
3185 007002 ENDF
3186 007002 $142:
3187 ;SIGNALS DONE
3188 007002 ENDDO
3189 007002 000762 BR $137
3190 007004 $140:
3191
3192 007004 IF NEW+2 LO OLD+2 THEN
3193 007004 026767 000150 000152 CMP NEW+2,OLD+2
3194 007012 103001 BHIS $143
3195 ; OK
3196 007014 ELSE
3197 007014 000412 BR $144
3198 007016 $143:
3199 ; NEW+2 >= OLD+2
3200 IF NEW+2 EQ OLD+2 AND NEW LO OLD THEN
3201 007016 026767 000136 000140 CMP NEW+2,OLD+2
3202 007024 001005 BNE $145
3203 007026 026767 000124 000126 CMP NEW,OLD
3204 007034 103001 BHIS $145
3205 ;OK
3206 007036 ELSE
3207 007036 000401 BR $146
3208 007040 $145:
3209 ;NEW+2 > OLD+2 OR
3210 ;(NEW+2 = OLD+2 AND
3211 ;NEW >= OLD)
3212 ;BAUD RATE DIDN'T CHANGE
3213 007040 ERRHRD B6, BAUDRATE
3214 007040 104126 ERROR B6
3215 007042 ENDF
3216 007042 $146:
3217 007042 ENDF
3218 007042 $144:
3219 ;UPDATE OLD TIME
3220 007042 LET OLD := NEW
3221 007042 016767 000110 000112 MOV NEW,OLD
3222 007050 LET OLD+2 := NEW+2
3223 007050 016767 000104 000106 MOV NEW+2,OLD+2
3224
3225 007056 ENDINC ;BAUD RATE
3226 007056 000714 BR $135
3227 007060 $136:
    
```

M06

```

3228 007060          LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
3229 007060 116703 172135      MOVB $USWR+1,R3
3230 007064 110346          MOVB R3, -(SP)
3231 007066 142716 000017      BICB #17, (SP)
3232 007072 142603          BICB (SP)+,R3
3233 007074          LET @TCSRHI :B= RATES(R3)      ; LIKE HE WANTED IT
3234 007074 116377 007104 172164      MOVB RATES(R3),@TCSRHI
3235
3236 007102          EXIT ;SKIP TABLE
3237 007102 000431      BR TST33          ;;;EXIT THIS TEST
3238
3239 007104
  
```

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 'SUSWR' TO CAUSE THE CORRESPONDING  
 ;\*\* BAUD TO BE SELECTED IN THE DLV11-E UPON  
 ;\*\* COMPLETION OF THIS TEST.

					BAUD	OFFSET INTO TABLE
3247					50	0
3248	007104	010	R0050:	.BYTE	010	0
3249	007105	030	R0070:	.BYTE	030	1
3250	007106	050	R0110:	.BYTE	050	2
3251	007107	070	R0135:	.BYTE	070	3
3252	007110	110	R0150:	.BYTE	110	4
3253	007111	130	R0300:	.BYTE	130	5
3254	007112	150	R0600:	.BYTE	150	6
3255	007113	170	R0200:	.BYTE	170	7
3256	007114	210	R1800:	.BYTE	210	10
3257	007115	230	R2000:	.BYTE	230	11
3258	007116	250	R2400:	.BYTE	250	12
3259	007117	270	R3600:	.BYTE	270	13
3260	007120	310	R4800:	.BYTE	310	14
3261	007121	330	R7200:	.BYTE	330	15
3262	007122	350	R9600:	.BYTE	350	16
3263	007123	370	R10000:	.BYTE	370	17

```

3264
3265 007124 040502 042125 051040 BAUDRATE: .ASCIZ /BAUD RATE DIDN'T CHANGE./
3266 007132 052101 020105 044504
3267 007140 047104 052047 041440
3268 007146 040510 043516 027105
3269 007154 000
3270          .EVEN
3271 007156 000000 000000      NEW: 0,0
3272 007162 000000 000000      OLD: 0,0
3273 007166
  
```

ENDTST

3274  
3275  
3276

```

3277
3278
3279
3280
3281
3282
3283
3284
3285
3286 007166 000004
3287 007170 012767 000010 171762
3288 007176 012767 000033 171774
3289
3290 007204
3291 007204 005067 002372
3292
3293
3294 007210
3295 007210 016703 172042
3296
3297 007214
3298 007214 062703 000004
3299
3300 007220
3301 007220 010146
3302 007222 010301
3303 007224 012721 011574
3304 007230 012711 000340
3305 007234 012601
3306 007236
3307 007236 012767 007244 171644
3308
3309 007244
3310 007244 042777 000100 172012
3311
3312
3313 007252 012746 000000
3314 007256 012746 007264
3315 007262 000002
3316 007264
3317
3318
3319 007264
3320 007264 052777 000100 171772
3321
3322
3323 007272
3324 007272 010546
3325 007274 012745 000010
3326 007300 004767 002210
3327 007304 012605
3328
3329
3330 007306
3331 007306 026727 002270 000001
3332 007314 001406
    
```

```

*****
*****
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
*****
TST33: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #33,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
;;CLEAR 'INTERRUPT OCCURED' FLAG
LET INTFLAG := #0
;;GET VECTOR ADDRESS
LET R3 := DLVEC
;;FOR THE TRANSMITTER
LET R3 := R3 + #4
SETVEC R3, #INTSRV, #PR7
MOV R1, -(SP)
MOV R3, R1
MOV #INTSRV, (R1)+
MOV #PR7, (R1)
MOV (SP)+, R1
BGNSUB
MOV #645, $LPERR
LET #ATCSR := #ATCSR CLR.BY #XMITIE
;;SET IT TO 0
MOV #PRO, -(SP) ;;PUT NEW PS ON STACK
MOV #655, -(SP) ;;PUT NEW PC ON STACK
RTI ;;POP NEW PC AND PS
655:
LET #ATCSR := #ATCSR SET.BY #XMITIE
;;NOW SET I.E. BIT
LET #ATCSR := #ATCSR SET.BY #XMITIE
;;LET INTERRUPT HAVE TIME TO OCCUR
WAITUS 10
MOV R5, -(SP)
MOV #10, -(R5)
JSR PC, WAIT
MOV (SP)+, R5
;;DID EXACTLY 1 INTERRUPT OCCUR
IF INTFLAG NE #1 THEN
    
```



```

3333                                     ;NO - WAS IT 0 OR MORE THAN ONCE
3334 007316                                     IF INTFLAG EQ #0 THEN
3335 007316 005767 002260          TST      INTFLAG
3336 007322 001002          BNE      $150
3337                                     ;TRANSMITTER DID NOT INTERRUPT IN TIME
3338 007324                                     ERRHRD 70,,DIDNOT
3339 007324 104106          ERROR    70
3340 007326                                     ELSE
3341 007326 000401          BR      $151
3342 007330          $150:
3343                                     ;TWICE
3344                                     ;TRANSMITTER INTERRUPTED TWICE
3345 007330                                     ERRHRD 71,,TWICE
3346 007330 104107          ERROR    71
3347 007332                                     ENDIF
3348 007332          $151:
3349 007332                                     ENDIF
3350 007332          $147:
3351 007332
3352                                     ENDSUB
3353                                     ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
3354 007332 012767 007340 171550      MOV     #64$, $LPERR
3355                                     BGNSUB
3356 007340                                     ;CLEAR 'INTERRUPT OCCURED' FLAG
3357 007340 005067 002236          CLR     INTFLAG
3358                                     LET INTFLAG := #0
3359 007344                                     ;CLEAR INTERRUPT ENABLE
3360 007344 042777 000100 171712      BIC     #XMITIE, @TCSR
3361                                     LET @TCSR := @TCSR CLR.BY #XMITIE
3362 007352 012746 000000          MOV     #PRO, -(SP)      ;;PUT NEW PS ON STACK
3363 007356 012746 007364          MOV     #65$, -(SP)    ;;PUT NEW PC ON STACK
3364 007362 000002          RTI     ;;POP NEW PC AND PS
3365 007364          65$:
3366                                     ;DARE IT TO HAPPEN
3367 007364                                     WAITUS 10
3368 007364 010546          MOV     R5, -(SP)
3369 007366 012745 000010          MOV     #10, -(R5)
3370 007372 004767 002116          JSR     PC, WAIT
3371 007376 012605          MOV     (SP)+, R5
3372 007400                                     IF INTFLAG NE #0 THEN
3373 007400 005767 002176          TST     INTFLAG
3374 007404 001401          BEQ     $152
3375                                     ;INTERRUPT OCCURED WITH I E CLEARED
3376 007406                                     ERRHRD 72,NOTENAB
3377 007406 104110          ERROR    72
3378 007410                                     ENDIF
3379 007410          $152:
3380 007410                                     BRESET
3381 007410 000005          RESET
3382 007412                                     ENDSUB
3383                                     ;RESTORE VECTOR AREA
3384 007412                                     CLRVEC R3
3385 007412 010146          MOV     R1, -(SP)      ;;PUSH R1 ON STACK
3386 007414 010246          MOV     R2, -(SP)      ;;PUSH R2 ON STACK
3387 007416 012701 000003          MOV     #R3, R1
3388 007422 010102          MOV     R1, R2
    
```

C07

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 82  
DVDVA.P11 06-MAY-77 15:29 T33 TRANSMITTER INTERRUPT LOGIC TEST

SEQ 0080

```
3389 007424 062702 000002      ADD     #2,R2
3390 007430 010221      MOV     R2,(R1)+
3391 007432 005011      CLR     (R1)
3392 007434 012602      MOV     (SP)+,R2      ;;POP STACK INTO R2
3393 007436 012601      MOV     (SP)+,R1      ;;POP STACK INTO R1
3394
3395 007440      ENDTST
3396
3397
3398
3399
3400
3401
```

3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411  
3412  
3413  
3414  
3415  
3416  
3417  
3418  
3419  
3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457

007440 000004  
007442 012767 000010 171510  
007450 012767 000034 171522  
  
007456  
007456 010146  
007460 016701 171572  
007464 012721 011574  
007470 012711 000340  
007474 012601  
  
007476  
007476 012767 007504 171404  
007504  
007504 005007 002072  
  
007510  
007510 052777 000004 171546  
  
007516  
007516 042777 000100 171534  
  
007524 012746 000000  
007530 012746 007536  
007534 000002  
007536  
  
007536  
007536 105077 .71526  
  
007542  
007542 010546  
007544 012745 177777  
007544 016745 171510  
007554 012745 000200  
007560 012745 000050  
007564 004767 001446  
007570 012605  
  
007572  
007572 052777 000100 171460  
  
007600  
007600 010546

```
*****  
*****  
*TEST 34 RECEIVER INTERRUPT LOGIC TEST  
* THIS TEST COVERS ALL OF THE RECEIVER  
* SIDE OF THE INTERRUPT LOGIC, BOTH DATASET  
* AND CHARACTER MODES.  
*****  
TST34: SCOPE  
MOV #10,$TIMES ;;DO 10 ITERATIONS  
MOV #34,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX  
;;CLEAR INTERRUPT OCCURED FLAG  
;;SET UP RECEIVER INTER.VECTOR  
SETVEC DLVEC,#INTSRV,#PR7  
MOV R1,-(SP)  
MOV DLVEC,R1  
MOV #INTSRV(R1)+  
MOV #PR7(R1)  
MOV (SP)+,R1  
;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE  
BGNSUB  
MOV #64,$SLPERR  
LET INTFLAG := #0  
;SET MAINT. BIT  
LET @TCSR := @TCSR SET.BY #MAINT  
;CLEAR INTERRUPTS  
LET @RCR := @RCR CLR.BY #RCVRIE  
;CHANGE PRIORITY  
; TO 0  
MOV #PRO,-(SP) ;;PUT NEW PS ON STACK  
MOV #65,-(SP) ;;PUT NEW PC ON STACK  
RTI ;;POP NEW PC AND PS  
  
65$:  
  
;SEND A CHARACTER  
LET @TBUF :B= #0  
  
;WAIT A MAXIMUM  
;OF 50 MSEC FOR  
;XMIT RDY TO SET IN TCSR  
CALL TIMER IN (<#50,#XMITRDY,TCSR,#SET)  
  
MOV R5,-(SP)  
MOV #SET,-(R5)  
MOV TCSR,-(R5)  
MOV #XMITRDY,-(R5)  
MOV #50,-(R5)  
JSR PC,TIMER  
MOV (SP)+,R5  
  
;SET INTERRUPT ENABLE  
LET @RCR := @RCR SET.BY #RCVRIE  
  
;LET IT COME IN.  
WAITUS 10  
  
MOV R5,-(SP)
```

E07

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 84  
 DVDVAA.P11 06-MAY-77 15:29 T34 RECEIVER INTERRUPT LOGIC TEST

SEQ 0082

```

3458 007602 012745 000010      MOV      #10,-(RS)
3459 007606 004767 001702      JSR      PC,WAIT
3460 007612 012605                MOV      (SP)+,R5
3461
3462                                ;DID HE DO IT RIGHT?
3463 007614                                IF INTFLAG NE #1 THEN
3464 007614 026727 001762 000001      CMP      INTFLAG,#1
3465 007622 001406                BEQ      $153
3466                                ;NONE OCCURED
3467 007624                                IF INTFLAG EQ #0 THEN
3468 007624 005767 001752      TST      INTFLAG
3469 007630 001002                BNE      $154
3470                                ;RECEIVER DID NOT INTERRUPT IN TIME
3471 007632                                ERRHRD 73,,DIDNOT
3472 007632 104111                ERROR    73
3473                                ;TWICE OR MORE
3474 007634                                ELSE
3475 007634 000401                BR       $155
3476 007636                $154:
3477                                ;RECEIVER INTERRUPTED TWICE
3478 007636                                ERRHRD 74,,TWICE
3479 007636 104112                ERROR    74
3480 007640                                ENDF
3481 007640                $155:
3482 007640                                ENDF
3483 007640                $153:
3484                                ;RESET MAINT. BIT.
3485 007640                                LET @TCSR := @TCSR CLR.BY #MAINT
3486 007640 042777 000004 171416      BIC      #MAINT,@TCSR
3487 007646                                ENDSUB
3488
3489
3490
3491
3492
3493
3494                                ; INTERRUPT WITHOUT I E SET.
3495 007646                                BGNSUB
3496 007646 012767 007654 171234      MOV      #64$,$LPERR
3497 007654                                LET @TCSR := @TCSR SET.BY #MAINT
3498 007654 052777 000004 171402      BIS      #MAINT,@TCSR
3499
3500                                ;CLEAR INTERRUPT FLAG
3501 007662                                LET INTFLAG := #0
3502 007662 005067 001714      CLR      INTFLAG
3503                                ;CLEAR INTERRUPT
3504 007666                                LET @RCR := @RCR CLR.BY #RCVRIE
3505 007666 042777 000100 171364      BIC      #RCVRIE,@RCR
3506                                ;DON'T LET THEM IN
3507 007674 012746 000340      MOV      #PR7,-(SP)      ;;PUT NEW PS ON STACK
3508 007700 012746 007706      MOV      #65$,-(SP)     ;;PUT NEW PC ON STACK
3509 007704 000002                RTI                    ;;POP NEW PC AND PS
3510 007706                65$:
3511                                ;DARE IT
3512 007706                                WAITUS 10
3513 007706 010546                MOV      RS,-(SP)

```

```

3514 007710 012745 000010      MOV      #10,-(RS)
3515 007714 004767 001574      JSR      PC,WAIT
3516 007720 012605              MOV      (SP)+,RS
3517
3518 007722              :DID IT HAPPEN?
3519 007722 005767 001654      TST      INTFLAG
3520 007726 001401              BEQ      $156
3521
3522 007730              :INTERRUPT OCCURED AT PRIORITY 7
3523 007730 104115              ERROR    77
3524 007732              ENDIF
3525 007732              :CLEAR THE WORLD
3526
3527 007732              BRESSET
3528 007732 000005              RESET
3529
3530
3531 007734              :RESET MAINT. BIT.
3532 007734 042777 000004 171322      BIC      @MAINT,@TCSR
3533 007742              LET @TCSR := @TCSR CLR.BY @MAINT
3534
3535              :PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3536 007742              BGNSUB
3537 007742 012767 007750 171140      MOV      @64S,$LPERR
3538 007750              IF @CABLE NOTSETIN $USWR THEN
3539 007750 032767 020000 171242      BIT      @CABLE,$USWR
3540 007756 001004              BNE      $157
3541
3542 007760              :CAN'T TEST WITHOLT A CABLE
3543 007760 012767 000001 171172      MOV      #1,$TIMES
3544 007766 000463              BR       TST$35
3545 007770              :::EXIT THIS TEST
3546 007770              ENDIF
3547
3548 007770              :CLEAR 'INTFLAG'
3549 007770 005067 001606      CLR      INTFLAG
3550
3551 007774              :CLEAR INTERRUPTS
3552 007774 042777 000040 171256      BIC      @DATAIE,@RCSR
3553
3554              :CHANGE PRIORITY
3555 010002 012746 000000      MOV      @PRC,-(SP)
3556 010006 012746 010014      MOV      @64S,-(SP)
3557 010012 000002              RTI
3558 010014              :::PUT NEW PS ON STACK
3559 010014              :::PUT NEW PC ON STACK
3560 010014 042777 000004 171236      BIC      @REQSEND,@RCSR
3561
3562 010022              :SET INTERRUPT ENABLE
3563 010022 052777 000040 171230      BIS      @DATAIE,@RCSR
3564 010030              LET @RCSR := @RCSR SET.BY @DATAIE
3565 010030 052777 000004 171222      BIS      @REQSEND,@RCSR
3566
3567 010036              :LET IT COME IN.
3568 010036 010546              WAITUS 10
3569 010040 012745 000010      MOV      RS,-(SP)
3569 010040 012745 000010      MOV      #10,-(RS)
    
```

\$156:

\$157:

64S:

G07

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 86  
DVDVAA.P11 06-MAY-77 15:29 T34 RECEIVER INTERRUPT LOGIC TEST

SEG 0084

```

3570 010044 004767 001444 JSR PC WAIT
3571 010050 012605 MOV (SP)+,R5
3572
3573
3574 010052 ; DID IT DO IT RIGHT?
3575 010052 026727 001524 000001 CMP INTFLAG,#1
3576 010060 001406 BEQ $160
3577 ; NONE OCCURED
3578 010062 IF INTFLAG EQ #0 THEN
3579 010062 005767 001514 YST INTFLAG
3580 010066 001002 BNE $161
3581 ; DATAINT DID NOT INTERRUPT IN TIME
3582 010070 ERRHRD 75,,DIDNOT
3583 010070 104113 ERROR 75
3584 ; TWICE OR MORE
3585 010072 ELSE
3586 010072 000401 BR $162
3587 010074 $161:
3588 ; DATAINT INTERRUPTED TWICE
3589 010074 ERRHRD 76,,TWICE
3590 010074 104114 ERROR 76
3591 010076 ENDF
3592 010076 $162:
3593 010076 ENDF
3594 010076 $160:
3595 010076 LET @RCSR := @RCSR CLR.BY @DATAIE
3596 010076 042777 000040 171154 BIC @DATAIE,@RCSR
3597 010104 ENDSUB
3598
3599 010104 LET R4 := @DLVEC
3600 010104 017704 171146 MOV @DLVEC,R4
3601 010110 CLRVEC R4
3602 010110 010146 MOV R1,-(SP) ;; PUSH R1 ON STACK
3603 010112 010246 MOV R2,-(SP) ;; PUSH R2 ON STACK
3604 010114 12701 000004 MOV @R4,R1
3605 010120 010102 MOV R1,R2
3606 010122 062702 000002 ADD #2,R2
3607 010126 010221 MOV R2,(R1)+
3608 010130 005011 CLR (R1)
3609 010132 012602 MOV (SP)+,R2 ;; POP STACK INTO R2
3610 010134 012601 MOV (SP)+,R1 ;; POP STACK INTO R1
3611 010136 ENDTST

```

H07

```

3612 .....*****
3613 .....*****
3614 .....*****
3615 *TEST 35 TEST ACTUAL DATA TRANSFERED
3616 * NON-INTERRUPT MAINTENANCE BIT SET
3617 .....*****
3617 010136 000004 †ST35: SCOPE
3618 010140 012767 000001 171012 MOV #1,$TIMES ;;DO 1 ITERATION
3619 010146 012767 000035 171024 MOV #35,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
3620 ;;SET MAINT. BIT
3621 010154 052777 000004 171102 BIS #MAINT,$TCSR LET $TCSR := $TCSR SET.BY #MAINT
3622 010154 052777 000004 171102
3623
3624 ;;CHANGE PRIORITY
3625 ;; TO 0
3626 010162 012746 000000 MOV #PRO,-(SP) ;;PUT NEW PS ON STACK
3627 010166 012746 010174 MOV #645,-(SP) ;;PUT NEW PC ON STACK
3628 010172 000002 RTI ;;POP NEW PC AND PS
3629 010174
3630
3631 ;;GET DATA MASK.
3631 010174 CALL DATLNG OUT <R1>
3632 010174 162705 000002 SUB #1*2,R5
3633 010200 004767 001210 JSR PC,DATLNG
3634 010204 012501 MOV (R5)+,R1
3635
3636
3637 ;;ALL BINARY CHAR.
3637 010206 INCR R2 FROM #0 TO #377 BY #1
3638 010206 005002 CLR R2
3639 010210 000401 BR $163
3640 010212 $164: INC R2
3641 010212 005202
3642 010214 $163: CMP R2,#377
3643 010214 020227 000377 BGT $165
3644 010220 003023
3645
3646
3647 ;;TRANSMIT CHAR IN R2
3648
3649 REPEAT
3649 010222
3650 010222 $166:
3651 ;;IS IT READY?
3652 010222 UNTIL #XMITRDY SETIN $TCSR
3653 010222 032777 000200 171034 BIT #XMITRDY,$TCSR
3654 010230 001774 BEQ $166
3655
3656 ;;TRANSMIT IT
3657 010232 LET $TBUF := R2
3658 010232 110277 171032 MOVB R2,$TBUF
3659
3660 REPEAT
3661 010236 $167:
3662 ;;WAIT TILL ITS HERE
3663 010236 UNTIL #RCVRDONE SETIN $RCSR
3664 010236 032777 000200 171014 BIT #RCVRDONE,$RCSR
3665 010244 001774 BEQ $167
3666
3667 ;;AND SAVE IT
3667 010246 LET R3 := $RBUF

```

```

3668 010246 017703 171010      MOV     @RBUF,R3
3669
3670
3671                                ;COMPARE TO SEE IF WE RECEIVED IT ALL
3672                                ;CLEAN OFF NON-DATA BITS
3673                                ;ON BOTH TRANSMITTED AND
3674                                LET R4 := R2 CLR.BY R1
3675 010252
3676 010252 010204      MOV     R2,R4
3677 010254 040104      BIC     R1,R4
3678 010256
3679 010256 040103      BIC     R1,R3
3680
3681                                ;RECEIVED DATA
3682                                IF R4 NE R3 THEN
3683 010260 020403      CMP     R4,R3
3684 010262 001401      BEQ     $170
3685
3686 010264
3687 010264 104116      ERROR  78
3688                                ;<TRANSMITTED> <RECEIVED>
3689 010266                                ENDIF
3690 010266                                $170:
3691 010266                                ENDINC ; R2
3692 010266 000751      BR     $164
3693 010270                                $165:
3694
3695                                ;RESET MAINT. BIT.
3696 010270
3697 010270 042777 000004 170766      BIC     #MAINT,@TCSR
3698 010276
3699
3700
3701
    
```



J07

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 89  
 DVDVA.P11 06-MAY-77 15:29

T35 TEST ACTUAL DATA TRANSFERED

SEG 0087

```

3702
3703
3704
3705
3706 010276 000004
3707 010300 012767 000001 170652
3708 010306 012767 000036 170664
3709 010314
3710 010314 032767 020000 170676
3711 010322 001004
3712
3713 010324
3714 010324 012767 000001 170626
3715 010332 000456
3716 010334
3717 010334
3718
3719 010334
3720 010334 042777 000004 170722
3721
3722
3723 010342 012746 000000
3724 010346 012746 010354
3725 010352 000002
3726 010354
3727
3728 010354
3729 010354 162705 000002
3730 010360 004767 001030
3731 010364 012501
3732 010366
3733 010366 017700 170670
3734
3735 010372
3736 010372 005002
3737 010374 000401
3738 010376
3739 010376 005202
3740 010400
3741 010400 020227 000377
3742 010404 003031
3743
3744
3745
3746
3747 010406
3748 010406
3749
3750 010406
3751 010406 032777 000200 170650
3752 010414 001774
3753
3754
3755 010416
3756 010416 110277 170646
3757

```

```

*****
*****
*TEST 36 TEST DATA THROUGH CABLE
*****
TST36: SCOPE
MOV #1,STIMES ;;DO 1 ITERATION
MOV #36,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
IF #CABLE, SUSWR THEN
;CAN'T TEST WITHOUT A CABLE
EXIT TST
MOV #1,STIMES
BR TST37 ;;EXIT THIS TEST
ENDIF
$171:
;DON'T USE MAINT.
LET @TCSR := @TCSR CLR.BY #MAINT
;CHANGE PRIORITY
;... TO 0
MOV #PRO,-(SP) ;;PUT NEW PS ON STACK
MOV #64$,-(SP) ;;PUT NEW PC ON STACK
RTI ;;POP NEW PC AND PS
;GET DATA MASK
CALL DATLNG OUT <R1>
SUB #1*2,R5
JSR PC,DATLNG
MOV (R5)+,R1
LET R0 := @RBUF ; START CLEAN
;BINARY COUNT PATTERN
INCR R2 FROM #0 TO #377 BY #1
CLR R2
BR $172
$173: INC R2
$172: CMP R2,#377
BGT $174
;TRANSMIT THE CHAR. IN R2.
REPEAT
;WAIT ON READY
UNTIL #XMITRDY SETIN @TCSR
BIT #XMITRDY,@TCSR
BEQ $175
;START IT ON ITS WAY
LET @TBUF :B= R2
;GIVE IT A HEAD START.

```

K07

```

3758 010422
3759 010422 010546
3760 010424 012745 000010
3761 010430 004767 001060
3762 010434 012605
3763 010436
3764 010436 $176:
3765
3766 010436
3767 010436 032777 000200 170614
3768 010444 001774
3769
3770
3771 010446
3772 010446 017703 170610
3773
3774
3775 010452
3776 010452 010204
3777 010454 040104
3778 010456
3779 010456 040103
3780
3781
3782 010460
3783 010460 020403
3784 010462 001401
3785
3786 010464
3787 010464 104117
3788
3789
3790 010466
3791 010466 $177:
3792 010466
3793 010466 000743
3794 010470 $174:
3795
3796
3797
3798 010470
3799
3800
3801
3802

```

```

MOV R5, -(SP)
MOV #10, -(R5)
JSR PC, WAIT
MOV (SP)+, R5

REPEAT
; IS IT HERE
UNTIL #RCVRDONE SETIN #RCSR

; RETRIEVE
LET R3 := #RE.

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

MOV R2, R4
BIC R1, R4
LET R3 := R3 CLR. BY R1

BIC R1, R3

; WE HAVE TROUBLE
IF R4 NE R3 THEN

; DATA COMPARE ERROR
ERRHAD 79, COMP, SBWAS

; <R2> <R3>
ENDIF

ENDINC ; R2

BR $173

ENDTST

```

L07

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 91  
DVDVAA.P11 06-MAY-77 15:29 T36 TEST DATA THROUGH CABLE

SEQ 0089

```

3803 .....
3804 .....
3805 .....
3806 .....
3807 .....
3808 010470 000004 .....
3809 010472 012767 000001 170460 .....
3810 010500 012767 000037 170472 .....
3811 .....
3812 .....
3813 010506 .....
3814 010506 162705 000002 .....
3815 010512 004767 000676 .....
3816 010516 012503 .....
3817 .....
3818 .....
3819 .....
3820 .....
3821 .....
3822 .....
3823 .....
3824 .....
3825 .....
3826 .....
3827 .....
3828 010520 012746 000000 .....
3829 010524 012746 010532 .....
3830 010530 000002 .....
3831 010532 .....
3832 .....
3833 010532 .....
3834 010532 016701 170520 .....
3835 .....
3836 010536 .....
3837 010536 012721 010710 .....
3838 010542 .....
3839 010542 012721 000340 .....
3840 .....
3841 .....
3842 010546 .....
3843 010546 012721 010646 .....
3844 010552 .....
3845 010552 012711 000340 .....
3846 .....
3847 .....
3848 010556 .....
3849 010556 005001 .....
3850 .....
3851 010560 .....
3852 010560 005002 .....
3853 .....
3854 010562 .....
3855 010562 005004 .....
3856 .....
3857 010564 .....
3858 010564 000005 .....

```

```

*****
*****
*TEST 37 FULL DATA TRANSFER WITH INTERRUPTS
* AND MAINTENANCE MODE.
*****
↑ST37: SCOPE
MOV #1,STIMES ;;DO 1 ITERATION
MOV #37,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
;;GET DATA MASK
CALL DATLNG OUT (R3)
SUB #1*2,R5
JSR PC,DATLNG
MOV (R5)+,R3
;SINCE THIS WILL BE A FULL SPEED INTERRUPT TEST
;IF A DATA COMPARE ERROR OCCURS THE REPORT
;WILL BE DEFERRED UNTIL ALL DATA PATTERNS HAVE
;BEEN TRANSFERED. IF THE NUMBER OF CHAR. TRANSMITTED
;DOES NOT EQUAL THE NUMBER RECEIVED THIS ALSO WILL
;BE REPORTED.
;CHANGE PRIORITY
;...TO 0
MOV #PRO,-(SP) ;;PUT NEW PS ON STACK
MOV #64$,-(SP) ;;PUT NEW PC ON STACK
RTI ;;POP NEW PC AND PS
64$:
;GET VECTOR ADDRESS
LET R1 := DLVEC
;RCVR VECTOR
LET (R1)+ := #REC
LET (R1)+ := #PR7
;POINT TO TRANSMITTER VECTOR
;AND SET IT UP ALSO
LET (R1)+ := #TRAN
LET (R1) := #PR7
;START COUNT AT 0
LET R1 := #0
;RECEIVER STORAGE
LET R2 := #0
;# OF RECEIVED CHAR. COUNT.
LET R4 := #0
BRESET ;SET UP ALL REGISTERS
RESET

```

```

3859                                     :SET UP MAINTENANCE
3860 010566                               LET @TCSR := @TCSR SET.BY @MAINT
3861 010566 052777 000004 170470      BIS      @MAINT,@TCSR
3862
3863                                     :SET I.E. IN TRANSMITTER
3864 010574                               LET @TCSR := @TCSR SET.BY @XMITIE
3865 010574 052777 000100 170462      BIS      @XMITIE,@TCSR
3866
3867                                     :AND RECEIVER
3868 010602                               LET @RCR := @RCR SET.BY @RCVRIE
3869 010602 052777 000100 170450      BIS      @RCVRIE,@RCR
3870
3871                                     :NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3872 010610                               REPEAT
3873 010610                               UNTIL R4 HIS NUMBER
3874 010610                               $200:
3875 010610 020467 000026              CMP      R4,NUMBER
3876 010614 102775                      BLO      $200
3877
3878                                     :DATA COMPARE ERRORS.
3879 010616                               IF ERRCNT NE #0 THEN
3880 010616 005767 000016              TST      ERRCNT
3881 010622 001405                      BEQ      $201
3882
3883                                     :SET UP FOR ERROR PRINTING
3884 010624                               LET R2 := SB
3885 010624 016702 000014              MOV      SB,R2
3886 010630                               LET R3 := WAS
3887 010630 016703 000011              MOV      WAS,R3
3888
3889                                     :DATA COMPARE ERROR
3890 010634                               ERRHRD 80,COMP,FIRST
3891 010634 104120                      ERROR   80
3892
3893                                     $201:
3894 010636                               ENDIF
3895 010636                               :<R3> OCCURED, FIRST: SB <SB>, WAS <WAS>
3896 010636 000462                      BR       TST40
3897 010640                               :SKIP OVER SUPPORT ROUTINES & STORAGE
3898 010642 001000                      EXIT
3899 010644 000
3900 010645 000
3901
3902
3903
3904                                     :TRANSMIT INTERRUPT HANDLER
3905 010646                               BGNSRV  TRAN
3906 010646                               TRAN:
3907
3908                                     :SET UP FOR TRANSFER
3909 010646                               LET HOLD := R1      CLR.BY R3
3910 010646 010167 000032              MOV      R1,HOLD
3911 010652 040367 000026              BIC      R3,HOLD
3912
3913                                     :AND SEND.
3914 010656                               LET @TBUF := HOLD
3914 010656 016777 000022 170404      MOV      HOLD,@TBUF

```

```

3915
3916 010664
3917 010664 005201          INC    R1
3918
3919 010666
3920 010666 020167 177750    CMP    R1,NUMBER
3921 010672 001003          BNE    $202
3922
3923 010674
3924 010674 042777 000100 170362  BIC    #XMITIE,@TCSR
3925 010702
3926 010702          $202:
3927
3928 010702 000401          BR     ZZZ
3929
3930 010704 000000
3931
3932 010706          ZZZ:
3933 010706 000002          RTI
3934
3935
3936
3937
3938 010710
3939 010710          REC:
3940
3941
3942 010710
3943 010710 017702 170346          MOV    @RBUF,R2
3944 010714 040302          BIC    R3,R2
3945
3946 010716
3947 010716 010467 000056          MOV    R4,RHLD
3948 010722 040367 000052          BIC    R3,RHLD
3949
3950
3951 010726
3952 010726 020267 000046          CMP    R2,RHLD
3953 010732 001412          BEQ    $203
3954
3955 010734
3956 010734 005767 177700          TST    ERRCNT
3957 010740 001005          BNE    $204
3958
3959 010742
3960 010742 116767 000032 177674          MOVB  RHLD,SB
3961 010750
3962 010750 110267 177671          MOVB  R2,WAS
3963 010754
3964 010754          $204:
3965
3966 010754
3967 010754 005267 177660          INC    ERRCNT
3968 010760
3969 010760          $203:
3970

```

```

: INCREMENT CHAR COUNT
LET R1 := R1 + #1

: ALL DONE
IF R1 EQ NUMBER THEN

: STOP INTERRUPT PROCESSING
LET @TCSR := @TCSR CLR.BY #XMITIE

ENDIF

; EXIT SRV
HOLD:0

ENDSRV

: RECEIVER INTERRUPT HANDLER
BGNSRV REC

: GET CHAR IN + MASK IT
LET R2 := @RBUF CLR.BY R3

: RHLD WILL CONTAIN EXPECTED INPUT
LET RHLD := R4 CLR.BY R3

: DO THEY COMPARE
IF R2 NE RHLD THEN

: FIRST ERROR
IF ERRCNT EQ #0 THEN

; SAVE RECORD OF FIRST MISS
LET SB :B= RHLD

LET WAS :B= R2

ENDIF

: COUNT IT.
LET ERRCNT := ERRCNT + #1

ENDIF

```

```

3971
3972 010760
3973 010760 005204          INC    R4
3974
3975 010762
3976 010762 020467 177654    CMP    R4,NUMBER
3977 010766 001003          BNE    $205
3978
3979 010770
3980 010770 042777 000100 170262  BIC    #RCVRIE,#RCSR
3981
3982 010776
3983 010776          $205:
3984
3985
3986 010776 000401          BR     ZZZZ
3987
3988 011000 000000
3989 011002          ZZZZ:
3990 011002
3991 011002 000002          PTI
3992
3993 011004          ENDSRV
3994
3995
3996

```

:COUNT THIS CHAR.  
LET R4 := R4 + #1  
:ALL DONE?  
IF R4 EQ NUMBER THEN  
:STOP RECEIVER INTERRUPTS  
LET @RCSR := @RCSR CLR.BY #RCVRIE  
:MAIN REPEAT LOOP IS CHECKING  
ENDIF  
:FOR 'R4 = NUMBER' ALSO  
; EXIT SRV  
RHL0:0  
ENDTST

```

3997
3998
3999
4000
4001
4002
4003 011004 000004
4004 011006 012767 000010 170144
4005 011014 012767 000040 170156
4006
4007 011022
4008 011022 052777 000004 170234
4009
4010 011030
4011 011030 052777 000001 170226
4012
4013 011036
4014 011036 012777 000252 170224
4015 011044
4016 011044
4017
4018 011044
4019 011044 032777 000200 170206
4020 011052 001774
4021
4022 011054
4023 011054 105777 170202
4024 011060 001401
4025
4026 011062
4027 011062 104121
4028 011064
4029 011064
4030 011064
4031 011064 000005
4032 011066
4033 011066 000413
4034 011070 051102 040505 020113
4035 011076 044504 020104 047516
4036 011104 020124 050505 040525
4037 011112 020114 000060
4038
4039 011116

```

```

*****
*****
*TEST 40 TEST BREAK GENERATION LOGIC
* TRANSMIT KNOWN CHAR WITH BREAK SET
* AND COMPARE RECEIVED WITH 0.
*****
TST40: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #40,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
;;SET MAINTENANCE BIT
LET @TCSR := @TCSR SET BY #MAINT
;;SET BREAK BIT
LET @TCSR := @TCSR SET BY #BREAK
;;NON-ZERO CHAR. '*'
LET @TBUF := #252
REPEAT
;;WAIT FOR DONE
UNTIL #RCVRDONE SET IN @RCR
BIT #RCVRDONE,@RCR
BEQ $206
IFB @RBUF NE #0 THEN
;;BREAK DID NOT EQUAL 0
ERRHRD B1 ,BADBRK
ENDIF
$207: BRESET ;CLEAN UP
RESET
BR TST41 ;;EXIT THIS TEST
BADBRK: .ASCIZ /BREAK DID NOT EQUAL 0/

ENDTST

```

4040									
4041									
4042									
4043									
4044	011116	000004							
4045	011120	012767	000001	170032					
4046	011126	104401	011134						
4047	011132	000404							
4048									
4049	011144								
4050	011144	016746	170104						
4051	011150	104402							
4052	011152	104401	011160						
4053	011156	000405							
4054									
4055	011172								
4056	011172	016746	170060						
4057	011176	104402							
4058	011200	104401	011206						
4059	011204	000405							
4060									
4061	011220								
4062	011220	016746	167666						
4063	011224	104405							
4064	011226	005067	167660						
4065	011232	000167	170462						

```

*****
; TEST 41      NOT A TEST - SEND BACK TO LOOP
*****
TST41:  SCOPE
        MOV     #1,STIMES      ;; DO 1 ITERATION
        TYPE   ,65$           ;; TYPE ASCIZ STRING
        BR     ,64$           ;; GET OVER THE ASCIZ
;:65$:  .ASCIZ  <CRLF>*CSR: *
64$:   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

```



E08

4066  
4067  
4068  
4069 011236  
4070 011236  
4071  
4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091 000001  
4092 000000  
4093  
4094 011236  
4095 011236 016567 000004 000136  
4096 011244  
4097 011244 016567 000000 000132  
4098 011252  
4099 011252 112767 000000 000126  
4100  
4101  
4102  
4103  
4104 011260  
4105 011260  
4106  
4107 011260  
4108 011260 035577 000002 000114  
4109 011266 001004  
4110 011270  
4111 011270 112767 000000 000111  
4112 011276  
4113 011276 000403  
4114 011300  
4115 011300  
4116 011300 112767 177777 000101  
4117 011306  
4118 011306  
4119  
4120  
4121 011306

```
;;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  
$212: ; TEST TO SEE IF WHICHBIT IS SET  
IF WHICHBIT(R5) NOTSETIN @REGSAV THEN  
LET HOLDSC :B= #CLR  
ELSE  
$214: LET HOLDSC :B= #SET ; REMEMBER THIS  
ENDIF  
$215: ; NOW SEE IF THAT WAS WHAT WE WANTED  
IFB HOLDSC EQ SETCLR(R5) THEN
```

```

4122 011306 126765 000075 000006      CMPB  HOLDSC,SETCLR(R5)
4123 011314 001003                    BNE   $216
4124                                ; JUST THE THING WE NEEDED
4125 011316                                LET   FLAG :B= #TRUE
4126 011316 112767 000001 000062      MOVB  #TRUE,FLAG
4127 011324                                ENDF
4128 011324                                $216:
4129
4130 011324                                EXIFB FLAG EQ #TRUE OR TIMSAV LE #0
4131 011324 126727 000056 000001      CMPB  FLAG,#TRUE
4132 011332 001414                    BEQ   $213
4133 011334 005767 000044            TST   TIMSAV
4134 011340 003411                    BLE   $213
4135                                ; ONE WAY OR THE OTHER, WE ARE DONE
4136                                ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4137
4138 011342                                WAITUS 10          ;WAIT FOR 10 MILLI-SECONDS
4139 011342 010546                    MOV   RS, -(SP)
4140 011344 012745 000010            MOV   #10, -(R5)
4141 011350 004767 000140            JSR   PC, WAIT
4142 011354 012605                    MOV   (SP)+,R5
4143 011356                                LET   TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4144 011356 005367 000022            DEC   TIMSAV
4145 011362                                ENDF
4146 011362 000736                    BR    $212
4147 011364                                $213:
4148
4149                                ; ONLY 2 WAYS TO GET HERE
4150                                ; 1). WE RAN OUT OF TIME---ERROR !!
4151                                ; 2). THE BIT IS IN THE CORRECT CONDITION--GOOD !!
4152
4153 011364                                IFB   FLAG EQ #TRUE THEN
4154 011364 126727 000016 000001      CMPB  FLAG,#TRUE
4155 011372 001001                    BNE   $217
4156 011374                                RETURN NO.ERROR   ; GOOD
4157 011374 000405                    BR    $210
4158 011376                                ENDF
4159 011376                                $217:
4160 011376                                RETURN ERROR      ; BAD
4161 011376 000261                    SEC
4162 011400 000404                    BR    $211
4163
4164 011402 000000                                REGSAV: .WORD 0
4165 011404 000000                                TIMSAV: .WORD 0
4166 011406 000                                FLAG:   .BYTE 0
4167 011407 000                                HOLDSC: .BYTE 0
4168                                ; WE ARE DONE GO BACK HOME
4169 011410                                ENDRTN
4170 011410                                $210:
4171 011410 000241                    CLC
4172 011412                                $211:
4173 011412 000207                    RTS   PC
    
```

```

4174
4175
4176 011414
4177 011414
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191 011414
4192 011414 005065 000000
4193 011420
4194 011420 016767 167574 000062
4195 011426 016746 000056
4196 011432 042716 000017
4197 011436 042667 000046
4198
4199 011442
4200 011442 012767 000001 167622
4201 011450 000402
4202 011452
4203 011452 005267 167614
4204 011456
4205 011456 026767 167610 000024
4206 011464 003006
4207 011466
4208 011466 006365 000000
4209 011472
4210 011472 052765 000001 000000
4211 011500
4212 011500 000764
4213 011502
4214 011502
4215 011502 005165 000000
4216 011506
4217 011506 000401
4218 011510 000000
4219 011512
4220 011512
4221 011512
4222 011512 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:
* SUSWR-- 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 SUSWR WORD.
*****

```

```

*****
LET MASK(R5) := #0 ; START
CLR MASK(R5)
LET NUMBR := SUSWR AND #DATA
MOV SUSWR, NUMBR
MOV NUMBR, -(SP)
BIC #DATA, (SP)
BIC (SP)+, NUMBR
INCR I FROM #1 TO NUMBR BY #1
MOV #1, I
BR $222
$223: INC I
$222: CMP I, NUMBR
BGT $224
LET MASK(R5) := MASK(R5) SHIFT #1
LET MASK(R5) := MASK(R5) SET.BY #1
ENDINC
BR $223
$224: LET MASK(R5) := COMP MASK(R5)
COM MASK(R5)
RETURN
BR $220
NUMBR: 0
ENDRTN
$220:
$221:
RTS PC

```

```

4223
4224
4225 011514
4226 011514
4227
4228
4229
4230
4231
4232
4233
4234 011514 010146
4235 011516 010246
4236 011520 010346
4237 011522
4238 011522 016501 000000
4239 011526
4240 011526 012702 000001
4241 011532 000402
4242 011534
4243 011534 062702 000001
4244 011540
4245 011540 020201
4246 011542 101010
4247 011544
4248 011544 005003
4249 011546 000401
4250 011550
4251 011550 005203
4252 011552
4253 011552 020327 000700
4254 011556 003001
4255 011560
4256 011560 000773
4257 011562
4258 011562
4259 011562 000764
4260 011564
4261 011564 012603
4262 011566 012602
4263 011570 012601
4264 011572
4265 011572
4266 011572
4267 011572 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 MICRO 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
$230:
ADD #01,R2
$227:
CMP R2,R1
BHI $231
INCR R3 FROM #0 TO #700 BY #1
CLR R3
BR $232
$233:
INC R3
$232:
CMP R3,#700
BGT $234
ENDINC
BR $233
$234:
ENDINC
BR $230
$231:
MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
ENDRTN
$225:
$226:
RTS PC

```

4268  
4269  
4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281  
4282  
4283  
4284

011574

011574

005267 000002

011600

000002

011602

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 ;THAT'S ALL
RTI
INTFLAG: 0

```



4324 011766  
 4325 011766  
 4326  
 4327  
 4328  
 4329  
 4330  
 4331  
 4332  
 4333  
 4334 011766  
 4335 011766  
 4336 011766  
 4337 011766 005767 000122  
 4338 011772 001027  
 4339 011774  
 4340 011774 026727 000116 000001  
 4341 012002 001003  
 4342 012004  
 4343 012004 005067 000106  
 4344 012010  
 4345 012010 000403  
 4346 012012  
 4347 012012  
 4348 012012 004767 000110  
 4349  
 4350 012016  
 4351 012016  
 4352 012016 012600  
 4353 012020  
 4354 012020  
 4355 012020  
 4356 012020 012767 000001 000066  
 4357 012026  
 4358 012026 012767 000001 167150  
 4359 012034  
 4360 012034 016767 167210 000056  
 4361 012042  
 4362 012042 016767 167176 000052  
 4363 012050  
 4364 012050 000410  
 4365 012052  
 4366 012052  
 4367 012052 012704 000010  
 4368 012056  
 4369 012056 006167 000032  
 4370 012062  
 4371 012062 060467 000032  
 4372 012066  
 4373 012066 060467 000030  
 4374 012072  
 4375 012072  
 4376 012072  
 4377 012072 036767 000016 167152  
 4378 012100 001732  
 4379

ROUTINE CYCLE

CYCLE:

```

*****
* 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.
*****

```

REPEAT

\$241:

IF BITMASK EQ #0 THEN

```

TST   BITMASK
BNE   $242

```

IF INITFLAG EQ #1 THEN

```

CMP   INITFLAG, #1
BNE   $243

```

LET INITFLAG := #0

CLR INITFLAG

ELSE

BR \$244

\$243:

CALL \$EOP ; AS A SUBROUTINE

JSR PC, \$EOP

SPECIALADDRESS:

```

; BECAUSE $EOP RETURNS AS A JUMP
LET RO := POP

```

MOV (SP)+, RO

ENDIF

\$244:

LET BITMASK := #1

MOV #1, BITMASK

LET \$DEVCT := #1

MOV #1, \$DEVCT

LET ADDRESS := \$BASE

MOV \$BASE, ADDRESS

LET VECTOR := \$VECT1

MOV \$VECT1, VECTOR

ELSE

BR \$245

\$242:

LET R4 := #10

MOV #10, R4

LET BITMASK := BITMASK ROTATE 1

ROL BITMASK

LET ADDRESS := ADDRESS + R4

ADD R4, ADDRESS

LET VECTOR := VECTOR + R4

ADD R4, VECTOR

ENDIF

\$245:

UNTIL BITMASK SETIN \$DEVCT

```

BIT   BITMASK, $DEVCT
BEQ   $241

```





```

4397
4398
4399
4400
4401
4402
4403
4404
4405
4406 012126
4407 012126 000004
4408 012130 005067 166746
4409 012134 005067 167020
4410 012140 005267 167036
4411 012144 042767 100000 167030
4412 012152 005327
4413 012154 000001
4414 012156 003022
4415 012160 012737
4416 012162 000001
4417 012164 012154
4418 012166 104401 012233
4419 012172 016746 167004
4420 012176 104405
4421 012200 104401 012230
4422 012204 013700 000042
4423 012210 001405
4424 012212 000005
4425 012214 004710
4426 012216 000240
4427 012220 000240
4428 012222 000240
4429 012224
4430 012224 000137
4431 012226 012016
4432 012230 377 377 000
4433 012233 015 042412 042116
4434 012240 050040 051501 020123
4435 012246 000043

```

.SBTTL END OF PASS ROUTINE

```

;*****
;INCREMENT THE PASS NUMBER ($PASS)
;INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
;TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
;IF THERES A MONITOR GO TO IT
;IF THERE ISN T JUMP TO SPECIALADDRESS

```

```

$EOP:
SCOPE
CLR $TSTNM ;ZERO THE TEST NUMBER
CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
INC $PASS ;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;LOOP?

$EOPCT: .WORD 1
BGT $DOAGN ;YES
MOV (PC)+,2(PC)+ ;RESTORE COUNTER

$ENDCT: .WORD 1
$EOPCT
TYPE $ENDMG ;TYPE "END PASS #"
MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
TYPDS ;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE $ENULL ;TYPE A NULL CHARACTER
$GET42: MOV #42,R0 ;GET MONITOR ADDRESS
BEQ $DOAGN ;BRANCH IF NO MONITOR
RESET ;CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;GO TO MONITOR
NOP ;SAVE ROOM
NOP ;FOR
NOP ;ACT11

$DOAGN: JMP 2(PC)+ ;RETURN
$RTNAD: .WORD SPECIALADDRESS
$ENULL: .BYTE -1,-1,0 ;NULL CHARACTER STRING
$ENDMG: .ASCIZ '<15><12>/END PASS #/'

```

```

4436 .SBTTL POWER DOWN AND UP ROUTINES
4437
4438
4439
4440 012250 012737 012410 000024 $PWRDN: MOV $SILLUP, @PWRVEC ;; SET FOR FAST UP
4441 012256 012737 000340 000026 MOV @340, @PWRVEC+2 ;; Prio:7
4442 012264 010046 MOV R0, -(SP) ;; PUSH R0 ON STACK
4443 012266 010146 MOV R1, -(SP) ;; PUSH R1 ON STACK
4444 012270 010246 MOV R2, -(SP) ;; PUSH R2 ON STACK
4445 012272 010346 MOV R3, -(SP) ;; PUSH R3 ON STACK
4446 012274 010446 MOV R4, -(SP) ;; PUSH R4 ON STACK
4447 012276 010546 MOV R5, -(SP) ;; PUSH R5 ON STACK
4448 012300 017746 166634 MOV @SWR, -(SP) ;; PUSH @SWR ON STACK
4449 012304 010667 000104 MOV SP, $SAVR6 ;; SAVE SP
4450 012310 012737 012322 000024 MOV $PWRUP, @PWRVEC ;; SET UP VECTOR
4451 012316 000000 HALT
4452 012320 000776 BR -2 ;; HANG UP
4453
4454
4455
4456 012322 012737 012410 000024 $PWRUP: MOV $SILLUP, @PWRVEC ;; SET FOR FAST DOWN
4457 012330 016706 000060 MOV $SAVR6, SP ;; GET SP
4458 012334 005067 000054 CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
4459 012340 005267 000050 1$: INC $SAVR6 ;; WAIT FOR THE INC
4460 012344 001375 BNE 1$ ;; OF WORD
4461 012346 012677 166566 MOV (SP)+, @SWR ;; POP STACK INTO @SWR
4462 012352 012605 MOV (SP)+, R5 ;; POP STACK INTO R5
4463 012354 012604 MOV (SP)+, R4 ;; POP STACK INTO R4
4464 012356 012603 MOV (SP)+, R3 ;; POP STACK INTO R3
4465 012360 012602 MOV (SP)+, R2 ;; POP STACK INTO R2
4466 012362 012601 MOV (SP)+, R1 ;; POP STACK INTO R1
4467 012364 012600 MOV (SP)+, R0 ;; POP STACK INTO R0
4468 012366 012737 012250 000024 MOV $PWRDN, @PWRVEC ;; SET UP THE POWER DOWN VECTOR
4469 012374 012737 000340 000026 MOV @340, @PWRVEC+2 ;; Prio:7
4470 012402 104401 TYPE REPORT THE POWER FAILURE
4471 012404 012416 $PWRMG: .WORD $POWER ;; POWER FAIL MESSAGE POINTER
4472 012406 000002 RTI
4473 012410 000000 $SILLUP: HALT ;; THE POWER UP SEQUENCE WAS STARTED
4474 012412 000776 BR -2 ;; BEFORE THE POWER DOWN WAS COMPLETE
4475 012414 000000 $SAVR6: 0 ;; PUT THE SP HERE
4476 012416 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
4477 012424 000122
4478 .EVEN

```

.SBTTL TYPE ROUTINE

4479  
4480  
4481  
4482  
4483  
4484  
4485  
4486  
4487  
4488  
4489  
4490  
4491  
4492  
4493  
4494  
4495  
4496  
4497  
4498  
4499  
4500  
4501  
4502  
4503  
4504  
4505  
4506  
4507  
4508  
4509  
4510  
4511  
4512  
4513  
4514  
4515  
4516  
4517  
4518  
4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534

012426 105767 166525  
012432 100002  
012434 000000  
012436 000430  
012440 010046  
012442 017600 000002  
012446 122767 000001 166540  
012454 0C1011  
012456 132767 000100 166531  
012464 001405  
012466 010067 000004  
012472 004767 000774  
012476 000000  
012500 132767 000040 166507  
012506 001003  
012510 112046  
012512 001005  
012514 005726  
012516 012600  
012520 062716 000002  
012524 000002  
012526 122716 000011  
012532 001430  
012534 122716 000200  
012540 001006  
012542 005726  
012544 104401  
012546 001171  
012550 105067 000130  
012554 000755  
012556 004767 000056  
012562 126726 166370  
012566 001350  
012570 016746 166360  
012574 105366 000001  
012600 002770  
012602 004767 000032  
012606 105367 000072

```
*****  
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.  
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.  
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.  
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.  
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.  
*  
*CALL:  
*1) USING A TRAP INSTRUCTION  
* TYPE ,MESADR ; ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING  
*OR  
* TYPE  
* MESADR  
*  
$TYPE: TSTB $TFPLG ; IS THERE A TERMINAL?  
BPL 1$ ; BR IF YES  
HALT ; HALT HERE IF NO TERMINAL  
BR 3$ ; LEAVE  
1$: MOV RO, -(SP) ; SAVE RO  
MOV 22(SP), RO ; GET ADDRESS OF ASCIZ STRING  
CMPB #APTENV, $ENV ; RUNNING IN APT MODE  
BNE 62$ ; NO GO CHECK FOR APT CONSOLE  
BITB #APTSPool, $ENVM ; SPOOL MESSAGE TO APT  
BEQ 62$ ; NO GO CHECK FOR CONSOLE  
MOV RO, 61$ ; SETUP MESSAGE ADDRESS FOR APT  
JSR PC, $ATY3 ; SPOOL MESSAGE TO APT  
61$: .WORD 0 ; MESSAGE ADDRESS  
62$: BITB #APTCSUP, $ENVM ; APT CONSOLE SUPPRESSED  
BNE 60$ ; YES, SKIP TYPE OUT  
2$: MOVB (RO)+, -(SP) ; PUSH CHARACTER TO BE TYPED ONTO STACK  
BNE 4$ ; BR IF IT ISN'T THE TERMINATOR  
TST (SP)+ ; IF TERMINATOR POP IT OFF THE STACK  
60$: MOV (SP)+, RO ; RESTORE RO  
3$: ADD #2, (SP) ; ADJUST RETURN PC  
RTI ; RETURN  
4$: CMPB #HT, (SP) ; BRANCH IF <HT>  
BEQ 8$ ;  
CMPB #CRLF, (SP) ; ; BRANCH IF NOT <CRLF>  
BNE 5$ ;  
TST (SP)+ ; ; POP <CR><LF> EQUIV  
TYPE ; ; TYPE A CR AND LF  
5$: CLRB $CHARCNT ; CLEAR CHARACTER COUNT  
BR 2$ ; GET NEXT CHARACTER  
6$: JSR PC, $TYPEC ; GO TYPE THIS CHARACTER  
CMPB $FILLC, (SP)+ ; IS IT TIME FOR FILLER CHARS.?  
BNE 2$ ; IF NO GO GET NEXT CHAR.  
MOV $NULL, -(SP) ; GET # OF FILLER CHARS. NEEDED  
 ; AND THE NULL CHAR.  
7$: DECB 1($F) ; DOES A NULL NEED TO BE TYPED?  
BLT 6$ ; BR IF NO--GO POP THE NULL OFF OF STACK  
JSR PC, $TYPEC ; GO TYPE A NULL  
DECB $CHARCNT ; DO NOT COUNT AS A COUNT
```

```

4535 012612 000770          BR      7$          ;;LOOP
4536
4537          ;HORIZONTAL TAB PROCESSOR
4538
4539 012614 112716 000040      8$:      MOVB   #' (SP)          ;; REPLACE TAB WITH SPACE
4540 012620 004767 000014      9$:      JSR    PC,$TYPEC          ;; TYPE A SPACE
4541 012624 132767 000007 000052      BITB   #7,$CHARCNT          ;; BRANCH IF NOT AT
4542 012632 001372          BNE    9$          ;; TAB STOP
4543 012634 005726          TST    (SP)+          ;; POP SPACE OFF STACK
4544 012636 000724          BR     2$          ;; GET NEXT CHARACTER
4545 012640 105777 166304      $TYPEC: TSTB   @2$PS          ;; WAIT UNTIL PRINTER IS READY
4546 012644 100375          BPL   $TYPEC
4547 012646 116677 000002 166276      MOVB   2(SP),@2$PB          ;; LOAD CHAR TO BE TYPED INTO DATA REG.
4548 012654 122766 000015 000002      CMPB   #CR,2(SP)          ;; IS CHARACTER A CARRIAGE RETURN?
4549 012662 001003          BNE    1$          ;; BRANCH IF NO
4550 012664 105067 000014          CLRB   $CHARCNT          ;; YES--CLEAR CHARACTER COUNT
4551 012670 000406          BR     $TYPEX          ;; EXIT
4552 012672 122766 000012 000002 1$:      CMPB   #LF,2(SP)          ;; IS CHARACTER A LINE FEED?
4553 012700 001402          BEQ   $TYPEX          ;; BRANCH IF YES
4554 012702 105227          INCB   (PC)+          ;; COUNT THE CHARACTER
4555 012704 000000      $CHARCNT: .WORD 0          ;; CHARACTER COUNT STORAGE
4556 012706 000207      $TYPEX: RTS    PC
4557

```

```

4558 .SBTTL TTY INPUT ROUTINE
4559
4560 ;:*****
4561 .ENABL LSB
4562
4563 ;:*****
4564 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4565 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
4566 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
4567 ;*WHEN OPERATING IN TTY FLAG MODE.
4568 012710 022767 000176 166222 $CKSWR: CMP #SWREG,SWR ;: IS THE SOFT-SWR SELECTED?
4569 012716 001074 BNE 15$ ;: BRANCH IF NO
4570 012720 105777 166220 TSTB @STKS ;: CHAR THERE?
4571 012724 100071 BPL 15$ ;: IF NO, DON'T WAIT AROUND
4572 012726 117746 166214 MOVB @STKB,-(SP) ;: SAVE THE CHAR
4573 012732 042716 177600 BIC #↑C177,(SP) ;: STRIP-OFF THE ASCII
4574 012736 022726 000007 CMP #7,(SP)+ ;: IS IT A CONTROL G?
4575 012742 001062 BNE 15$ ;: NO, RETURN TO USER
4576 012744 126727 166164 000001 CMPB $AUTOB,#1 ;: ARE WE RUNNING IN AUTO-MODE?
4577 012752 001456 BEQ 15$ ;: BRANCH IF YES
4578
4579 012754 104401 013435 SGTSWR: TYPE ,SCNTLG ;: ECHO THE CONTROL-G (↑G)
4580 012760 104401 013442 TYPE ,SMSWR ;: TYPE CURRENT CONTENTS
4581 012764 016746 165206 MOV SWREG,-(SP) ;: SAVE SWREG FOR TYPEOUT
4582 012770 104402 TYPOC ;: GO TYPE--OCTAL ASCII(ALL DIGITS)
4583 012772 104401 013453 TYPE ,SMNEW ;: PROMPT FOR NEW SWR
4584 012776 005046 19$: CLR -(SP) ;: CLEAR COUNTER
4585 013000 005046 CLR -(SP) ;: THE NEW SWR
4586 013002 105777 166136 7$: TSTB @STKS ;: CHAR THERE?
4587 013006 100375 BPL 7$ ;: IF NOT TRY AGAIN
4588
4589 013010 117746 166132 MOVB @STKB,-(SP) ;: PICK UP CHAR
4590 013014 042716 177600 BIC #↑C177,(SP) ;: MAKE IT 7-BIT ASCII
4591
4592
4593
4594 013020 021627 000025 9$: CMP (SP),#25 ;: IS IT A CONTROL-U?
4595 013024 001005 BNE 10$ ;: BRANCH IF NOT
4596 013026 104401 013430 TYPE ,SCNTLU ;: YES, ECHO CONTROL-U (↑U)
4597 013032 062706 000006 20$: ADP #6,SP ;: IGNORE PREVIOUS INPUT
4598 013036 000757 BR 19$ ;: LET'S TRY IT AGAIN
4599
4600
4601 013040 021627 000015 10$: CMP (SP),#15 ;: IS IT A <CR>?
4602 013044 001022 BNE 16$ ;: BRANCH IF NO
4603 013046 005766 000004 TST 4(SP) ;: YES, IS IT THE FIRST CHAR?
4604 013052 001403 BEQ 11$ ;: BRANCH IF YES
4605 013054 016677 000002 166056 MOV 2(SP),@SWR ;: SAVE NEW SWR
4606 013062 062706 000006 11$: ADD #6,SP ;: CLEAR UP STACK
4607 013066 104401 001171 14$: TYPE ,SCRLF ;: ECHO <CR> AND <LF>
4608 013072 126727 166037 000001 CMPB $INTAG,#1 ;: RE-ENABLE TTY KBD INTERRUPTS?
4609 013100 001003 BNE 15$ ;: BRANCH IF NOT
4610 013102 012777 000100 166034 MOV #100,@STKS ;: RE-ENABLE TTY KBD INTERRUPTS
4611 013110 000002 15$: RTI ;: RETURN
4612 013112 004767 177522 16$: JSR PC,$TYPEC ;: ECHO CHAR
4613 013116 021627 000060 CMP (SP),#60 ;: CHAR < 0?

```

```

4614 013122 002420          BLT      18$          ;; BRANCH IF YES
4615 013124 021627 000067  CMP      (SP),#67    ;; CHAR > 7?
4616 013130 003015          BGT      18$          ;; BRANCH IF YES
4617 013132 042726 000060  BIC      #60,(SP)+   ;; STRIP-OFF ASCII
4618 013136 005766 000002  TST      2(SP)       ;; IS THIS THE FIRST CHAR
4619 013142 001403          BEQ      17$          ;; BRANCH IF YES
4620 013144 006316          ASL      (SP)        ;; NO, SHIFT PRESENT
4621 013146 006316          ASL      (SP)        ;; CHAR OVER TO MAKE
4622 013150 006316          ASL      (SP)        ;; ROOM FOR NEW ONE.
4623 013152 005266 000002  17$: INC      2(SP)    ;; KEEP COUNT OF CHAR
4624 013156 056616 177776  BIS      -2(SP),(SP) ;; SET IN NEW CHAR
4625 013162 000707          BR       7$          ;; GET THE NEXT ONE
4626 013164 104401 001170  18$: TYPE   $QUES    ;; TYPE ?(CR)(LF)
4627 013170 000720          BR      20$          ;; SIMULATE CONTROL-U
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639 013172 011646          $RDCHR: MOV      (SP),-(SP) ;; PUSH DOWN THE PC
4640 013174 016666 000004 000002  MOV      4(SP),2(SP) ;; SAVE THE PS
4641 013202 105777 165736          1$: TSTB   2$TKS    ;; WAIT FOR
4642 013206 100375          BPL      1$          ;; A CHARACTER
4643 013210 117766 165732 000004  MOVB    2$TKB,4(SP) ;; READ THE TTY
4644 013216 042766 177600 000004  BIC      #1C(177),4(SP) ;; GET RID OF JUNK IF ANY
4645 013224 026627 000004 000023  CMP      4(SP),#23   ;; IS IT A CONTROL-S?
4646 013232 001013          BNE      3$          ;; BRANCH IF NO
4647 013234 105777 165704          2$: TSTB   2$TKS    ;; WAIT FOR A CHARACTER
4648 013240 100375          BPL      2$          ;; LOOP UNTIL ITS THERE
4649 013242 117746 165700          MOVB    2$TKB,-(SP) ;; GET CHARACTER
4650 013246 042716 177600          BIC      #1C177,(SP) ;; MAKE IT 7-BIT ASCII
4651 013252 022627 000021          CMP      (SP)+,#21   ;; IS IT A CONTROL-Q?
4652 013256 001366          BNE      2$          ;; IF NOT DISCARD IT
4653 013260 000750          BR       1$          ;; YES, RESUME
4654 013262 026627 000004 000140  3$: CMP      4(SP),#140 ;; IS IT UPPER CASE?
4655 013270 002407          BLT      4$          ;; BRANCH IF YES
4656 013272 026627 000004 000175  CMP      4(SP),#175  ;; IS IT A SPECIAL CHAR?
4657 013300 003003          BGT      4$          ;; BRANCH IF YES
4658 013302 042766 000040 000004  BIC      #40,4(SP)   ;; MAKE IT UPPER CASE
4659 013310 000002          4$: RTI              ;; GO BACK TO USER
4660
4661
4662
4663
4664
4665
4666
4667 013312 010346          $RDLIN: MOV      R3,-(SP) ;; SAVE R3
4668 013314 012703 013420          1$: MOV      #1TTYIN,R3 ;; GET ADDRESS
4669 013320 022703 013430          2$: CMP      #1TTYIN+8.,R3 ;; BUFFER FULL?

```

\*\*\*\*\*

THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY

\*CALL:

\* RDCHR INPUT A SINGLE CHARACTER FROM THE TTY  
\* RETURN HERE CHARACTER IS ON THE STACK  
\* WITH PARITY BIT STRIPPED OFF

\*\*\*\*\*

THIS ROUTINE WILL INPUT A STRING FROM THE TTY

\*CALL:

\* RDLIN INPUT A STRING FROM THE TTY  
\* RETURN HERE ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK  
\* TERMINATOR WILL BE A BYTE OF ALL 0'S

4670	013324	101405				BLOS	45		:: BR IF YES
4671	013326	104410				RDCHR			:: GO READ ONE CHARACTER FROM THE TTY
4672	013330	112613				MOVB	(SP)+, (R3)		:: GET CHARACTER
4673	013332	122713	000177		105:	CMPB	#177, (R3)		:: IS IT A RUBOUT
4674	013336	001003				BNE	35		:: SKIP IF NOT
4675	013340	104401	001170		45:	TYPE	\$QUES		:: TYPE A '?'
4676	013344	000763				BR	15		:: CLEAR THE BUFFER AND LOOP
4677	013346	111367	000044		35:	MOVB	(R3), 95		:: ECHO THE CHARACTER
4678	013352	104401	013416			TYPE	95		
4679	013356	122723	000015			CMPB	#15, (R3)+		:: CHECK FOR RETURN
4680	013362	001356				BNE	25		:: LOOP IF NOT RETURN
4681	013364	105063	177777			CLRB	-1(R3)		:: CLEAR RETURN (THE 15)
4682	013370	104401	001172			TYPE	\$LF		:: TYPE A LINE FEED
4683	013374	012603				MOV	(SP)+, R3		:: RESTORE R3
4684	013376	011646				MOV	(SP), -(SP)		:: ADJUST THE STACK AND PUT ADDRESS OF THE
4685	013400	016666	000004	000002		MOV	4(SP), 2(SP)		:: FIRST ASCII CHARACTER ON IT
4686	013406	012766	013420	000004		MOV	#STTYIN, 4(SP)		
4687	013414	000002				RTI			:: RETURN
4688	013416	000			95:	.BYTE	0		:: STORAGE FOR ASCII CHAR. TO TYPE
4689	013417	000				.BYTE	0		:: TERMINATOR
4690	013420	000010				\$TTYIN:	.BLKB	8.	:: RESERVE 8 BYTES FOR TTY INPUT
4691	013430	052536	005015	000		\$CNTLU:	.ASCIZ	/↑U/<15><12>	:: CONTROL "U"
4692	013435	136	006507	000012		\$CNTLG:	.ASCIZ	/↑G/<15><12>	:: CONTROL "G"
4693	013442	005015	053523	020122		\$MSWR:	.ASCIZ	<15><12>/SWR = /	
4694	013450	020075	000						
4695	013453	040	047040	053505		\$MNEW:	.ASCIZ	/ NEW = /	
4696	013460	036440	000040						

4697  
4698  
4699  
4700  
4701  
4702  
4703  
4704  
4705  
4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751  
4752

013464 112767 000001 000236  
013472 112767 000001 000226  
013500 000403  
013502 112767 000001 000220  
013510 010046  
013512 010146  
013514 105767 000206  
013520 001450  
013522 122767 000001 165464  
013530 001031  
013532 132767 000100 165455  
013540 001425  
013542 017600 000004  
013546 062766 000002 000004  
013554 005767 165414  
013560 001375  
013562 010067 165422  
013566 105720  
013570 001376  
013572 166700 165412  
013576 006200  
013600 010067 165406  
013604 012767 000004 165362  
013612 000413  
013614 017667 000004 000016  
013622 062766 000002 000004  
013630 016746 164142  
013634 004767 176566  
013640 000000  
013642  
013642 105767 000062  
013646 001416  
013650 005767 165340  
013654 001413  
013656 005767 165312  
013662 001375  
013664 017667 000004 165304  
013672 062766 000002 000004  
013700 005267 165270  
013704 105067 000020  
013710 105067 000013  
013714 105067 000006  
013720 012601  
013722 012600  
013724 000207  
013726 000  
013727 000  
013730 000  
013732  
000200  
000001  
000100

.SBTTL APT COMMUNICATIONS ROUTINE

```

*****
SATY1: MOVB #1, $FFLG ;; TO REPORT FATAL ERROR
SATY3: MOVB #1, $MFLG ;; TO TYPE A MESSAGE
BR SATYC
SATY4: MOVB #1, $FFLG ;; TO ONLY REPORT FATAL ERROR
SATYC:
MOV RO, -(SP) ;; PUSH RO ON STACK
MOV R1, -(SP) ;; PUSH R1 ON STACK
TSTB $MFLG ;; SHOULD TYPE A MESSAGE?
BEQ $S IF NOT: BR
CMPB #APTENV, $ENV ;; OPERATING UNDER APT?
BNE $S IF NOT: BR
BITB #APTSPool, $ENVM ;; SHOULD SPOOL MESSAGES?
BEQ $S IF NOT: BR
MOV #4(SP), RO ;; GET MESSAGE ADDR.
ADD #2, 4(SP) ;; BUMP RETURN ADDR.
1$: TST $MSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
BNE $S IF NOT: WAIT
MOV RO, $MSGAD ;; PUT ADDR IN MAILBOX
2$: TSTB (RO)+ ;; FIND END OF MESSAGE
BNE $S
SUB $MSGAD, RO ;; SUB START OF MESSAGE
ASR RO ;; GET MESSAGE LNTH IN WORDS
MOV RO, $MSGLGT ;; PUT LENGTH IN MAILBOX
MOV #4, $MSGTYPE ;; TELL APT TO TAKE MSG.
BR $S
3$: MOV #4(SP), 4$ ;; PUT MSG ADDR IN JSR LINKAGE
ADD #2, 4(SP) ;; BUMP RETURN ADDRESS
MOV 177776, -(SP) ;; PUSH 177776 ON STACK
JSR PC, $TYPE ;; CALL TYPE MACRO
4$: .WORD 0
5$:
10$: TSTB $FFLG ;; SHOULD REPORT FATAL ERROR?
BEQ $S IF NOT: BR
TST $ENV ;; RUNNING UNDER APT?
BEQ $S IF NOT: BR
TST $MSGTYPE ;; FINISHED LAST MESSAGE?
BNE $S IF NOT: WAIT
MOV #4(SP), $FATAL ;; GET ERROR #
ADD #2, 4(SP) ;; BUMP RETURN ADDR.
INC $MSGTYPE ;; TELL APT TO TAKE ERROR
12$: CLRB $FFLG ;; CLEAR FATAL FLAG
CLRB $LFLG ;; CLEAR LOG FLAG
CLRB $MFLG ;; CLEAR MESSAGE FLAG
MOV (SP)+, R1 ;; POP STACK INTO R1
MOV (SP)+, RO ;; POP STACK INTO RO
RTS PC ;; RETURN
$MFLG: .BYTE 0 ;; MESSG. FLAG
$LFLG: .BYTE 0 ;; LOG FLAG
$FFLG: .BYTE 0 ;; FATAL FLAG
.EVEN
APTSIZE=200
APTENV=001
APTSPool=100

```



H09

MRINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 113  
DVDVA.A.P11 06-MAY-77 15:29 APT COMMUNICATIONS ROUTINE

SEQ 0111

4753

000040

APTC SUP=040

4754  
4755  
4756  
4757  
4758  
4759  
4760  
4761  
4762  
4763  
4764  
4765  
4766  
4767  
4768  
4769  
4770  
4771  
4772  
4773  
4774  
4775  
4776  
4777  
4778  
4779  
4780  
4781  
4782  
4783  
4784  
4785  
4786  
4787  
4788  
4789  
4790  
4791  
4792  
4793  
4794  
4795  
4796  
4797  
4798  
4799  
4800  
4801  
4802  
4803  
4804  
4805  
4806  
4807

013732  
013732 104407  
013734 105267 165143  
013740 001775  
013742 016777 165134 165172  
013750 032777 002000 165162  
013756 001402  
013760 104401 001164  
013764 005267 165122  
013770 011667 165122  
013774 162767 000002 165114  
014002 117767 165110 155104  
014010 032777 020000 165122  
014016 001004  
014020 004767 175560  
014024 104401 001171  
014030  
014030 122767 000001 165156  
014036 001007  
014040 116767 165050 000004  
014046 004767 177430  
014052 000  
014053 000  
014054 000777  
014056 005777 165056  
014062 100002  
014064 000000  
014066 104407  
014070 032777 001000 165042  
014076 001402  
014100 016716 165004  
014104 005767 165052  
014110 001402  
014112 016716 165044  
014116  
014116 022737 012214 000042  
014124 001001  
014126 000000  
014130  
014130 000002

.SBTTL ERROR HANDLER ROUTINE

\*\*\*\*\*  
\*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,  
\*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL  
\*AND GO TO MYTYPE ON ERROR  
\*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
\*SW15=1 HALT ON ERROR  
\*SW13=1 INHIBIT ERROR TYPEOUTS  
\*SW10=1 BELL ON ERROR  
\*SW09=1 LOOP ON ERROR  
\*CALL  
\* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER

\$ERROR:

7\$: CKSWR ;;TEST FOR CHANGE IN SOFT-SWR  
INCB \$ERFLG ;;SET THE ERROR FLAG  
BEQ 7\$ ;;DON'T LET THE FLAG GO TO ZERO  
MOV \$STNM, \$DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG  
BIT #BIT10, \$SWR ;;BELL ON ERROR?  
BEQ 1\$ ;;NO - SKIP  
TYPE \$BELL ;;RING BELL  
1\$: INC \$ERTTL ;;COUNT THE NUMBER OF ERRORS  
MOV (SP), \$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION  
SUB #2, \$ERRPC  
MOVB \$ERRPC, \$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE  
BIT #BIT13, \$SWR ;;SKIP TYPEOUT IF SET  
BNE 20\$ ;;SKIP TYPEOUTS  
JSR PC, MYTYPE ;;GO TO USER ERROR ROUTINE  
TYPE \$CRLF  
20\$: CMPB #APTENV, \$ENV ;;RUNNING IN APT MODE  
BNE 2\$ ;;NO SKIP APT ERROR REPORT  
MOVB \$ITEMB, 21\$ ;;SET ITEM NUMBER AS ERROR NUMBER  
TSR PC, \$ATY4 ;;REPORT FATAL ERROR TO APT  
21\$: .BYTE 0  
.BYTE 0  
22\$: BR 22\$ ;;APT ERROR LOOP  
2\$: TST \$SWR ;;HALT ON ERROR  
BPL 3\$ ;;SKIP IF CONTINUE  
HALT ;;HALT ON ERROR!  
CKSWR ;;TEST FOR CHANGE IN SOFT-SWR  
3\$: BIT #BIT09, \$SWR ;;LOOP ON ERROR SWITCH SET?  
BEQ 4\$ ;;BR IF NO  
MOV \$LPERR, (SP) ;;FUDGE RETURN FOR LOOPING  
4\$: TST \$ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS  
BEQ 5\$ ;;BR IF NONE  
MOV \$ESCAPE, (SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE  
5\$: CMP #SENDAD, \$#42 ;;ACT-11 AUTO-ACCEPT?  
BNE 6\$ ;;BRANCH IF NO  
HALT ;;YES  
6\$: RTI ;;RETURN

4808  
4809  
4810  
4811  
4812  
4813  
4814  
4815  
4816  
4817  
4818  
4819  
4820  
4821  
4822  
4823  
4824  
4825  
4826  
4827  
4828  
4829  
4830  
4831  
4832  
4833  
4834  
4835  
4836  
4837  
4838  
4839  
4840  
4841  
4842  
4843  
4844  
4845  
4846  
4847  
4848  
4849  
4850  
4851  
4852  
4853  
4854  
4855  
4856  
4857  
4858  
4859  
4860  
4861  
4862  
4863

014132  
014132 104407  
014134 032777 040000 164776  
014142 001114  
014144 000416  
014146 013746 000004  
014152 012737 014172 000004  
014160 005737 177060  
014164 012637 000004  
014170 000463  
014172 022626  
014174 012637 000004  
014200 000423  
014 12  
014202 032777 000400 164730  
014210 001404  
014212 127767 164722 164662  
014220 001465  
014222 105767 164655  
014226 001421  
014230 126767 164661 164645  
014236 101015  
014240 032777 001000 164672  
014246 001404  
014250 016767 164634 164630  
014256 000446  
014260 105067 164617  
014264 005067 164670  
014270 000415  
014272 032777 004000 164640  
014300 001011  
014302 005767 164674  
014306 001406  
014310 005267 164570  
014314 026767 164640 164562  
014322 002024  
014324 012767 000001 164552  
014332 016767 000052 164620  
014340 105267 164536  
014344 116767 164532 164626

.SBTTL SCOPE HANDLER ROUTINE

```
*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW11=1      INHIBIT ITERATIONS
*SW09=1      LOOP ON ERROR
*SW08=1      LOOP ON TEST IN SWR<7:0>
*CALL
*          SCOPE          ;;SCOPE=IOT
```

```
SSCOPE:
1$:      CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
        BIT          #BIT14,$SWR      ;;LOOP ON PRESENT TEST?
        BNE          $OVER          ;;YES IF SW14=1
        *****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR          6$
        MOV          2#ERRVEC, -(SP)  ;;IF RUNNING ON THE "XOR" TESTER CHANGE
        MOV          #55, 2#ERRVEC  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
        TST          2#177060      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
        MOV          (SP)+, 2#ERRVEC  ;;SET FOR TIMEOUT
        BR          $SVLAD          ;;TIME OUT ON XOR?
        CMP          (SP)+, (SP)+    ;;RESTORE THE ERROR VECTOR
        MOV          (SP)+, 2#ERRVEC  ;;GO TO THE NEXT TEST
        BR          7$              ;;CLEAR THE STACK AFTER A TIME OUT
        *****END OF CODE FOR THE XOR TESTER*****
        BIT          #BIT08,$SWR      ;;LOOP ON SPEC. TEST?
        BEQ          2$              ;;BR IF NO
        CMPB         2#SWR,$STNM      ;;ON THE RIGHT TEST? SWR<7:0>
        BEQ          $OVER          ;;BR IF YES
        TSTB         $ERFLG          ;;HAS AN ERROR OCCURRED?
        BEQ          3$              ;;BR IF NO
        CMPB         $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
        BHI          3$              ;;BR IF NO
        BIT          #BIT09,$SWR      ;;LOOP ON ERROR?
        BEQ          4$              ;;BR IF NO
        MOV          $LPERR,$LPADR    ;;SET LOOP ADDRESS TO LAST SCOPE
        BR          $OVER
        CLRB         $ERFLG          ;;ZERO THE ERROR FLAG
        CLR          $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
        BR          1$              ;;ESCAPE TO THE NEXT TEST
        BIT          #BIT11,$SWR      ;;INHIBIT ITERATIONS?
        BNE          1$              ;;BR IF YES
        TST          $PASS          ;;IF FIRST PASS OF PROGRAM
        BEQ          1$              ;;INHIBIT ITERATIONS
        INC          $ICNT          ;;INCREMENT ITERATION COUNT
        CMP          $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
        BGE          $OVER          ;;BR IF MORE ITERATION REQUIRED
        MOV          #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
        MOV          $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
        INCB         $STNM          ;;COUNT TEST NUMBERS
        MOVB        $STNM,$TESTN    ;;SET TEST NUMBER IN APT MAILBOX
```

K09

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 116  
DVDVA.A.P11 06-MAY-77 15:29 SCOPE HANDLER ROUTINE

SEG 0114

4864	014352	011667	164530		MOV	(SP), \$LPADR	:: SAVE SCOPE LOOP ADDRESS
4865	014356	011667	164526		MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS
4866	014362	005067	164574		CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
4867	014366	112767	000001	164521	MOVB	#1, \$SERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4868	014374	016777	164502	164540	\$OVER: MOV	\$STNM, @DISPLAY	:: DISPLAY TEST NUMBER
4869	014402	016716	164500		MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS
4870	014406	000002			RTI		:: FIXES PS
4871	014410	003720			\$MXCNT: 2000.		:: MAX. NUMBER OF ITERATIONS

4872  
4873  
4874  
4875  
4876  
4877  
4878  
4879  
4880  
4881  
4882  
4883  
4884 014412  
4885 014412 010046  
4886 014414 010146  
4887 014416 010246  
4888 014420 010346  
4889 014422 010546  
4890 014424 012746 020200  
4891 014430 016605 000020  
4892 014434 100004  
4893 014436 005405  
4894 014440 112766 000055 000001  
4895 014446 005000  
4896 014450 012703 014626  
4897 014454 112723 000040  
4898 014460 005002  
4899 014462 016001 014616  
4900 014466 160105  
4901 014470 002402  
4902 014472 005202  
4903 014474 000774  
4904 014476 060105  
4905 014500 005702  
4906 014502 001002  
4907 014504 105716  
4908 014506 100407  
4909 014510 106316  
4910 014512 103003  
4911 014514 116663 000001 177777  
4912 014522 052702 000060  
4913 014526 052702 000040  
4914 014532 110223  
4915 014534 005720  
4916 014536 020027 000010  
4917 014542 002746  
4918 014544 003002  
4919 014546 010502  
4920 014550 000764  
4921 014552 105726  
4922 014554 100003  
4923 014556 116663 177777 177776  
4924 014564 105013  
4925 014566 012605  
4926 014570 012603  
4927 014572 012602

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

\*\*\*\*\*  
\*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.  
1\$: CLR R0 ;:ZERO THE CONSTANTS INDEX  
MOV #5DBLK,R3 ;:SETUP THE OUTPUT POINTER  
MOVB #' ,(R3)+ ;:SET THE FIRST CHARACTER TO A BLANK  
2\$: CLR R2 ;:CLEAR THE BCD NUMBER  
MOV \$DTBL(R0),R1 ;:GET THE CONSTANT  
3\$: SUB R1,R5 ;:FORM THIS BCD DIGIT  
BLT 4\$ ;:BR IF DONE  
INC R2 ;:INCREASE THE BCD DIGIT BY 1  
BR 3\$  
4\$: 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  
5\$: ASLB (SP) ;:MSD?  
BCC 6\$ ;:BR IF NO  
MOVB 1(SP),-1(R3) ;:YES--SET THE SIGN  
6\$: BIS #'0,R2 ;:MAKE THE BCD DIGIT ASCII  
7\$: 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  
8\$: TSTB (SP)+ ;:WAS THE LSD THE FIRST NON-ZERO?  
9\$: BPL 9\$ ;:BR IF NO  
MOVB -1(SP),-2(R3) ;:YES--SET THE SIGN FOR TYPING  
9\$: 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

```

4928 014574 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
4929 014576 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
4930 014600 104401      TYPE      $DBLK      ;;NOW TYPE THE NUMBER
4931 014604 016666 014626 000002 000004      MOV      2(SP),4(SP)  ;;ADJUST THE STACK
4932 014612 012616      MOV      (SP)+,(SP)
4933 014614 000002      RTI
4934 014616 023420      $DTBL: 10000.      ;;RETURN TO USER
4935 014620 001750      1000.
4936 014622 000144      100.
4937 014624 000012      10.
4938 014626 000004      $DBLK: .BLKW 4

```

```

4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964 014636 017646 000000
4965 014642 116667 000001 000211
4966 014650 112667 000207
4967 014654 062716 000002
4968 014660 000406
4969 014662 112767 000001 000171
4970 014670 112757 000006 000165
4971 014676 112767 000005 000154
4972 014704 010346
4973 014706 010446
4974 014710 010546
4975 014712 116704 000145
4976 014716 005404
4977 014720 062704 000006
4978 014724 110467 000132
4979 014730 116704 000125
4980 014734 016605 000012
4981 014740 005003
4982 014742 006105 1$:
4983 014744 000404
4984 014746 006105 2$:
4985 014750 006105
4986 014752 006105
4987 014754 010503
4988 014756 006103 3$:
4989 014760 105367 000076
4990 014764 100016
4991 014766 042703 177770
4992 014772 001002
4993 014774 005704
4994 014776 001403

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS    ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*STYPOS OR STYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    ;;CALL FOR TYPEOUT
*STYPOS: MOV      2(SP),-(SP)    ;;PICKUP THE MODE
*         MOVB   1(SP),SOFILL    ;;LOAD ZERO FILL SWITCH
*         MOVB   (SP)+,SOMODE+1  ;;NUMBER OF DIGITS TO TYPE
*         ADD    #2,(SP)        ;;ADJUST RETURN ADDRESS
*         BR     STYPON
*STYPOC: MOVB   #1,SOFILL        ;;SET THE ZERO FILL SWITCH
*         MOVB   #6,SOMODE+1     ;;SET FOR SIX(6) DIGITS
*STYPON: MOVB   #5,SOCNT         ;;SET THE ITERATION COUNT
*         MOV    R3,-(SP)        ;;SAVE R3
*         MOV    R4,-(SP)        ;;SAVE R4
*         MOV    R5,-(SP)        ;;SAVE R5
*         MOVB   SOMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
*         NEG    R4
*         ADD    #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
*         MOVB   R4,SOMODE       ;;SAVE IT FOR USE
*         MOVB   SOFILL,R4       ;;GET THE ZERO FILL SWITCH
*         MOV    12(SP),R5       ;;PICKUP THE INPUT NUMBER
*         CLR    R3              ;;CLEAR THE OUTPUT WORD
*         ROL   R5               ;;ROTATE MSB INTO "C"
*         BR    3$              ;;GO DO MSB
*         ROL   R5               ;;FORM THIS DIGIT
*         ROL   R5
*         ROL   R5
*         MOV   R5,R3
*         ROL   R3               ;;GET LSB OF THIS DIGIT
*         DECB  SOMODE           ;;TYPE THIS DIGIT?
*         BR   NO                ;;BR IF NO
*         BIC  #177770,R3       ;;GET RID OF JUNK
*         BNE  4$               ;;TEST FOR 0
*         TST  R4               ;;SUPPRESS THIS 0?
*         BEQ  5$               ;;BR IF YES

```

B10

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 120  
DVDVA.P11 06-MAY-77 15:29 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0118

4995	015000	005204		4\$:	INC	R4	:: DON'T SUPPRESS ANYMORE 0'S
4996	015002	052703	000060		BIS	#'0,R3	:: MAKE THIS DIGIT ASCII
4997	015006	052703	000040	5\$:	BIS	#' ,R3	:: MAKE ASCII IF NOT ALREADY
4998	015012	110367	000040		MOV	R3,8\$	:: SAVE FOR TYPING
4999	015016	104401	015056		TYPE	8\$	:: GO TYPE THIS DIGIT
5000	015022	105367	000032	7\$:	DECB	\$OCNT	:: COUNT BY 1
5001	015026	003347			BGT	2\$	:: BR IF MORE TO DO
5002	015030	002402			BLT	6\$	:: BR IF DONE
5003	015032	005204			INC	R4	:: INSURE LAST DIGIT ISN'T A BLANK
5004	015034	000744			BR	2\$	:: GO DO THE LAST DIGIT
5005	015036	012605		6\$:	MOV	(SP)+,R5	:: RESTORE R5
5006	015040	012604			MOV	(SP)+,R4	:: RESTORE R4
5007	015042	012603			MOV	(SP)+,R3	:: RESTORE R3
5008	015044	016666	000002 000004		MOV	2(SP),4(SP)	:: SET THE STACK FOR RETURNING
5009	015052	012616			MOV	(SP)+,(SP)	
5010	015054	000002			RTI		:: RETURN
5011	015056	000		8\$:	.BYTE	0	:: STORAGE FOR ASCII DIGIT
5012	015057	000			.BYTE	0	:: TERMINATOR FOR TYPE ROUTINE
5013	015060	000		\$OCNT:	.BYTE	0	:: OCTAL DIGIT COUNTER
5014	015061	000		\$OFILL:	.BYTE	0	:: ZERO FILL SWITCH
5015	015062	000000		\$OMODE:	.WORD	0	:: NUMBER OF DIGITS TO TYPE



5016  
5017  
5018  
5019  
5020  
5021  
5022  
5023  
5024 015064 010046  
5025 015066 016600 000002  
5026 015072 005740  
5027 015074 111000  
5028 015076 006300  
5029 015100 016000 015120  
5030 015104 000200  
5031  
5032  
5033  
5034  
5035 015106 011646  
5036 015110 016666 000004 000002  
5037 015116 000002  
5038  
5039  
5040  
5041  
5042  
5043  
5044  
5045  
5046 015120 015106  
5047 015122 012426  
5048 015124 014662  
5049 015126 014636  
5050 015130 014676  
5051 015132 014412  
5052  
5053 015134 012760  
5054  
5055 015136 012710  
5056 015140 013172  
5057 015142 013312  
5058 000001

.SBTTL TRAP DECODER

\*\*\*\*\*  
; THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION  
; AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS  
; OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL  
; GO TO THAT ROUTINE.

\$TRAP: MOV RO, -(SP) ;: SAVE RO  
MOV 2(SP), RO ;: GET TRAP ADDRESS  
TST -(RO) ;: BACKUP BY 2  
MOVB (RO), RO ;: GET RIGHT BYTE OF TRAP  
ASL RO ;: POSITION FOR INDEXING  
MOV \$TRPAD(RO), RO ;: INDEX TO TABLE  
RTS RO ;: GO TO ROUTINE

;; THIS IS USE TO HANDLE THE "GETPRI" MACRO

\$TRAP2: MOV (SP), -(SP) ;: MOVE THE PC DOWN  
MOV 4(SP), 2(SP) ;: MOVE THE PSW DOWN  
RTI ;: RESTORE THE PSW

.SBTTL TRAP TABLE

; THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
; BY THE "TRAP" INSTRUCTION.

ROUTINE  
-----  
\$TRPAD: .WORD \$TRAP2  
\$TYPE ;: CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE  
\$TYPOC ;: CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)  
\$TYPOS ;: CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)  
\$TYPON ;: CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)  
\$TYPDS ;: CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)  
\$GTSWR ;: CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING  
\$CKSWR ;: CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT SWR  
\$RDCHR ;: CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE  
\$RDLIN ;: CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE

.END

ABASE = 175610	3#	1008	1049	
ACDM1 = 000000	1008			
ACDM2 = 000000	1008			
ACPUOP = 000000	1008	1023		
ADDRES = 012120	4360*	4371*	4381	4388#
ADDMD = 000000	1008			
ADDW1 = 000000	1008			
ADDW10 = 000000	1008			
ADDW11 = 000000	1008			
ADDW12 = 000000	1008			
ADDW13 = 000000	1008			
ADDW14 = 000000	1008			
ADDW15 = 000000	1008			
ADDW2 = 000000	1008			
ADDW3 = 000000	1008			
ADDW4 = 000000	1008			
ADDW5 = 000000	1008			
ADDW6 = 000000	1008			
ADDW7 = 000000	1008			
ADDW8 = 000000	1008			
ADDW9 = 000300	1008			
ADEVCT = 000000	1008	1014		
ADEVN = 000001	3#	1008	1050	
AENV = 000000	1008	1019		
AENVN = 000000	1008	1020		
AFATAL = 000000	1008	1011		
AMADR1 = 000000	1008	1036		
AMADR2 = 000000	1008	1040		
AMADR3 = 000000	1008	1043		
AMADR4 = 000000	1008	1046		
AMAMS1 = 000000	1008	1030		
AMAMS2 = 000000	1008	1038		
AMAMS3 = 000000	1008	1041		
AMAMS4 = 000000	1008	1044		
AMSGAD = 000000	1008	1016		
AMSGLG = 000000	1008	1017		
AMSGTY = 000000	1008	1010		
AMTYP1 = 000000	1008	1031		
AMTYP2 = 000000	1008	1039		
AMTYP3 = 000000	1008	1042		
AMTYP4 = 000000	1008	1045		
APASS = 000000	1008	1013		
APRIOR = 000000	1008			
APTCSU = 000040	4509	4753#		
APTEMV = 000001	4502	4709	4751#	4785
APTSIZ = 000200	1119	4750#		
APTSP0 = 000100	4504	4711	4752#	
ASWREG = 000000	1008	1021		
ATESTN = 000000	1008	1012		
AUNIT = 000000	1008	1015		
AUSMR = 071110	3#	1008	1022	
AVECT1 = 000300	3#	1008	1047	
AVECT2 = 000000	1008	1048		
BAOBRK = 011070	4034#			
BAUD = 007400	912#			
BAUDRA = 007124	3265#			



MAINDEC-11-DVDVA-A  
DVDVAA.P11 06-MAY-77 15:29

MACY11 27(1006)

16-MAY-77 10:36 PAGE 125  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0122

ERRVEC=	000004	790#	1104	1105*	1116*	4829	4830*	4832*	4835*								
EVENOD=	000040	906#															
FALSE =	000000	4092#	4099														
FLAG	011406	2865#	2880*	2889	4099*	4126*	4131	4154	4166#								
FRERR =	020000	843#															
FRFD =	040000	915#	1548														
GNS =	***** U	923	1142	4048	4054	4060	4291	4297	4304	4310	4316	5047	5048	5049			
		5050	5051	5053	5055	5056	5057										
GTSWR =	104406	1137	5053#														
HOLD	010704	3910#	3911*	3914	3930#												
HOLDSC	011407	4111#	4116*	4122	4167#												
HOWLON=	000000	4071#	4097														
HT =	000011	700#	4517	4558													
I	001272	1076#	1196*	1217*	1220	1222	4200*	4203*	4205								
ILLMEM=	000004	802#	1191	1227													
INITFL	012116	1145#	4340	4343*	4387#												
INTFLA	011602	1203#	1208	3291*	3331	3335	3357*	3373	3424*	3464	3468	3502*	3519	3549*			
		3575	3579	4281*	4284#												
INTSKV	011574	1192	3303	3417	4271#												
IOTVEC=	000020	795#	1088*	1089*													
TICLRE	006560	3107#															
_F =	000012	701#	4552	4558													
LOOP	001720	1146#	4065														
MAINT =	000004	875#	1329	1342	1345	1359	1362	1376	1381	2747	2806	2863	3141	3427			
		3486	3498	3532	3622	3697	3720	3861	4008								
MSK =	000000	4178#	4192*	4208*	4210*	4215*											
MX	006310	2891	2895	2968#													
MYTYPE	011604	4287#	4782														
NEW	007156	3164*	3166*	3182*	3184*	3193	3201	3203	3221	3223	3271#						
NUMBER	010642	3875	3898#	3920	3976												
NUMBER	011510	4194*	4195	4197*	4205	4218#											
OLD	007162	3137*	3139*	3193	3201	3203	3221*	3223*	3272#								
ORERR =	040000	842#	3014	3053	3081												
PARITY=	000020	905#															
PBAUDS=	004000	865#															
PBAUD0=	010000	864#															
PBAUD1=	020000	863#															
PBAUD2=	040000	862#															
PBAUD3=	100000	861#															
PBR =	000200	908#	3128														
PERR =	010000	844#															
PIRQ =	177772	707#															
PIRQVE=	000240	801#															
PRO =	000000	724#	3313	3362	3433	3555	3626	3723	3828								
PR1 =	000040	725#															
PR2 =	000100	726#															
PR3 =	000140	727#															
PR4 =	000200	728#															
PR5 =	000240	729#															
PR6 =	000300	730#															
PR7 =	000340	731#	1193	1401	2390	2488	2591	3304	3418	3507	3839	3845					
PS =	177776	704#	705														
PSW =	177776	705#															
PWRVEC=	000024	796#	1094*	1095*	4440*	4441*	4450*	4456*	4468*	4469*							
RATES	007104	3155	3234	3239#													
RBUF	001262	1072#	1162*	1163*	2841	2955	3010	3053	3081	3096	3152	3668	3733	3772			











K10

MAINDEC-11-DVDVA-A  
DVDVA.P11

MACY11 27(1006)  
06-MAY-77 15:29

16-MAY-77 10:36 PAGE 130  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0127

3162	3164#	3165	3166#	3167	3176#	3177	3182#	3183	3184#	3185	3221#	3222	
3223#	3224	3229#	3233	3234#	3235	3291#	3292	3295#	3296	3298#	3299	3301#	
3302#	3303#	3304#	3305#	3306	3307#	3308	3310#	3311	3320#	3321	3324#	3354#	
3355	3357#	3358	3360#	3361	3368#	3387#	3388#	3390#	3391#	3392	3415#	3416#	
3417#	3418#	3419#	3420	3422#	3423	3424#	3425	3427#	3428	3430#	3431	3440#	
3441	3445#	3454#	3455	3457#	3486#	3487	3496#	3497	3498#	3499	3502#	3503	
3505#	3506	3513#	3532#	3533	3537#	3538	3543#	3544	3549#	3550	3552#	3553	
3560#	3561	3563#	3564	3565#	3566	3568#	3596#	3597	3600#	3601	3604#	3605#	
3607#	3608#	3609	3622#	3623	3632#	3638#	3639	3641	3645	3658#	3659	3668#	
3669	3676#	3678	3679#	3680	3697#	3698	3714#	3715	3720#	3721	3729#	3733#	
3734	3736#	3737	3739	3743	3756#	3757	3759#	3772#	3773	3776#	3778	3779#	
3780	3814#	3834#	3835	3837#	3838	3839#	3840	3843#	3844	3845#	3846	3849#	
3850	3852#	3853	3855#	3856	3861#	3862	3865#	3866	3868#	3869	3884#	3885	
3886#	3887	3910#	3912	3914#	3915	3917#	3918	3924#	3925	3943#	3945	3947#	
3949	3960#	3961	3962#	3963	3967#	3968	3973#	3974	3980#	3981	4008#	4009	
4011#	4012	4014#	4015	4095#	4096	4097#	4098	4099#	4100	4111#	4112	4116#	
4117	4126#	4127	4139#	4144#	4145	4157#	4158	4161#	4163	4162#	4193	4194#	
4198	4200#	4201	4203	4207	4208#	4209	4210#	4211	4215#	4216	4238#	4239	
4240#	4241	4243	4247	4248#	4249	4251	4255	4281#	4282	4343#	4344	4348#	
4352#	4353	4356#	4357	4358#	4359	4360#	4361	4362#	4363	4367#	4368	4369#	
4370	4371#	4372	4373#	4374	4381#	4382	4383#	4384					
1090	4768#												
SERRPC	001116												
SERRTB	001254												
SERTTL	001112												
SESCAP	001162												
SETABL	001214												
SETEND	001254												
SFATAL	001176												
SFFLG	013730												
SFILLC	001156												
SFILLS	001155												
SFSAND=	000310												
	3#												
	1465	1497	1513	1529	1550	1563	1579	1596	1615	1641	1557	1674	1693
	1719	1735	1752	1771	1797	1813	1830	1849	1881	1915	1948	978	1993
	2028	2051	2071	2091	2116	2139	2159	2179	2204	2227	2247	2267	2292
	2315	2335	2355	2382	2414	2432	2441	2458	2480	2513	2537	2560	2563
	2620	2642	2709	2789	2846	2879	2897	2916	2920	2923	2943	2960	3016
	3034	3055	3083	3098	3130	3171	3174	3195	3203	3333	3337	3375	3466
	3470	3521	3541	3577	3581	3685	3712	3785	3882	3922	3954	3958	3978
	4025	4110	4124	4156	4339	4342							
SFSBAO=	000401												
	3#												
	1465	1497	1513	1529	1550	1563	1579	1596	1615	1641	1657	1674	1693
	1719	1735	1752	1771	1797	1813	1830	1849	1881	1915	1948	1978	1993
	2028	2051	2071	2091	2116	2139	2159	2179	2204	2227	2247	2267	2292
	2315	2335	2355	2382	2414	2432	2441	2458	2480	2513	2537	2560	2583
	2620	2642	2709	2789	2846	2879	2897	2916	2920	2923	2943	2960	3016
	3034	3055	3083	3098	3130	3171	3174	3195	3205	3333	3337	3375	3466
	3470	3521	3541	3577	3581	3685	3712	3785	3882	3922	3954	3958	3978
	4025	4110	4124	4156	4339	4342							
SFSBLA=	000170												
SFSCAS=	000150												
SFSDEC=	000220												
SFSG00=	000400												
	3#												
	1208	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	
	1463	1495	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691
	1717	1733	1750	1769	1795	1811	1828	1847	1879	1913	1946	1976	1991

SFSIF = 000110

2026	2049	2069	2089	2114	2137	2157	2177	2202	2225	2245	2265	2290
2313	2333	2353	2380	2412	2430	2439	2456	2478	2511	2535	2558	2581
2618	2640	2686	2707	2727	2770	2787	2828	2844	2877	2895	2913	2914
2918	2921	2941	2958	3014	3032	3053	3081	3096	3128	3168	3169	3172
3193	3201	3331	3335	3373	3464	3468	3519	3539	3575	3579	3683	3710
3783	3880	3920	3952	3956	3976	4023	4108	4122	4154	4337	4340	
3#	1208	1214	1253	1259	1269	1275	1286	1292	1305	1311	1329	1335
1345	1351	1362	1368	1381	1387	1411	1417	1427	1433	1444	1450	1463
1469	1495	1501	1511	1517	1527	1533	1548	1554	1561	1567	1577	1583
1594	1600	1613	1619	1639	1645	1655	1661	1672	1678	1691	1697	1717
1723	1733	1739	1750	1756	1769	1775	1795	1801	1811	1817	1828	1834
1847	1853	1879	1890	1913	1924	1946	1959	1976	1983	1991	2006	2026
2033	2049	2057	2069	2077	2089	2097	2114	2121	2137	2145	2157	2165
2177	2185	2202	2209	2225	2233	2245	2253	2265	2273	2290	2297	2313
2321	2333	2341	2353	2361	2380	2387	2412	2418	2430	2436	2439	2445
2456	2462	2478	2485	2511	2517	2535	2541	2558	2564	2581	2588	2618
2624	2640	2646	2686	2691	2707	2713	2727	2732	2770	2775	2787	2793
2828	2833	2844	2850	2877	2882	2887	2895	2905	2918	2921	2932	2934
2941	2948	2958	2964	3014	3025	3032	3045	3053	3063	3081	3093	3096
3103	3128	3134	3172	3178	3186	3193	3197	3201	3203	3207	3216	3218
3331	3335	3341	3348	3350	3373	3379	3464	3468	3475	3481	3483	3519
3525	3539	3546	3575	3579	3586	3592	3594	3683	3690	3710	3717	3783
3790	3880	3891	3920	3926	3952	3956	3964	3969	3976	3983	4023	4029
4108	4113	4118	4122	4128	4154	4159	4337	4340	4345	4354	4364	4375
3#	3144	3226	3638	3692	3736	3793	4200	4212	4240	4248	4256	4259
3#	4105	4131	4146									

SFSINC= 000210  
SFSLOO= 000200  
SFSNAM= 000160  
SFSNO = 000403

1463	1495	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691
1717	1733	1750	1769	1795	1811	1828	1847	1879	1913	1946	1976	1991
2026	2049	2069	2089	2114	2137	2157	2177	2202	2225	2245	2265	2290
2313	2333	2353	2380	2412	2430	2439	2456	2478	2511	2535	2558	2581
2618	2640	2707	2787	2844	2877	2895	2914	2918	2921	2941	2958	3014
3032	3053	3081	3096	3128	3169	3172	3193	3201	3203	3331	3335	3373
3464	3468	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956
3976	4108	4337	4340									

SFSOR = 000320

3#	1210	1255	1271	1288	1307	1331	1347	1364	1383	1413	1429	1446
1465	1497	1513	1529	1550	1563	1579	1596	1615	1641	1657	1674	1693
1719	1735	1752	1771	1797	1813	1830	1849	1881	1915	1948	1978	1993
2028	2051	2071	2091	2116	2139	2159	2179	2204	2227	2247	2267	2292
2315	2335	2355	2382	2414	2432	2441	2458	2480	2513	2537	2560	2583
2620	2642	2709	2789	2846	2879	2897	2916	2920	2923	2943	2960	3016
3034	3055	3083	3098	3130	3171	3174	3195	3203	3205	3333	3337	3375
3466	3470	3521	3541	3577	3581	3685	3712	3785	3882	3922	3954	3958
3978	4025	4110	4124	4156	4339	4342						
3#	4071	4170	4178	4220	4227	4265	4288	4321	4326	4392		

SFSRTN= 000300  
SFSSEL= 000140  
SFSUNT= 000130

3873	1198	1222	2875	2889	3650	3653	3661	3664	3748	3751	3764	3767
3#	3875	4016	4019	4335	4377							
3#	2913	2914	2936	3168	3169	3189	3203					
3#	1208	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444
1463	1495	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691
1717	1733	1750	1769	1795	1811	1828	1847	1879	1913	1946	1976	1991
2026	2049	2069	2089	2114	2137	2157	2177	2202	2225	2245	2265	2290
2313	2333	2353	2380	2412	2430	2439	2456	2478	2511	2535	2558	2581
2618	2640	2707	2787	2844	2877	2895	2914	2918	2921	2941	2958	3014

SFSWHI= 000120  
SFSYES= 000402

MAINDEC-11-DVDVA-A  
DVDVA.P11 06-MAY-77

MACY11 27(1006)  
15:29

16-MAY-77 10:36 PAGE 132  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0129

	3032	3053	3081	3096	3128	3169	3172	3193	3201	3203	3331	3335	3373
	3464	3468	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956
	3976	4023	4108	4122	4154	4337	4340						
\$GADR 001120	979#												
\$GDRAT 001124	981#												
\$GET42 012204	4422#												
\$GTSWR 012760	4580#	5053											
\$HD = 000000	680												
\$HIBTS 001000	955#												
\$ICNT 001104	972#	4857*	4858	4860*	4871								
\$IFLEV= 177777	3#	1208#	1214#	1253#	1259#	1269#	1275#	1286#	1292#	1305#	1311#	1329#	1335#
	1045#	1351#	1362#	1368#	1381#	1387#	1411#	1417#	1427#	1433#	1444#	1450#	1463#
	1469#	1495#	1501#	1511#	1517#	1527#	1533#	1548#	1554#	1561#	1567#	1577#	1583#
	1594#	1600#	1613#	1619#	1639#	1645#	1655#	1661#	1672#	1678#	1691#	1697#	1717#
	1723#	1733#	1739#	1750#	1756#	1769#	1775#	1795#	1801#	1811#	1817#	1828#	1834#
	1847#	1853#	1879#	1890#	1913#	1924#	1946#	1959#	1976#	1983#	1991#	2006#	2026#
	2033#	2049#	2057#	2069#	2077#	2089#	2097#	2114#	2121#	2137#	2145#	2157#	2165#
	2177#	2185#	2202#	2209#	2225#	2233#	2245#	2253#	2265#	2273#	2290#	2297#	2313#
	2321#	2333#	2341*	2353#	2361#	2380#	2387#	2412#	2418#	2430#	2436#	2439#	2445#
	2456#	2462#	2478#	2485#	2511#	2517#	2535#	2541#	2558#	2564#	2581#	2588#	2618#
	2624#	2640#	2646#	2666#	2691#	2707#	2713#	2727#	2732#	2770#	2775#	2787#	2793#
	2828#	2833#	2844#	2850#	2877#	2887#	2895#	2905#	2918#	2921#	2932#	2934#	2941#
	2948#	2958#	2964#	3014#	3025#	3032#	3045#	3053#	3063#	3081#	3093#	3096#	3103#
	3128#	3134#	3172#	3186#	3193#	3201#	3216#	3218#	3331#	3335#	3348#	3350#	3373#
	3379#	3464#	3468#	3481#	3483#	3519#	3525#	3539#	3546#	3575#	3579#	3592#	3594#
	3683#	3690#	3710#	3717#	3783#	3790#	3880#	3891#	3920#	3926#	3952#	3956#	3964#
	3969#	3976#	3983#	4023#	4029#	4108#	4118#	4122#	4128#	4154#	4159#	4337#	4340#
	4354#	4375#											
\$ILLUP 012410	4440	4456	4473#										
\$INTAG 001135	986#	4608	4697										
\$ISKO = 000001	1208#	1214	1253#	1259	1269#	1275	1286#	1292	1305#	1311	1329#	1335	1345#
	1351	1362#	1368	1381#	1387	1411#	1417	1427#	1433	1444#	1450	1463#	1469
	1495#	1501	1511#	1517	1527#	1533	1548#	1554	1561#	1567	1577#	1583	1594#
	1600	1613#	1619	1639#	1645	1655#	1661	1672#	1678	1691#	1697	1717#	1723
	1733#	1739	1750#	1756	1769#	1775	1795#	1801	1811#	1817	1826#	1834	1847#
	1853	1879#	1890	1913#	1924	1946#	1959	1976#	1983	1991#	2006	2026#	2033
	2049#	2057	2069#	2077	2089#	2097	2114#	2121	2137#	2145	2157#	2165	2177#
	2185	2202#	2209	2225#	2233	2245#	2253	2265#	2273	2290#	2297	2313#	2321
	2333#	2341	2353#	2361	2380#	2387	2412#	2418	2430#	2436	2439#	2445	2456#
	2462	2478#	2485	2511#	2517	2535#	2541	2558#	2564	2581#	2588	2618#	2624
	2640#	2646	2686#	2691	2707#	2713	2727#	2732	2770#	2775	2787#	2793	2828#
	2833	2844#	2850	2877#	2887	2895#	2905	2916#	2934	2941#	2948	2958#	2964
	3014#	3025	3032#	3045	3053#	3063	3081#	3093	3096#	3103	3128#	3134	3172#
	3186	3193#	3218	3331#	3350	3373#	3379	3464#	3483	3519#	3525	3539#	3546
	3575#	3594	3683#	3690	3710#	3717	3783#	3790	3880#	3891	3920#	3926	3952#
	3969	3976#	3983	4023#	4029	4108#	4118	4122#	4128	4154#	4159	4337#	4375
\$ISK1 = 000001	2921#	2932	3201#	3216	3335#	3348	3458#	3481	3579#	3592	3956#	3964	4340#
	4354												
\$ITEMB 001114	976#	4299	4779*	4787	4808								
\$LF 001172	1003#	4558	4682	4691	4808								
\$FLG 013727	4741#	4747#											
\$LOCTA= 177777	3#	1198	1199	1209	1210	1214	1215	1223	1224	1254	1255	1259	1260
	1270	1271	1275	1276	1287	1288	1292	1293	1306	1307	1311	1312	1330
	1331	1335	1336	1346	1347	1351	1352	1363	1364	1368	1369	1382	1383
	1387	1388	1412	1413	1417	1418	1428	1429	1433	1434	1445	1446	1450
	1451	1464	1465	1469	1470	1496	1497	1501	1502	1512	1513	1517	1518

1528	1529	1533	1534	1549	1550	1554	1555	1562	1563	1567	1568	1578
1579	1583	1584	1595	1596	1600	1601	1614	1615	1619	1620	1640	1641
1645	1646	1656	1657	1661	1662	1673	1674	1678	1679	1692	1693	1697
1698	1718	1719	1723	1724	1734	1735	1739	1740	1751	1752	1756	1757
1770	1771	1775	1776	1796	1797	1801	1802	1812	1813	1817	1818	1829
1830	1834	1835	1848	1849	1853	1854	1880	1881	1890	1891	1914	1915
1924	1925	1947	1948	1959	1960	1977	1978	1983	1984	1992	1993	2006
2007	2027	2028	2033	2034	2050	2051	2057	2058	2070	2071	2077	2078
2090	2091	2097	2098	2115	2116	2121	2122	2138	2139	2145	2146	2158
2159	2165	2166	2178	2179	2185	2186	2203	2204	2209	2210	2226	2227
2233	2234	2246	2247	2253	2254	2266	2267	2273	2274	2291	2292	2297
2298	2314	2315	2321	2322	2334	2335	2341	2342	2354	2355	2361	2362
2381	2382	2387	2388	2413	2414	2418	2419	2431	2432	2436	2437	2440
2441	2445	2446	2457	2458	2462	2463	2479	2480	2495	2486	2512	2513
2517	2518	2536	2537	2541	2542	2559	2560	2564	2565	2582	2583	2588
2589	2619	2620	2624	2625	2641	2642	2646	2647	2686	2687	2691	2692
2708	2709	2713	2714	2727	2728	2732	2733	2770	2771	2775	2776	2788
2789	2793	2794	2828	2829	2833	2834	2845	2846	2850	2851	2875	2876
2878	2879	2882	2883	2884	2887	2888	2890	2891	2892	2893	2894	2896
2897	2905	2906	2913	2914	2915	2916	2919	2920	2922	2923	2932	2933
2934	2935	2936	2937	2938	2942	2943	2948	2949	2959	2960	2964	2965
3015	3016	3025	3026	3033	3034	3045	3046	3054	3055	3063	3064	3082
3083	3093	3094	3097	3098	3103	3104	3129	3130	3134	3135	3145	3146
3147	3148	3149	3150	3151	3168	3169	3170	3171	3173	3174	3178	3179
3180	3186	3187	3189	3190	3191	3194	3195	3197	3198	3199	3202	3203
3204	3205	3207	3208	3209	3216	3217	3218	3219	3226	3227	3228	3232
3333	3336	3337	3341	3342	3343	3348	3349	3350	3351	3374	3375	3379
3380	3465	3466	3469	3470	3475	3476	3477	3481	3482	3483	3484	3520
3521	3525	3526	3540	3541	3546	3547	3576	3577	3580	3581	3586	3587
3588	3592	3593	3594	3595	3639	3640	3641	3642	3643	3644	3645	3650
3651	3654	3655	3661	3662	3665	3666	3684	3685	3640	3691	3692	3693
3694	3711	3712	3717	3718	3737	3738	3739	3740	3741	3742	3743	3748
3749	3752	3753	3764	3765	3768	3769	3784	3785	3790	3791	3793	3794
3795	3873	3874	3876	3877	3881	3882	3891	3892	3921	3922	3926	3927
3953	3954	3957	3958	3964	3965	3969	3970	3977	3978	3983	3984	4016
4017	4020	4021	4024	4025	4029	4030	4070	4105	4106	4109	4110	4113
4114	4115	4118	4119	4123	4124	4128	4129	4132	4133	4134	4135	4146
4147	4148	4155	4156	4157	4158	4159	4160	4162	4163	4170	4171	4172
4173	4177	4201	4202	4203	4204	4205	4206	4207	4212	4213	4214	4217
4218	4220	4221	4222	4226	4241	4242	4243	4244	4245	4246	4247	4249
4250	4251	4252	4253	4254	4255	4256	4257	4258	4259	4260	4261	4265
4266	4267	4287	4321	4322	4323	4325	4335	4336	4338	4339	4341	4342
4345	4346	4347	4354	4355	4364	4365	4366	4375	4376	4378	4379	4385
4386	4392	4393	4394									
973*	1100*	4848*	4864*	4869	4871							
974*	1101*	1200*	1250*	1264*	1280*	1297*	1326*	1340*	1356*	1373*	1408*	1422*
1438*	1455*	1490*	1506*	1522*	1558*	1572*	1588*	1605*	1636*	1650*	1666*	1682*
1714*	1728*	1744*	1761*	1792*	1806*	1822*	1839*	1877*	1910*	1944*	1988*	2042*
2062*	2082*	2130*	2150*	2170*	2218*	2238*	2258*	2306*	2326*	2346*	2397*	2424*
2450*	2495*	2522*	2546*	2606*	2629*	2663*	2695*	2750*	2780*	2808*	2983*	3030*
3049*	3067*	3307*	3354*	3422*	3496*	3537*	4798	4848	4865*	4871		
3#	1198	1199	1208	1210	1214	1222	1253	1255	1259	1269	1271	1275
1286	1288	1292	1305	1307	1311	1329	1331	1335	1345	1347	1351	1362
1364	1368	1381	1383	1387	1411	1413	1417	1427	1429	1433	1444	1446
1450	1463	1465	1469	1495	1497	1501	1511	1513	1517	1527	1529	1533
1548	1550	1554	1561	1563	1567	1577	1579	1583	1594	1596	1600	1613

SLPADR 001106  
SLPERR 001110

SLSTCN= 177777

1615	1619	1639	1641	1645	1655	1657	1661	1672	1674	1678	1691	1693
1697	1717	1719	1723	1733	1735	1739	1750	1752	1756	1769	1771	1775
1795	1797	1801	1811	1813	1817	1828	1830	1834	1847	1849	1853	1879
1881	1890	1913	1915	1924	1946	1948	1959	1976	1978	1983	1991	1993
2006	2026	2028	2033	2049	2051	2057	2069	2071	2077	2089	2091	2097
2114	2116	2121	2137	2139	2145	2157	2159	2165	2177	2179	2185	2202
2204	2209	2225	2227	2233	2245	2247	2253	2265	2267	2273	2290	2292
2297	2313	2315	2321	2333	2335	2341	2353	2355	2361	2380	2382	2387
2412	2414	2418	2430	2432	2436	2439	2441	2445	2456	2458	2462	2478
2480	2485	2511	2513	2517	2535	2537	2541	2558	2560	2564	2581	2583
2588	2618	2620	2624	2640	2642	2646	2686	2687	2691	2707	2709	2713
2727	2728	2732	2770	2771	2775	2787	2789	2793	2828	2829	2833	2844
2846	2850	2875	2876	2877	2879	2883	2884	2887	2889	2894	2895	2897
2905	2910	2914	2916	2918	2920	2921	2923	2932	2934	2936	2937	2941
2943	2948	2958	2960	2964	3014	3016	3025	3032	3034	3045	3053	3055
3063	3081	3083	3093	3096	3098	3103	3128	3130	3134	3144	3146	3147
3148	3151	3168	3169	3171	3172	3174	3179	3180	3186	3169	3190	3193
3195	3198	3199	3201	3205	3208	3209	3216	3218	3226	3227	3331	3333
3335	3337	3342	3343	3348	3350	3373	3375	3379	3464	3466	3468	3470
3476	3477	3481	3483	3519	3521	3525	3539	3541	3546	3575	3577	3579
3581	3587	3588	3592	3594	3638	3640	3641	3642	3645	3650	3651	3653
3661	3662	3664	3683	3685	3690	3692	3693	3710	3712	3717	3736	3738
3739	3740	3743	3748	3749	3751	3764	3765	3767	3783	3785	3790	3793
3794	3873	3874	3875	3880	3882	3891	3920	3922	3926	3952	3954	3956
3958	3964	3969	3976	3978	3983	4016	4017	4019	4023	4025	4029	4071
4105	4106	4108	4110	4114	4115	4118	4122	4124	4128	4146	4147	4154
4156	4159	4170	4178	4200	4202	4203	4204	4207	4212	4213	4220	4227
4240	4242	4243	4244	4247	4248	4250	4251	4252	4255	4256	4257	4259
4260	4265	4288	4321	4326	4335	4336	4337	4339	4340	4342	4346	4347
4354	4365	4366	4375	4377	4392							
	1145	1146	1148	1149	1152	1153	1155	1156	1157	1158	1160	1161
1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174
1175	1187	1188	1190	1191	1192	1193	1194	1195	1196	1197	1200	1201
1203	1204	1208	1209	1210	1217	1218	1219	1220	1221	1222	1223	1224
1227	1228	1229	1230	1231	1232	1250	1251	1253	1254	1255	1264	1265
1266	1267	1269	1270	1271	1280	1281	1283	1284	1286	1287	1288	1297
1298	1300	1301	1305	1306	1307	1326	1327	1329	1330	1331	1340	1341
1342	1343	1345	1346	1347	1356	1357	1359	1360	1362	1363	1364	1373
1374	1376	1377	1381	1382	1383	1408	1409	1411	1412	1413	1422	1423
1424	1425	1427	1428	1429	1438	1439	1441	1442	1444	1445	1446	1455
1456	1458	1459	1463	1464	1465	1490	1491	1492	1493	1495	1496	1497
1506	1507	1509	1510	1511	1512	1513	1522	1523	1524	1525	1527	1528
1529	1548	1549	1550	1551	1552	1558	1559	1561	1562	1563	1572	1573
1574	1575	1577	1578	1579	1588	1589	1591	1592	1594	1595	1596	1605
1606	1608	1609	1613	1614	1615	1636	1637	1639	1640	1641	1650	1651
1652	1653	1655	1656	1657	1666	1667	1669	1670	1672	1673	1674	1682
1683	1686	1687	1691	1692	1693	1714	1715	1717	1718	1719	1728	1729
1730	1731	1733	1734	1735	1744	1745	1747	1748	1750	1751	1752	1761
1762	1764	1765	1769	1770	1771	1792	1793	1795	1796	1797	1806	1807
1808	1809	1811	1812	1813	1822	1823	1825	1826	1828	1829	1830	1839
1840	1842	1843	1847	1848	1849	1877	1878	1879	1880	1881	1910	1911
1913	1914	1915	1944	1945	1946	1947	1948	1976	1977	1978	1980	1981
1988	1989	1991	1992	1993	2026	2027	2028	2030	2031	2042	2043	2046
2047	2049	2050	2051	2062	2063	2066	2067	2069	2070	2071	2082	2083
2086	2087	2089	2090	2091	2114	2115	2116	2118	2119	2130	2131	2134
2135	2137	2138	2139	2150	2151	2154	2155	2157	2158	2159	2170	2171

SLSTIN= 000000

2174	2175	2177	2178	2179	2202	2203	2204	2206	2207	2218	2219	2222
2223	2225	2226	2227	2238	2239	2242	2243	2245	2246	2247	2258	2259
2262	2263	2265	2266	2267	2290	2291	2292	2294	2295	2306	2307	2310
2311	2313	2314	2315	2326	2327	2330	2331	2333	2334	2335	2346	2347
2350	2351	2353	2354	2355	2380	2381	2382	2384	2385	2397	2398	2400
2401	2403	2404	2405	2406	2407	2409	2410	2412	2413	2414	2424	2425
2428	2429	2430	2431	2432	2439	2440	2441	2450	2451	2454	2455	2456
2457	2458	2478	2479	2480	2482	2483	2495	2496	2499	2500	2502	2503
2504	2505	2506	2508	2509	2511	2512	2513	2522	2523	2527	2528	2530
2531	2532	2533	2534	2535	2536	2537	2546	2547	2550	2551	2553	2554
2555	2556	2557	2558	2559	2560	2581	2582	2583	2585	2586	2600	2601
2602	2603	2606	2607	2610	2611	2613	2614	2615	2616	2617	2618	2619
2620	2629	2630	2632	2633	2635	2636	2637	2638	2639	2640	2641	2642
2663	2664	2671	2672	2676	2677	2678	2679	2680	2681	2682	2683	2686
2687	2695	2696	2702	2703	2707	2708	2709	2719	2720	2721	2722	2723
2724	2725	2724	2727	2729	2747	2748	2750	2751	2754	2755	2760	2761
2762	2763	27	2765	2766	2767	2770	2771	2780	2781	2787	2788	2789
2806	2807	2808	2809	2814	2815	2819	2820	2821	2822	2823	2824	2825
2826	2828	2829	2841	2842	2844	2845	2846	2863	2864	2865	2866	2867
2868	2873	2874	2877	2878	2879	2880	2881	2882	2883	2885	2886	2889
2890	2891	2892	2893	2895	2896	2897	2902	2903	2914	2915	2916	2918
2919	2920	2921	2922	2923	2929	2930	2936	2937	2941	2942	2943	2955
2956	2958	2959	2960	2983	2984	2990	2991	2993	2994	2995	2996	2997
3000	3001	3003	3004	3005	3006	3007	3010	3011	3014	3015	3016	3022
3023	3030	3031	3032	3033	3034	3042	3043	3049	3050	3053	3054	3055
3067	3068	3072	3073	3075	3076	3077	3078	3079	3081	3082	3083	3090
3091	3096	3097	3098	3128	3129	3130	3131	3132	3137	3138	3139	3140
3141	3142	3144	3145	3146	3147	3148	3149	3150	3151	3152	3153	3155
3156	3158	3159	3161	3162	3164	3165	3166	3167	3169	3170	3171	3172
3173	3174	3176	3177	3178	3179	3182	3183	3184	3185	3189	3190	3193
3194	3195	3197	3198	3201	3202	3203	3204	3205	3207	3208	3221	3222
3223	3224	3226	3227	3229	3230	3231	3232	3233	3234	3235	3291	3292
3295	3296	3298	3299	3301	3302	3303	3304	3305	3306	3307	3308	3310
3311	3320	3321	3324	3325	3326	3327	3328	3331	3332	3333	3335	3336
3337	3341	3342	3354	3355	3357	3358	3360	3361	3368	3364	3370	3371
3372	3373	3374	3375	3387	3388	3389	3390	3391	3392	3415	3416	3417
3418	3419	3420	3422	3423	3424	3425	3427	3428	3430	3431	3440	3441
3445	3446	3447	3448	3449	3450	3451	3452	3454	3455	3457	3458	3459
3460	3461	3464	3465	3466	3468	3469	3470	3475	3476	3486	3487	3496
3497	3498	3499	3502	3503	3505	3506	3513	3514	3515	3516	3517	3519
3520	3521	3532	3533	3537	3538	3539	3540	3541	3543	3544	3549	3550
3552	3553	3560	3561	3563	3564	3565	3566	3568	3569	3570	3571	3572
3575	3576	3577	3579	3580	3581	3586	3587	3596	3597	3600	3601	3604
3605	3606	3607	3608	3609	3622	3623	3632	3633	3634	3635	3638	3639
3640	3641	3642	3643	3644	3645	3653	3654	3655	3658	3659	3664	3665
3666	3668	3669	3676	3677	3678	3679	3680	3683	3684	3685	3692	3693
3697	3698	3710	3711	3712	3714	3715	3720	3721	3729	3730	3731	3732
3733	3734	3736	3737	3738	3739	3740	3741	3742	3743	3751	3752	3753
3756	3757	3759	3760	3761	3762	3763	3767	3768	3769	3772	3773	3776
3777	3778	3779	3780	3783	3784	3785	3793	3794	3814	3815	3816	3817
3834	3835	3837	3838	3839	3840	3843	3844	3845	3846	3849	3850	3852
3853	3855	3856	3861	3862	3865	3866	3868	3869	3875	3876	3877	3880
3881	3882	3884	3885	3886	3887	3910	3911	3912	3914	3915	3917	3918
3920	3921	3922	3924	3925	3943	3944	3945	3947	3948	3949	3952	3953
3954	3956	3957	3958	3960	3961	3962	3963	3967	3968	3973	3974	3976
3977	3978	3980	3981	4008	4009	4011	4012	4014	4015	4019	4020	4021



4023	4024	4025	4095	4096	4097	4098	4099	4100	4108	4109	4110	4111
4112	4113	4114	4116	4117	4122	4123	4124	4126	4127	4131	4132	4133
4134	4135	4139	4140	4141	4142	4143	4144	4145	4146	4147	4154	4155
4156	4157	4158	4161	4162	4163	4171	4172	4173	4174	4192	4193	4194
4195	-196	4197	4198	4200	4201	4202	4203	4204	4205	4206	4207	4208
4209	4210	4211	4212	4213	4215	4216	4217	4218	4222	4223	4238	4239
4240	4241	4242	4243	4244	4245	4246	4247	4248	4249	4250	4251	4252
4253	4254	4255	4256	4257	4259	4260	4267	4268	4281	4282	4323	4324
4337	4338	4339	4340	4341	4342	4343	4344	4345	4346	4348	4349	4352
4353	4356	4357	4358	4359	4360	4361	4362	4363	4364	4365	4367	4368
4369	4370	4371	4372	4373	4374	4377	4378	4379	4381	4382	4383	4384
4385	4386	4394	4395									
3#	1198	1199	1208	1210	1214	1222	1253	1255	1259	1269	1271	1275
1286	1288	1292	1305	1307	1311	1329	1331	1335	1345	1347	1351	1362
1364	1368	1381	1383	1387	1411	1413	1417	1427	1429	1433	1444	1446
1450	1463	1465	1469	1495	1497	1501	1511	1513	1517	1527	1529	1533
1548	1550	1554	1561	1563	1567	1577	1579	1583	1594	1596	1600	1613
1615	1619	1639	1641	1645	1655	1657	1661	1672	1674	1678	1691	1693
1697	1717	1719	1723	1733	1735	1739	1750	1752	1756	1769	1771	1775
1795	1797	1801	1811	1813	1817	1828	1830	1834	1847	1849	1853	1879
1881	1890	1913	1915	1924	1946	1948	1959	1976	1978	1983	1991	1993
2006	2026	2028	2033	2049	2051	2057	2069	2071	2077	2089	2091	2097
2114	2116	2121	2137	2139	2145	2157	2159	2165	2177	2179	2185	2202
2204	2209	2225	2227	2233	2245	2247	2253	2265	2267	2273	2290	2292
2297	2313	2315	2321	2333	2335	2341	2353	2355	2361	2380	2382	2387
2412	2414	2418	2430	2432	2436	2439	2441	2445	2456	2458	2462	2478
2480	2485	2511	2513	2517	2535	2537	2541	2558	2560	2564	2581	2583
2588	2618	2620	2624	2640	2642	2646	2686	2687	2691	2707	2709	2713
2727	2728	2732	2770	2771	2775	2787	2789	2793	2828	2829	2833	2844
2846	2850	2875	2876	2877	2879	2882	2883	2884	2887	2889	2895	2897
2905	2913	2914	2916	2918	2920	2921	2923	2932	2934	2936	2937	2941
2943	2948	2958	2960	2964	3014	3016	3025	3032	3034	3045	3053	3055
3063	3081	3083	3093	3096	3098	3103	3128	3130	3134	3144	3146	3147
3148	3151	3168	3169	3171	3172	3174	3178	3179	3180	3186	3189	3190
3193	3195	3197	3198	3199	3201	3205	3207	3208	3209	3216	3218	3226
3227	3331	3333	3335	3337	3341	3342	3343	3348	3350	3373	3375	3379
3464	3466	3468	3470	3475	3476	3477	3481	3483	3519	3521	3525	3539
3541	3546	3575	3577	3579	3581	3586	3587	3588	3592	3594	3638	3640
3641	3642	3645	3650	3651	3653	3661	3662	3664	3683	3685	3690	3692
3693	3710	3712	3717	3736	3738	3739	3740	3743	3748	3749	3751	3764
3765	3767	3783	3785	3790	3793	3794	3873	3874	3875	3880	3882	3891
3920	3922	3926	3952	3954	3956	3958	3964	3969	3976	3978	3983	4016
4017	4019	4023	4025	4029	4071	4105	4106	4108	4110	4113	4114	4115
4118	4122	4124	4128	4131	4146	4147	4154	4156	4159	4170	4178	4200
4202	4203	4204	4207	4212	4213	4220	4227	4240	4242	4243	4244	4247
4248	4250	4251	4252	4255	4256	4257	4259	4260	4265	4288	4321	4326
4335	4336	4337	4339	4340	4342	4345	4346	4347	4354	4364	4365	4366
4375	4377	4392										
3#	1198	1199	1214	1215	1259	1260	1275	1276	1292	1293	1311	1312
1335	1336	1351	1352	1368	1369	1387	1388	1417	1418	1433	1434	1450
1451	1469	1470	1501	1502	1517	1518	1533	1534	1554	1555	1567	1568
1583	1584	1600	1601	1619	1620	1645	1646	1661	1662	1678	1679	1697
1698	1723	1724	1739	1740	1756	1757	1775	1776	1801	1802	1817	1818
1834	1835	1853	1854	1890	1891	1924	1925	1959	1960	1983	1984	2006
2007	2033	2034	2057	2058	2077	2078	2097	2098	2121	2122	2145	2146
2165	2166	2185	2186	2209	2210	2233	2234	2253	2254	2273	2274	2297

SLSTST= 177777

SLSTA= 000000

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0134

		2298	2321	2322	2341	2342	2361	2362	2387	2388	2418	2419	2436	2437
		2445	2446	2462	2463	2485	2486	2517	2518	2541	2542	2564	2565	2588
		2589	2624	2625	2646	2647	2691	2692	2713	2714	2732	2733	2775	2776
		2793	2794	2833	2834	2850	2851	2875	2876	2883	2884	2887	2888	2893
		2894	2905	2906	2913	2914	2932	2933	2934	2935	2937	2938	2948	2949
		2964	2965	3025	3026	3045	3046	3063	3064	3093	3094	3103	3104	3134
		3135	3146	3147	3148	3149	3168	3169	3179	3180	3186	3187	3190	3191
		3198	3199	3208	3209	3216	3217	3218	3219	3227	3228	3342	3343	3348
		3349	3350	3351	3379	3380	3476	3477	3481	3482	3483	3484	3525	3526
		3546	3547	3587	3588	3592	3593	3594	3595	3640	3641	3642	3643	3650
		3651	3661	3662	3590	3691	3693	3694	3717	3718	3738	3739	3740	3741
		3748	3749	3764	3765	3790	3791	3794	3795	3873	3874	3891	3892	3926
		3927	3964	3965	3969	3970	3983	3984	4016	4017	4029	4030	4070	4071
		4105	4106	4114	4115	4118	4119	4128	4129	4147	4148	4159	4160	4170
		4171	4172	4173	4177	4178	4202	4203	4204	4205	4213	4214	4220	4221
		4222	4226	4227	4242	4243	4244	4245	4250	4251	4252	4253	4257	4258
		4260	4261	4265	4266	4267	4287	4288	4321	4322	4323	4325	4326	4335
		4336	4346	4347	4354	4355	4365	4366	4375	4376	4392	4393	4394	
SMADR1	001226	1036#												
SMADR2	001232	1040#												
SMADR3	001236	1043#												
SMADR4	001242	1046#												
SMAIL	001174	956	960	1009#	1118	1133	1185	1246	1322	1399	1487	1546	1633	1711
		1789	1870	1904	1938	1972	2023	2111	2199	2287	2377	2475	2578	2660
		2743	2802	2860	2980	3126	3288	3411	3619	3708	3810	4005	4502	4785
		4863												
SMAMS1	001224	1030#												
SMAMS2	001230	1038#												
SMAMS3	001234	1041#												
SMAMS4	001240	1044#												
SMBADR	001002	956#												
SBCALL=	000000	1#	3											
SMFLG	013726	4701*	4707	4742*	4746#									
SMNEW	013453	4583	4695#											
SMGAD	001210	1016#	4717*	4720										
SMGLG	001212	1017#	4722*											
SMGTY	001174	1010#	4715	4723*	4735	4739*								
SMWR	013442	4580	4693#											
SMTYP1	001225	1031#												
SMTYP2	001231	1039#												
SMTYP3	001235	1042#												
SMTYP4	001241	1045#												
SMXCNT	014410	4861	4871#											
SNESTL=	177777	3#	1198#	1208#	1214#	1222#	1253#	1259#	1269#	1275#	1286#	1292#	1305#	1311#
		1329#	1335#	1345#	1351#	1362#	1368#	1381#	1387#	1411#	1417#	1427#	1433#	1444#
		1450#	1463#	1469#	1495#	1501#	1511#	1517#	1527#	1533#	1548#	1554#	1561#	1567#
		1577#	1583#	1594#	1600#	1613#	1619#	1639#	1645#	1655#	1661#	1672#	1678#	1691#
		1697#	1717#	1723#	1733#	1739#	1750#	1756#	1769#	1775#	1795#	1801#	1811#	1817#
		1828#	1834#	1847#	1853#	1879#	1890#	1913#	1924#	1946#	1959#	1976#	1983#	1991#
		2006#	2026#	2033#	2049#	2057#	2069#	2077#	2089#	2097#	2114#	2121#	2137#	2145#
		2157#	2165#	2177#	2185#	2202#	2209#	2225#	2233#	2245#	2253#	2265#	2273#	2290#
		2297#	2313#	2321#	2333#	2341#	2353#	2361#	2380#	2387#	2412#	2418#	2430#	2436#
		2439#	2445#	2456#	2462#	2478#	2485#	2511#	2517#	2535#	2541#	2558#	2564#	2581#
		2588#	2618#	2624#	2640#	2646#	2686#	2691#	2707#	2713#	2727#	2732#	2770#	2775#
		2787#	2793#	2828#	2833#	2844#	2850#	2875#	2877#	2882#	2887#	2889#	2895#	2905#
		2913#	2918#	2921#	2932#	2934#	2936#	2941#	2948#	2958#	2964#	3014#	3025#	3032#





SSAVLE= 177777	3#	2936#	2937#	3147#	3151#	3189#	3190#	3641#	3645#	3739#	3743#	4146#	4147#
	4203#	4207#	4243#	4247#	4251#	4255#							
SSAVRE= ***** U	5058												
SSAVR6 012414	4449#	4457	4458#	4459#	4475#								
SSCOPE 014132	1088	4822#											
SSETUP= 000137	1079#	1087	1088	1090	1092	1094	1096	1097	1098	1100	1127	1130	4408
	4563	4697	4769	4795	4803	4823							
SSSKO = 000233	2936#	2937	3147#	3151	3189#	3190	3641#	3645	3739#	3743	4146#	4147	4203#
	4207	4243#	4247	4251#	4255								
SSTUP = 177777	1079#												
SSVLAD 014340	4833	4862#											
SSVPC = 000204	932#	937											
SSWR = 167400	3#	680	684	685	686	687	688	689	690	998	999	1000	1097
	1098	1100	1101	1184	1245	1321	1398	1486	1545	1632	1710	1788	1869
	1903	1937	1971	2022	2110	2198	2286	2376	2474	2577	2659	2742	2801
	2859	2979	3125	3287	3410	3618	3707	3809	4004	4045	4403	4409	4424
	4430	4432	4472	4760	4761	4762	4763	4764	4773	4780	4752	4796	4808
	4814	4815	4816	4817	4818	4824	4836	4838	4839	4842	4843	4844	4851
	4852	4853	4865	4868	4871								
SSWREG 001216	1021#	1121											
SSWRMK= 000000	690	691	4818	4819	4840								
STAGLE= 177777	3#	1199#	1210#	1214#	1222#	1255#	1259#	1271#	1275#	1288#	1292#	1307#	1311#
	1331#	1335#	1347#	1351#	1364#	1368#	1383#	1387#	1413#	1417#	1429#	1433#	1446#
	1450#	1465#	1469#	1497#	1501#	1513#	1517#	1529#	1533#	1550#	1554#	1563#	1567#
	1579#	1583#	1596#	1600#	1615#	1619#	1641#	1645#	1657#	1661#	1674#	1678#	1693#
	1697#	1719#	1723#	1735#	1739#	1752#	1756#	1771#	1775#	1797#	1801#	1813#	1817#
	1830#	1834#	1849#	1853#	1881#	1890#	1915#	1924#	1948#	1959#	1978#	1983#	1993#
	2006#	2028#	2033#	2051#	2057#	2071#	2077#	2091#	2097#	2116#	2121#	2139#	2145#
	2159#	2165#	2179#	2185#	2204#	2209#	2227#	2233#	2247#	2253#	2267#	2273#	2292#
	2297#	2315#	2321#	2335#	2341#	2355#	2361#	2382#	2387#	2414#	2418#	2432#	2436#
	2441#	2445#	2458#	2462#	2480#	2485#	2513#	2517#	2537#	2541#	2560#	2564#	2583#
	2588#	2620#	2624#	2642#	2646#	2687#	2691#	2709#	2713#	2728#	2732#	2771#	2775#
	2789#	2793#	2829#	2833#	2846#	2850#	2876#	2879#	2883#	2884#	2887#	2889#	2897#
	2905#	2914#	2916#	2920#	2923#	2932#	2934#	2936#	2943#	2948#	2960#	2964#	3016#
	3025#	3034#	3045#	3055#	3063#	3083#	3093#	3098#	3103#	3130#	3134#	3146#	3148#
	3151#	3169#	3171#	3174#	3179#	3180#	3186#	3189#	3195#	3198#	3199#	3205#	3208#
	3209#	3216#	3218#	3226#	3227#	3333#	3337#	3342#	3343#	3348#	3350#	3375#	3379#
	3466#	3470#	3476#	3477#	3481#	3483#	3521#	3525#	3541#	3546#	3577#	3581#	3587#
	3588#	3592#	3594#	3640#	3642#	3645#	3651#	3653#	3662#	3664#	3685#	3690#	3692#
	3693#	3712#	3717#	3738#	3740#	3743#	3749#	3751#	3765#	3767#	3785#	3790#	3793#
	3794#	3874#	3875#	3882#	3891#	3922#	3926#	3954#	3958#	3964#	3969#	3978#	3983#
	4017#	4019#	4025#	4029#	4106#	4110#	4114#	4115#	4118#	4124#	4128#	4131	4146#
	4156#	4159#	4202#	4204#	4207#	4212#	4213#	4242#	4244#	4247#	4250#	4252#	4255#
	4256#	4257#	4259#	4260#	4336#	4339#	4342#	4346#	4347#	4354#	4365#	4366#	4375#
	4377#												
STAGNU= 000246	3#	1198	1199#	1209	1210#	1254	1255#	1270	1271#	1287	1288#	1306	1307#
	1330	1331#	1346	1347#	1363	1364#	1382	1383#	1412	1413#	1428	1429#	1445
	1446#	1464	1465#	1496	1497#	1512	1513#	1528	1529#	1549	1550#	1562	1563#
	1578	1579#	1595	1596#	1614	1615#	1640	1641#	1656	1657#	1673	1674#	1692
	1693#	1718	1719#	1734	1735#	1751	1752#	1770	1771#	1796	1797#	1812	1813#
	1829	1830#	1848	1849#	1880	1881#	1914	1915#	1947	1948#	1977	1978#	1992
	1993#	2027	2028#	2050	2051#	2070	2071#	2090	2091#	2115	2116#	2138	2139#
	2158	2159#	2178	2179#	2203	2204#	2226	2227#	2246	2247#	2266	2267#	2291
	2292#	2314	2315#	2334	2335#	2354	2355#	2381	2382#	2413	2414#	2431	2432#
	2440	2441#	2457	2458#	2479	2480#	2512	2513#	2536	2537#	2559	2560#	2582
	2583#	2619	2620#	2641	2642#	2686	2687#	2708	2709#	2727	2728#	2770	2771#

# H11

MAINDEC-11-DVDVA-A  
DVDVAA.P11

MACY11 27(1006)  
06-MAY-77 15:29

16-MAY-77 10:36 PAGE 140  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0137

STEMP = 000300

2788	2789	2828	2829	2845	2846	2875	2876	2878	2879	2882	2884	2890
2893	2894	2896	2897	2913	2914	2915	2916	2919	2920	2922	2923	2942
2943	2959	2960	3015	3016	3033	3034	3054	3055	3082	3083	3097	3098
3129	3130	3145	3146	3147	3150	3151	3168	3169	3170	3171	3173	3174
3178	3180	3194	3195	3197	3199	3202	3204	3205	3207	3209	3332	3333
3336	3337	3341	3343	3374	3375	3465	3466	3469	3470	3475	3477	3520
3521	3540	3541	3576	3577	3580	3581	3586	3588	3639	3640	3641	3644
3645	3650	3651	3661	3662	3684	3685	3711	3712	3737	3738	3739	3742
3743	3748	3749	3764	3765	3784	3785	3873	3874	3881	3882	3921	3922
3953	3954	3957	3958	3977	3978	4016	4017	4024	4025	4071	4105	4106
4109	4110	4113	4115	4123	4124	4155	4156	4178	4201	4202	4203	4206
4207	4227	4241	4242	4243	4246	4247	4249	4250	4251	4254	4255	4288
4326	4335	4336	4338	4339	4341	4342	4345	4347	4364	4366		
1145	1146	1152	1153	1155	1156	1157	1158	1160	1161	1162	1164	1165
1167	1168	1170	1171	1173	1174	1175	1187	1188	1190	1191	1192	1193
1194	1195	1196	1197	1200	1201	1203	1204	1214	1217	1218	1219	1221
1222	1223	1227	1228	1230	1231	1232	1250	1251	1259	1264	1265	1266
1267	1275	1280	1281	1283	1284	1292	1297	1298	1300	1301	1311	1326
1327	1331	1340	1341	1342	1343	1351	1356	1357	1359	1360	1368	1373
1374	1376	1377	1387	1408	1409	1417	1422	1423	1424	1425	1433	1438
1439	1441	1442	1450	1455	1456	1458	1459	1469	1490	1491	1492	1493
1501	1506	1507	1509	1510	1517	1522	1523	1524	1525	1533	1551	1552
1554	1558	1559	1567	1572	1573	1574	1575	1583	1588	1589	1591	1592
1600	1605	1606	1608	1609	1619	1636	1637	1645	1650	1651	1652	1653
1661	1666	1667	1669	1670	1678	1682	1683	1686	1687	1697	1714	1715
1723	1728	1729	1730	1731	1739	1744	1745	1747	1748	1756	1761	1762
1764	1765	1775	1792	1793	1801	1806	1807	1808	1809	1817	1822	1823
1825	1826	1834	1839	1840	1842	1843	1853	1877	1878	1890	1910	1911
1924	1944	1945	1959	1980	1981	1983	1988	1989	2006	2030	2031	2033
2042	2043	2046	2047	2057	2062	2063	2066	2067	2077	2082	2083	2086
2087	2097	2118	2119	2121	2130	2131	2134	2135	2145	2150	2151	2154
2155	2165	2170	2171	2174	2175	2185	2206	2207	2209	2218	2219	2222
2223	2233	2238	2239	2242	2243	2253	2258	2259	2262	2263	2273	2294
2295	2297	2306	2307	2310	2311	2321	2326	2327	2330	2331	2341	2346
2347	2350	2351	2361	2384	2385	2387	2397	2398	2400	2401	2409	2410
2418	2424	2425	2428	2429	2436	2445	2450	2451	2454	2455	2462	2482
2483	2485	2495	2496	2499	2500	2508	2509	2517	2522	2523	2527	2528
2541	2546	2547	2550	2551	2564	2585	2586	2588	2600	2601	2602	2603
2606	2607	2610	2611	2624	2629	2630	2632	2633	2646	2663	2664	2671
2672	2691	2695	2696	2702	2703	2713	2732	2747	2748	2750	2751	2754
2755	2775	2780	2781	2793	2806	2807	2808	2809	2814	2815	2833	2841
2842	2850	2863	2864	2865	2866	2867	2868	2873	2874	2880	2881	2882
2883	2885	2886	2887	2889	2892	2902	2903	2905	2929	2930	2932	2934
2936	2937	2948	2955	2956	2964	2983	2984	2990	2991	3000	3001	3010
3011	3022	3023	3025	3030	3031	3042	3043	3045	3049	3050	3063	3067
3068	3072	3073	3090	3091	3093	3103	3131	3132	3134	3137	3138	3139
3140	3141	3142	3148	3151	3152	3153	3155	3156	3158	3159	3161	3162
3164	3165	3166	3167	3176	3177	3178	3179	3182	3183	3184	3185	3186
3189	3190	3197	3198	3207	3208	3216	3218	3221	3222	3223	3224	3226
3227	3229	3233	3234	3235	3291	3292	3295	3296	3298	3299	3301	3302
3303	3304	3305	3306	3307	3308	3310	3311	3320	3321	3341	3342	3348
3350	3354	3355	3357	3358	3360	3361	3379	3387	3388	3390	3391	3392
3415	3416	3417	3418	3419	3420	3422	3423	3424	3425	3427	3428	3430
3431	3440	3441	3454	3455	3475	3476	3481	3483	3486	3487	3496	3497
3498	3499	3502	3503	3505	3506	3525	3532	3533	3537	3538	3543	3544
3546	3549	3550	3552	3553	3560	3561	3563	3564	3565	3566	3586	3587

	3592#	3594#	3596#	3597#	3600#	3601#	3604#	3605#	3607#	3608#	3609#	3622#	3623#
	3642#	3645#	3653#	3654#	3658#	3659#	3664#	3665#	3668#	3669#	3676#	3678#	3679#
	3680#	3690#	3692#	3693#	3697#	3698#	3714#	3715#	3717#	3720#	3721#	3733#	3734#
	3740#	3743#	3751#	3752#	3756#	3757#	3767#	3768#	3772#	3773#	3776#	3778#	3779#
	3780#	3790#	3793#	3794#	3834#	3835#	3837#	3838#	3839#	3840#	3843#	3844#	3845#
	3846#	3849#	3850#	3852#	3853#	3855#	3856#	3861#	3862#	3865#	3866#	3868#	3869#
	3875#	3876#	3884#	3885#	3886#	3887#	3891#	3910#	3912#	3914#	3915#	3917#	3918#
	3924#	3925#	3926#	3943#	3945#	3947#	3949#	3960#	3961#	3962#	3963#	3964#	3967#
	3968#	3969#	3973#	3974#	3980#	3981#	3983#	4008#	4009#	4011#	4012#	4014#	4015#
	4019#	4020#	4029#	4095#	4096#	4097#	4098#	4099#	4100#	4111#	4112#	4113#	4114#
	4116#	4117#	4119#	4126#	4127#	4128#	4131#	4132#	4134#	4144#	4145#	4146#	4147#
	4159#	4170#	4192#	4193#	4194#	4198#	4204#	4207#	4208#	4209#	4210#	4211#	4212#
	4213#	4215#	4216#	4220#	4238#	4239#	4244#	4247#	4252#	4255#	4256#	4257#	4259#
	4260#	4265#	4281#	4282#	4321#	4343#	4344#	4345#	4346#	4352#	4353#	4354#	4356#
	4357#	4358#	4359#	4360#	4361#	4362#	4363#	4364#	4365#	4367#	4368#	4369#	4370#
	4371#	4372#	4373#	4374#	4375#	4377#	4378#	4381#	4382#	4383#	4384#	4392#	
\$TESTN 001200	1012#	1185#	1246#	1322#	1399#	1487#	1546#	1633#	1711#	1789#	1870#	1904#	1938#
	1972#	2023#	2111#	2199#	2287#	2377#	2475#	2578#	2660#	2743#	2802#	2860#	2980#
\$TIMES 001160	3126#	3288#	3411#	3619#	3708#	3810#	4005#	4293#	4863#				
	998#	1097#	1184#	1245#	1321#	1398#	1486#	1545#	1551#	1632#	1710#	1788#	1869#
	1903#	1937#	1971#	1980#	2022#	2030#	2110#	2118#	2198#	2206#	2286#	2294#	2376#
	2384#	2474#	2482#	2577#	2585#	2659#	2742#	2801#	2859#	2902#	2929#	2979#	3022#
	3042#	3090#	3125#	3131#	3287#	3410#	3543#	3618#	3707#	3714#	3809#	4004#	4045#
	4409#	4851#	4858#	4861#	4871#								
\$TKB 001146	991#	4561#	4572#	4589#	4649#								
\$TKS 001144	990#	4561#	4570#	4586#	4610#	4641#	4647#						
\$TN = 000042	3#	680#	1177#	1184#	1185#	1239#	1245#	1246#	1317#	1321#	1322#	1394#	1398#
	1399#	1476#	1486#	1487#	1540#	1545#	1546#	1552#	1626#	1632#	1633#	1704#	1710#
	1711#	1782#	1788#	1789#	1865#	1869#	1870#	1899#	1903#	1904#	1933#	1937#	1938#
	1967#	1971#	1972#	1981#	2017#	2022#	2023#	2031#	2105#	2110#	2111#	2119#	2193#
	2198#	2199#	2207#	2281#	2286#	2287#	2295#	2369#	2376#	2377#	2385#	2469#	2474#
	2475#	2483#	2567#	2573#	2577#	2578#	2586#	2653#	2659#	2660#	2736#	2742#	2743#
	2797#	2801#	2802#	2853#	2859#	2860#	2903#	2930#	2967#	2975#	2979#	2980#	3023#
	3043#	3061#	3091#	3106#	3118#	3125#	3126#	3132#	3237#	3278#	3287#	3288#	3403#
	3410#	3411#	3544#	3613#	3618#	3619#	3703#	3707#	3708#	3715#	3804#	3809#	3810#
	3895#	3998#	4004#	4005#	4033#	4041#	4045#						
\$TPB 001152	993#	4547#	4558#										
\$TPFLG 001157	997#	4496#	4558#										
\$TPS 001150	992#	4545#	4558#										
\$TRAP 015064	1092#	5024#											
\$TRAP2 015106	5035#	5046#											
\$TRAP = 000012	5039#	5048#	5049#	5050#	5051#	5052#	5053#	5054#	5055#	5056#	5057#	5058#	
\$TRPAD 015120	5029#	5046#											
\$TSKO = 000241	1199#	1222#	1255#	1259#	1271#	1275#	1288#	1292#	1307#	1311#	1331#	1335#	1347#
	1351#	1364#	1368#	1383#	1387#	1413#	1417#	1429#	1433#	1446#	1450#	1465#	1469#
	1497#	1501#	1513#	1517#	1529#	1533#	1550#	1554#	1563#	1567#	1579#	1583#	1596#
	1600#	1615#	1619#	1641#	1645#	1657#	1661#	1674#	1678#	1693#	1697#	1719#	1723#
	1735#	1739#	1752#	1756#	1771#	1775#	1797#	1801#	1813#	1817#	1830#	1834#	1849#
	1853#	1881#	1890#	1915#	1924#	1948#	1959#	1978#	1983#	1993#	2006#	2028#	2033#
	2051#	2057#	2071#	2077#	2091#	2097#	2116#	2121#	2139#	2145#	2159#	2165#	2179#
	2185#	2204#	2209#	2227#	2233#	2247#	2253#	2267#	2273#	2292#	2297#	2315#	2321#
	2335#	2341#	2355#	2361#	2382#	2387#	2414#	2418#	2432#	2436#	2441#	2445#	2458#
	2462#	2480#	2485#	2513#	2517#	2537#	2541#	2560#	2564#	2583#	2588#	2620#	2624#
	2642#	2646#	2687#	2691#	2709#	2713#	2728#	2732#	2771#	2775#	2789#	2793#	2829#
	2833#	2846#	2850#	2876#	2889#	2897#	2905#	2914#	2936#	2943#	2948#	2960#	2964#
	3016#	3025#	3034#	3045#	3055#	3063#	3083#	3093#	3098#	3103#	3130#	3134#	3146#

MAINDEC-11-DVDVA-A  
DVDVA.A.P11

MACY11 27(1006)  
06-MAY-77 15:29

16-MAY-77 10:36 PAGE 142  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0139

	3148	3151#	3227	3333#	3350	3375#	3379	3466#	3483	3521#	3525	3541#	3546
	3577#	3594	3640#	3642	3645#	3693	3712#	3717	3738#	3740	3743#	3794	3874#
	3875	3882#	3891	3922#	3926	3954#	3969	3978#	3983	4017#	4019	4025#	4029
	4106#	4146	4156#	4159	4202#	4204	4207#	4213	4242#	4244	4247#	4260	4336#
	4377												
STSK1 = 000245	1210#	1214	2879#	2883	2884#	2887	2916#	2936	3151#	3226	3337#	3342	3343#
	3348	3470#	3476	3477#	3481	3581#	3587	3588#	3592	3645#	3692	3743#	3793
STSK2 = 000244	3958#	3964	4106#	4131	4146	4207#	4212	4247#	4259	4339#	4365	4366#	4375
	2920#	2934	3169#	3189	3195#	3198	3199#	3218	3651#	3653	3662#	3664	3685#
	3690	3749#	3751	3765#	3767	3785#	3790	4110#	4114	4115#	4118	4124	4128
	4250#	4252	4255#	4257	4342#	4346	4347#	4354					
STSK3 = 000233	2923#	2932	3171#	3189	3205#	3208	3209#	3216	4255#	4256			
STSK4 = 000142	3174#	3179	3180#	3186									
STSTM 001004	957#												
STSTM 001102	970#	4408#	4772	4808	4813	4840	4862#	4863	4868	4872			
STTYIN 013420	4668	4669	4686	4690#									
STYPBN= ***** U	5052												
STYPOS 014412	4884#	5051											
STYPE 012426	4496#	4728	5039	5047									
STYPEC 012640	4526	4533	4540	4545#	4546	4612							
STYPEX 012706	4551	4553	4556#										
STYPOC 014662	4969#	5048											
STYPON 014676	4968	4971#	5050										
STYPOS 014636	4964#	5049											
SUNIT 001206	1015#												
SUNITM 001010	959#												
SUSWR 001220	1022#	1548	1976	2026	2114	2202	2290	2380	2478	2581	3128	3229	3539
	3710	4194											
SVECT1 001244	1047#	4362											
SVECT2 001246	1048#												
SXTSTR 014144	4827#												
SYESNO= 000001	1163#	1164#	1166#	1167#	1169#	1170#	1172#	1173#	1217#	1218#	1220#	1221#	1229#
	1230#	1266#	1267#	1283#	1284#	1300#	1301#	1342#	1343#	1359#	1360#	1376#	1377#
	1424#	1425#	1441#	1442#	1458#	1459#	1492#	1493#	1509#	1510#	1524#	1525#	1574#
	1575#	1591#	1592#	1608#	1609#	1652#	1653#	1669#	1670#	1686#	1687#	1730#	1731#
	1747#	1748#	1764#	1765#	1808#	1809#	1825#	1826#	1842#	1843#	2046#	2047#	2066#
	2067#	2086#	2087#	2134#	2135#	2154#	2155#	2174#	2175#	2222#	2223#	2224#	2243#
	2262#	2263#	2310#	2311#	2330#	2331#	2350#	2351#	2400#	2401#	2428#	2429#	2454#
	2455#	2499#	2500#	2527#	2528#	2550#	2551#	2600#	2601#	2610#	2611#	2632#	2633#
	2747#	2748#	2806#	2807#	2863#	2864#	2885#	2886#	3141#	3142#	3147#	3148#	3182#
	3183#	3184#	3185#	3230#	3233#	3298#	3299#	3310#	3311#	3320#	3321#	3360#	3361#
	3389#	3390#	3427#	3428#	3430#	3431#	3454#	3455#	3486#	3487#	3498#	3499#	3505#
	3506#	3532#	3533#	3552#	3553#	3560#	3561#	3563#	3564#	3565#	3566#	3596#	3597#
	3606#	3607#	3622#	3623#	3641#	3642#	3677#	3678#	3679#	3680#	3697#	3698#	3720#
	3721#	3739#	3740#	3777#	3778#	3779#	3780#	3861#	3862#	3865#	3866#	3868#	3869#
	3911#	3912#	3917#	3918#	3924#	3925#	3944#	3945#	3948#	3949#	3967#	3968#	3973#
	3974#	3980#	3981#	4008#	4009#	4011#	4012#	4144#	4145#	4195#	4198#	4203#	4204#
	4208#	4209#	4210#	4211#	4215#	4216#	4243#	4244#	4251#	4252#	4281#	4282#	4369#
	4370#	4371#	4372#	4373#	4374#	4383#	4384#						
	4071#	4178#	4227#	4288#	4326#								
SSARGC= 000000	1208#	1253#	1269#	1286#	1305#	1329#	1345#	1362#	1381#	1411#	1427#	1444#	1463#
SSBYTE= 000403	1495#	1511#	1527#	1548#	1561#	1577#	1594#	1613#	1639#	1655#	1672#	1691#	1717#
	1733#	1750#	1769#	1795#	1811#	1828#	1847#	1879#	1913#	1946#	1976#	1991#	2026#
	2049#	2069#	2089#	2114#	2137#	2157#	2177#	2202#	2225#	2245#	2265#	2290#	2313#
	2333#	2353#	2380#	2412#	2430#	2439#	2456#	2478#	2511#	2535#	2558#	2581#	2618#
	2640#	2707#	276 #	2844#	2877#	2895#	2914#	2918#	2921#	2941#	2958#	3014#	3032#

K11

MAINDEC-11-DVDVA-A  
DVDVA.P11

MACY11 27(1006)  
06-MAY-77 15:29

16-MAY-77 10:36 PAGE 143  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0140

SSDST = 000067  
SSFLAG = 000001

SSFROM = 000000

SSGET4 = 000000  
SSL0C = 012100

SSL0CN = 000000  
SSRE TU = 000000  
SSPTN1 = 000237  
SSPTN2 = 000240

3053#	3081#	3096#	3128#	3169#	3172#	3193#	3201#	3203#	3331#	3335#	3373#	3464#
3468#	3519#	3539#	3575#	3579#	3683#	3710#	3783#	3880#	3920#	3952#	3956#	3976#
4023#	4108#	4122#	4154#	4337#	4340#							
3230#	4195#											
1208#	1210#	1214#	1253#	1255#	1259#	1269#	1271#	1275#	1286#	1288#	1292#	1305#
1307#	1311#	1329#	1331#	1335#	1345#	1347#	1351#	1362#	1364#	1368#	1381#	1383#
1387#	1411#	1413#	1417#	1427#	1429#	1433#	1444#	1446#	1450#	1463#	1465#	1469#
1495#	1497#	1501#	1511#	1513#	1517#	1527#	1529#	1533#	1548#	1550#	1554#	1561#
1563#	1567#	1577#	1579#	1583#	1594#	1596#	1600#	1613#	1615#	1619#	1639#	1641#
1645#	1655#	1657#	1661#	1672#	1674#	1678#	1691#	1693#	1697#	1717#	1719#	1723#
1733#	1735#	1739#	1750#	1752#	1756#	1769#	1771#	1775#	1795#	1797#	1801#	1811#
1813#	1817#	1828#	1830#	1834#	1847#	1849#	1853#	1879#	1881#	1890#	1913#	1915#
1924#	1946#	1948#	1959#	1976#	1978#	1983#	1991#	1993#	2006#	2026#	2028#	2033#
2049#	2051#	2057#	2069#	2071#	2077#	2089#	2091#	2097#	2114#	2116#	2121#	2137#
2139#	2145#	2157#	2159#	2165#	2177#	2179#	2185#	2202#	2204#	2209#	2225#	2227#
2233#	2245#	2247#	2253#	2265#	2267#	2273#	2290#	2292#	2297#	2313#	2315#	2321#
2333#	2335#	2341#	2353#	2355#	2361#	2380#	2382#	2387#	2412#	2414#	2418#	2430#
2432#	2436#	2439#	2441#	2445#	2456#	2458#	2462#	2478#	2480#	2485#	2511#	2513#
2517#	2535#	2537#	2541#	2558#	2560#	2564#	2581#	2583#	2588#	2618#	2620#	2624#
2640#	2642#	2646#	2686#	2691#	2707#	2709#	2713#	2727#	2732#	2770#	2775#	2787#
2789#	2793#	2828#	2833#	2844#	2846#	2850#	2877#	2879#	2887#	2895#	2897#	2905#
2913#	2914#	2916#	2918#	2920#	2921#	2923#	2932#	2934#	2941#	2943#	2948#	2958#
2960#	2964#	3014#	3016#	3025#	3032#	3034#	3045#	3053#	3055#	3063#	3081#	3083#
3093#	3096#	3098#	3103#	3128#	3130#	3134#	3168#	3169#	3171#	3172#	3174#	3186#
3193#	3195#	3201#	3203#	3205#	3216#	3218#	3331#	3333#	3335#	3337#	3348#	3350#
3373#	3375#	3379#	3464#	3466#	3468#	3470#	3481#	3483#	3519#	3521#	3525#	3539#
3541#	3546#	3575#	3577#	3579#	3581#	3592#	3594#	3683#	3685#	3690#	3710#	3712#
3717#	3783#	3785#	3790#	3880#	3882#	3891#	3920#	3922#	3926#	3952#	3954#	3956#
3958#	3964#	3969#	3976#	3978#	3983#	4023#	4025#	4029#	4108#	4110#	4118#	4122#
4124#	4128#	4154#	4156#	4159#	4337#	4339#	4340#	4342#	4354#	4375#		
1148#	2403#	2502#	2530#	2553#	2613#	2635#	2676#	2719#	2760#	2819#	2993#	3003#
3075#	3324#	3368#	3445#	3457#	3513#	3568#	3632#	3729#	3759#	3814#	4139#	4348#
4424#												
1209#	1210#	1223#	1224#	1254#	1255#	1270#	1271#	1287#	1288#	1306#	1307#	1330#
1331#	1346#	1347#	1363#	1364#	1382#	1383#	1412#	1413#	1428#	1429#	1445#	1446#
1464#	1465#	1496#	1497#	1512#	1513#	1528#	1529#	1549#	1550#	1562#	1563#	1578#
1579#	1595#	1596#	1614#	1615#	1640#	1641#	1656#	1657#	1673#	1674#	1692#	1693#
1718#	1719#	1734#	1735#	1751#	1752#	1770#	1771#	1796#	1797#	1812#	1813#	1829#
1830#	1848#	1849#	1880#	1881#	1914#	1915#	1947#	1948#	1977#	1978#	1992#	1993#
2027#	2028#	2050#	2051#	2070#	2071#	2090#	2091#	2115#	2116#	2138#	2139#	2158#
2159#	2178#	2179#	2203#	2204#	2226#	2227#	2246#	2247#	2266#	2267#	2291#	2292#
2314#	2315#	2334#	2335#	2354#	2355#	2381#	2382#	2413#	2414#	2431#	2432#	2440#
2441#	2457#	2458#	2479#	2480#	2512#	2513#	2536#	2537#	2559#	2560#	2582#	2583#
2619#	2620#	2641#	2642#	2686#	2687#	2708#	2709#	2727#	2728#	2770#	2771#	2788#
2789#	2828#	2829#	2845#	2846#	2878#	2879#	2892#	2893#	2896#	2897#	2915#	2916#
2919#	2920#	2922#	2923#	2942#	2943#	2959#	2960#	3015#	3016#	3033#	3034#	3054#
3055#	3082#	3083#	3097#	3098#	3129#	3130#	3170#	3171#	3173#	3174#	3194#	3195#
3202#	3203#	3204#	3205#	3332#	3333#	3336#	3337#	3374#	3375#	3465#	3466#	3469#
3470#	3520#	3521#	3540#	3541#	3576#	3577#	3580#	3581#	3654#	3655#	3665#	3666#
3684#	3685#	3711#	3712#	3752#	3753#	3768#	3769#	3784#	3785#	3876#	3877#	3881#
3882#	3921#	3922#	3953#	3954#	3957#	3958#	3977#	3978#	4020#	4021#	4024#	4025#
4109#	4110#	4123#	4124#	4155#	4156#	4338#	4339#	4341#	4342#	4378#	4379#	
4071#	4173#	4178#	4222#	4227#	4267#	4288#	4323#	4326#	4394#			
4071#	4157#	4161#	4171#	4178#	4221#	4227#	4266#	4288#	4322#	4326#	4393#	
4071#	4157#	4170#	4178#	4217#	4220#	4227#	4265#	4289#	4321#	4326#	4385#	4392#
4071#	4162#	4172#	4178#	4221#	4227#	4266#	4288#	4322#	4326#	4393#		





\$152	007410	3374	3379#
\$153	007640	3465	3483#
\$154	007636	3469	3476#
\$155	007640	3475	3481#
\$156	007732	3520	3525#
\$157	007770	3540	3546#
\$16	002662	1464	1469#
\$160	010076	3576	3594#
\$161	010074	3580	3587#
\$162	010076	3586	3592#
\$163	010214	3639	3642#
\$164	010212	3640#	3692
\$165	010270	3644	3693#
\$166	010222	3650#	3654
\$167	010236	3661#	3665
\$17	002726	1496	1501#
\$170	010266	3684	3690#
\$171	010334	3711	3717#
\$172	010400	3737	3740#
\$173	010376	3738#	3793
\$174	010470	3742	3794#
\$175	010406	3748#	3752
\$176	010436	3764#	3768
\$177	010466	3784	3790#
\$2	002126	1209	1214#
\$20	002754	1512	1517#
\$200	010610	3873#	3876
\$201	010636	3881	3891#
\$202	010702	3921	3926#
\$203	010760	3953	3969#
\$204	010754	3957	3964#
\$205	010776	3977	3983#
\$206	011044	4016#	4020
\$207	011064	4024	4029#
\$21	003002	1528	1533#
\$210	011410	4157	4170#
\$211	011412	4162	4172#
\$212	011260	4105#	4146
\$213	011364	4132	4134
\$214	011300	4109	4114#
\$215	011306	4113	4118#
\$216	011324	4123	4128#
\$217	011376	4155	4159#
\$22	003040	1549	1554#
\$220	011512	4217	4220#
\$221	011512	4221#	
\$222	011456	4201	4204#
\$223	011452	4202#	4212
\$224	011502	4206	4213#
\$225	011572	4265#	
\$226	011572	4266#	
\$227	011540	4241	4244#
\$23	003060	1562	1567#
\$230	011534	4242#	4259
\$231	011564	4246	4260#
\$232	011552	4249	4252#

4147#



\$233	011550	4250#	4256
\$234	011562	4254	4257#
\$235	011764	4321#	
\$236	011764	4322#	
\$237	012124	4385	4392#
\$24	003106	1578	1583#
\$240	012124	4393#	
\$241	011766	4335#	4378
\$242	012052	4338	4365#
\$243	012012	4341	4346#
\$244	012020	4345	4354#
\$245	012072	4364	4375#
\$25	003134	1595	1600#
\$26	003164	1614	1619#
\$27	003222	1640	1645#
\$3	002240	1254	1259#
\$30	003250	1656	1661#
\$31	003276	1673	1678#
\$32	003326	1692	1697#
\$33	003364	1718	1723#
\$34	003412	1734	1739#
\$35	003440	1751	1756#
\$36	003470	1770	1775#
\$37	003526	1796	1801#
\$4	002266	1270	1275#
\$40	003554	1812	1817#
\$40CAT=	***** U	4782	4824
\$41	003602	1829	1834#
\$42	003632	1848	1853#
\$43	003672	1880	1890#
\$44	003732	1914	1924#
\$45	003772	1947	1959#
\$46	004030	1977	1983#
\$47	004052	1992	2006#
\$5	002314	1287	1292#
\$50	004110	2027	2033#
\$51	004136	2050	2057#
\$52	004164	2070	2077#
\$53	004212	2090	2097#
\$54	004250	2115	2121#
\$55	004276	2138	2145#
\$56	004324	2158	2165#
\$57	004352	2178	2185#
\$6	002344	1306	1311#
\$60	004410	2203	2209#
\$61	004436	2226	2233#
\$62	004464	2246	2253#
\$63	004512	2266	2273#
\$64	004550	2291	2297#
\$65	004576	2314	2321#
\$66	004624	2334	2341#
\$67	004652	2354	2361#
\$7	002402	1330	1335#
\$70	004710	2381	2387#
\$71	004770	2413	2418#
\$72	005016	2431	2436#



BEGIN	1#														
BGNHRO	3#														
BGNHM	3#														
BGNINI	3#														
BGNMOD	3#	4067													
BGNMSG	3#														
BGNSFT	3#														
BGNSRV	3#	3905	3938	4271											
BGNSUB	3#	1199	1249	1263	1279	1296	1325	1339	1355	1372	1407	1421	1437	1454	1489
	1505	1521	1557	1571	1587	1604	1635	1649	1665	1681	1713	1727	1743	1760	1791
	1805	1821	1838	1876	1909	1943	1987	2041	2061	2081	2129	2149	2169	2217	2237
	2257	2305	2325	2345	2396	2423	2449	2494	2521	2545	2605	2628	2662	2694	2749
	2779	2807	2982	3029	3048	3066	3306	3353	3421	3495	3536				
BGNSW	3#														
BRESET	3#	1175	1302	1378	1460	1610	1688	1766	1844	1887	1921	1956	2003	2783	3380
	3527	3857	4030												
CALL	1#	1147	2403	2502	2530	2553	2613	2635	2675	2718	2759	2818	2953	3003	3075
	3324	3368	3444	3457	3513	3568	3631	3728	3759	3813	4139	4347			
CASE	1#														
CKLOOP	3#	1854	1892	1926	1960	2008									
CLRVEC	3#	1224	3384	3601											
COMMEN	1#	802#													
DECR	1#														
DECRU	1#														
DEFAULT	1#														
DEVREG	3#														
DEVTYP	3#														
DISPAT	3#														
ELSE	1#	2881	3177	3196	3206	3340	3474	3585	4112	4344	4363				
END	1#														
ENDCLN	3#														
ENDCOM	1#	802#													
ENDDC	1#														
ENDDO	1#	2935	3188												
ENDHRO	3#														
ENDHM	3#														
ENDIF	1#	1213	1258	1274	1291	1310	1334	1350	1367	1386	1416	1432	1449	1468	1500
	1516	1532	1553	1566	1582	1599	1618	1644	1660	1677	1696	1722	1738	1755	1774
	1800	1816	1833	1852	1889	1923	1958	1982	2005	2032	2056	2076	2096	2120	2144
	2164	2184	2208	2232	2252	2272	2296	2320	2340	2360	2386	2417	2435	2444	2461
	2484	2516	2540	2563	2587	2623	2645	2690	2712	2731	2774	2792	2832	2849	2886
	2904	2931	2933	2947	2963	3024	3044	3062	3092	3102	3133	3185	3215	3217	3347
	3349	3378	3480	3482	3524	3545	3591	3593	3689	3716	3789	3890	3925	3963	3968
	3982	4028	4117	4127	4158	4353	4374								
ENDINC	1#	3225	3691	3792	4211	4255	4258								
ENDINI	3#														
ENDLOO	1#	4145													
ENDMOD	3#														
ENDMSG	3#														
ENDRTN	1#	4169	4219	4264	4320	4391									
ENDSEL	1#														
ENDSFT	3#														
ENDSRV	3#	3932	3990	4282											
ENDSUB	3#	1215	1260	1276	1293	1312	1336	1352	1369	1388	1418	1434	1451	1470	1502
	1518	1534	1568	1584	1601	1620	1646	1662	1679	1698	1724	1740	1757	1776	1802
	1818	1835	1855	1893	1927	1961	2009	2058	2078	2098	2146	2166	2186	2234	2254

	2274	2322	2342	2362	2420	2446	2463	2518	2542	2565	2625	2647	2692	2733	2777
	2794	2834	3026	3046	3064	3104	3351	3382	3487	3533	3597				
ENDSW	3#														
ENDTST	3#	1235	1313	1389	1471	1535	1621	1699	1777	1856	1894	1928	1962	2010	2099
	2187	2275	2363	2464	2568	2648	2734	2795	2851	2971	3114	3273	3395	3611	3698
	3798	3993	4039												
EQUALS	3#														
ERROF	3#	1211													
ERRHRD	3#	1256	1272	1289	1308	1332	1348	1365	1384	1414	1430	1447	1466	1498	1514
	1530	1564	1580	1597	1616	1642	1658	1675	1694	1720	1736	1753	1772	1798	1814
	1831	1850	1884	1918	1949	1995	2052	2072	2092	2140	2160	2180	2228	2248	2268
	2316	2336	2356	2415	2433	2442	2459	2514	2538	2561	2621	2643	2688	2710	2729
	2772	2790	2830	2847	2899	2925	2944	2961	3017	3036	3057	3084	3099	3213	3338
	3345	3376	3471	3478	3522	3582	3589	3686	3786	3888	4026				
ERROR	696#	1212	1257	1273	1290	1309	1333	1349	1366	1385	1415	1431	1448	1467	1499
	1515	1531	1565	1581	1598	1617	1643	1659	1676	1695	1721	1737	1754	1773	1799
	1815	1832	1851	1885	1919	1950	1996	2053	2073	2093	2141	2161	2161	2229	2249
	2269	2317	2337	2357	2416	2434	2443	2460	2515	2539	2562	2622	2644	2689	2711
	2730	2773	2791	2831	2848	2900	2926	2945	2962	3018	3037	3058	3085	3100	3214
	3339	3346	3377	3472	3479	3523	3583	3590	3687	3787	3889	4027			
ESCAPE	1#	802#													
EXIF	1#														
EXIFB	1#	4130													
EXIT	3#	1550	1979	2029	2117	2205	2293	2383	2481	2566	2584	2901	2928	2966	3021
	3041	3060	3089	3105	3130	3236	3542	3713	3894	4032					
GETPRI	1#	802#													
GETSWR	1#	802#	1130#												
GPHARD	3#														
GPRMA	3#														
GPRMD	3#														
GPRML	3#														
HEADER	3#														
IF	1#	1207	1252	1268	1285	1304	1328	1344	1361	1380	1410	1426	1443	1462	1494
	1510	1526	1547	1560	1576	1593	1612	1638	1654	1671	1690	1716	1732	1749	1768
	1794	1810	1827	1846	1878	1912	1945	1975	1990	2025	2048	2068	2088	2113	2136
	2156	2176	2201	2224	2244	2264	2289	2312	2332	2352	2379	2411	2429	2438	2455
	2477	2510	2534	2557	2580	2617	2639	2706	2786	2843	2876	2894	2917	2920	2940
	2957	3013	3031	3052	3080	3095	3127	3171	3192	3200	3330	3334	3372	3463	3467
	3518	3538	3574	3578	3682	3709	3782	3879	3919	3951	3955	3975	4107	4336	4339
IFB	1#	4022	4121	4153											
IFCOND	1#														
IF.ERR	1#	2685	2726	2769	2827										
IF.NO.	1#														
INCR	1#	3143	3637	3735	4199	4247									
INCRU	1#	4239													
INLINE	1#														
LASTAD	3#														
LEAVE	1#														
LET	1#	1144	1151	1154	1156	1159	1161	1164	1167	1170	1173	1186	1190	1191	1192
	1193	1194	1195	1200	1202	1216	1218	1227	1226	1230	1231	1250	1264	1265	1280
	1282	1297	1299	1326	1340	1341	1356	1358	1373	1375	1408	1422	1423	1438	1440
	1455	1457	1490	1491	1506	1508	1522	1523	1551	1558	1572	1573	1588	1590	1605
	1607	1636	1650	1651	1666	1668	1682	1685	1714	1726	1729	1744	1746	1761	1763
	1792	1806	1807	1822	1824	1839	1841	1877	1910	1944	1980	1988	2030	2042	2045
	2062	2065	2082	2085	2118	2130	2133	2150	2153	2170	2173	2206	2218	2221	2238
	2241	2258	2261	2294	2306	2309	2326	2329	2346	2349	2384	2397	2399	2408	2424





	3464	3468	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956	3976	4023
\$CHK1	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	4108	4122	4154	4337	4340										
	1203	1227	1230	1231	1250	1264	1280	1297	1326	1340	1356	1373	1408	1422	1438
	1455	1490	1506	1522	1551	1558	1572	1588	1605	1636	1650	1666	1682	1714	1728
	1744	1761	1792	1806	1822	1839	1877	1910	1944	1980	1988	2030	2042	2062	2082
	2118	2130	2150	2170	2206	2218	2238	2258	2294	2306	2326	2346	2384	2397	2409
	2424	2450	2482	2495	2508	2522	2546	2585	2602	2606	2629	2663	2671	2695	2702
	2750	2754	2780	2808	2814	2841	2865	2867	2873	2880	2902	2929	2955	2983	2990
	3000	3010	3022	3030	3042	3049	3067	3072	3090	3131	3137	3139	3144	3147	3152
	3155	3158	3161	3164	3166	316	3221	3223	3234	3291	3295	3301	3302	3303	3304
	3305	3307	3354	3357	3387	3390	3391	3415	3416	3417	3418	3419	3422	3424	3440
	3496	3502	3537	3543	3549	3600	3604	3607	3608	3638	3641	3658	3668	3714	3733
	3736	3739	3756	3772	3834	3837	3839	3843	3845	3849	3852	3855	3884	3886	3914
	3960	3962	4014	4095	4097	4099	4111	4116	4126	4192	4200	4203	4215	4238	4240
	4243	4248	4251	4343	4352	4356	4358	4360	4362	4367	4381				
\$CKOP2	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	1441	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
	1825	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
	2428	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3182	3184
	3229	3298	3310	3320	3360	3388	3427	3430	3454	3486	3498	3505	3532	3552	3560
	3563	3565	3596	3605	3622	3676	3679	3697	3720	3776	3779	3861	3865	3868	3910
	3917	3924	3943	3947	3967	3973	3980	4008	4011	4144	4194	4208	4210	4281	4369
	4371	4373	4383												
\$CKR6	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$CMND	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750	1769
	1795	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114	2137
	2157	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439	2456
	2478	2511	2535	2558	2581	2618	2640	2707	2787	2844	2877	2895	2914	2918	2921
	2941	2958	3014	3032	3053	3081	3096	3128	3169	3172	3193	3201	3203	3331	3335
	3373	3464	3468	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956	3976
	4023	4108	4122	4154	4337	4340									
\$COMP2	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750	1769
	1795	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114	2137
	2157	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439	2456
	2478	2511	2535	2558	2581	2618	2640	2686	2707	2727	2770	2787	2828	2844	2877
	2895	2914	2918	2921	2941	2958	3014	3032	3053	3081	3096	3128	3144	3169	3172
	3193	3201	3331	3335	3373	3464	3468	3519	3539	3575	3579	3638	3683	3710	3736
	3783	3880	3920	3952	3956	3976	4023	4108	4122	4154	4200	4240	4248	4337	4340
\$COUNT	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	3324	3368	3445	3457	3513	3568	3632	3729	3759	3814	4139	4348			
\$DOO	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$ELSE	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$ERRMS	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$EXIFA	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$EXIFO	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$EXIF2	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$EXIF3	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
\$GENBR	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	1496	1512	1528	1549	1562	1578	1595	1614	1640	1656	1673	1692	1718	1734	1751
	1770	1796	1812	1829	1848	1880	1914	1947	1977	1992	2027	2050	2070	2090	2115
	2138	2158	2178	2203	2226	2246	2266	2291	2314	2334	2354	2381	2413	2431	2440
	2457	2479	2512	2536	2559	2582	2619	2641	2686	2708	2727	2770	2788	2828	2845
	2878	2882	2890	2892	2896	2915	2919	2922	2936	2942	2959	3015	3033	3054	3082

	3097	3129	3145	3150	3170	3173	3178	3189	3194	3197	3202	3204	3207	3226	3332
	3336	3341	3374	3465	3469	3475	3520	3540	3576	3580	3586	3639	3644	3654	3665
	3684	3692	3711	3737	3742	3752	3768	3784	3793	3876	3881	3921	3953	3957	3977
	4020	4024	4109	4113	4123	4132	4134	4146	4155	4157	4162	4201	4206	4212	4217
\$GENTA	1#	4241	4246	4249	4254	4256	4259	4338	4341	4345	4364	4378	4385		
	1501	1198	1214	1259	1275	1292	1311	1335	1351	1368	1387	1417	1433	1450	1469
	1775	1517	1533	1554	1567	1583	1600	1619	1645	1661	1678	1697	1723	1739	1756
	2145	1801	1817	1834	1853	1890	1924	1959	1983	2006	2033	2057	2077	2097	2121
	2462	2165	2185	2209	2233	2253	2273	2297	2321	2341	2351	2387	2418	2436	2445
	2875	2485	2517	2541	2564	2588	2624	2646	2691	2713	2732	2775	2793	2833	2850
	3103	2883	2887	2933	2905	2913	2932	2934	2937	2948	2964	3025	3045	3063	3093
	3350	3134	3146	3148	3168	3179	3186	3190	3198	3208	3216	3218	3227	3342	3348
	3693	3379	3476	3481	3483	3525	3546	3587	3592	3594	3640	3642	3650	3661	3690
	4029	3717	3738	3740	3748	3764	3790	3794	3873	3891	3926	3964	3969	3983	4016
	4244	4105	4114	4118	4128	4147	4159	4170	4172	4202	4204	4213	4220	4221	4242
	4393	4250	4252	4257	4260	4265	4266	4321	4322	4335	4346	4354	4365	4375	4392
\$IF	1#	1208	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	1463	1495
	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750	1769
	1795	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114	2137
	2157	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439	2456
	2478	2511	2535	2558	2581	2618	2640	2707	2787	2844	2877	2895	2918	2921	2941
	2958	3014	3032	3053	3081	3096	3128	3172	3193	3201	3331	3335	3373	3464	3468
	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956	3976	4023	4108	4122
	4154	4337	4340												
\$IFCOD	1#	1208	1222	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	1463
	1495	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750
	1769	1795	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114
	2137	2157	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439
	2456	2478	2511	2535	2558	2581	2618	2640	2707	2787	2844	2877	2895	2918	2941
	2918	2921	2941	2958	3014	3032	3053	3081	3096	3128	3169	3172	3193	3201	3203
	3331	3335	3373	3464	3468	3519	3539	3575	3579	3653	3664	3683	3710	3751	3767
	3783	3875	3880	3920	3952	3956	3976	4019	4023	4108	4122	4154	4337	4340	4377
\$IFCON	1#	2686	2727	2770	2828										
\$IFOPR	1#	1209	1223	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1496	1512	1528	1549	1562	1578	1595	1614	1640	1656	1673	1692	1718	1734	1751
	1770	1796	1812	1829	1848	1880	1914	1947	1977	1992	2027	2050	2070	2090	2115
	2138	2158	2178	2203	2226	2246	2266	2291	2314	2334	2354	2381	2413	2431	2440
	2457	2479	2512	2536	2559	2582	2619	2641	2686	2708	2727	2770	2788	2828	2845
	2878	2892	2896	2915	2919	2922	2942	2959	3015	3033	3054	3082	3097	3129	3170
	3173	3194	3202	3204	3332	3336	3374	3465	3469	3520	3540	3576	3580	3654	3665
	3684	3711	3752	3768	3784	3876	3881	3921	3953	3957	3977	4020	4024	4109	4123
	4155	4338	4341	4378											
\$LET	1#	1145	1152	1155	1157	1160	1162	1165	1168	1171	1174	1187	1190	1191	1192
	1193	1194	1196	1200	1203	1217	1219	1227	1228	1230	1231	1250	1264	1266	1280
	1283	1297	1300	1326	1340	1342	1356	1359	1373	1376	1408	1422	1424	1438	1441
	1455	1458	1490	1492	1506	1509	1522	1524	1551	1558	1572	1574	1588	1591	1605
	1608	1636	1650	1652	1666	1669	1682	1686	1714	1728	1730	1744	1747	1761	1764
	1792	1806	1808	1822	1825	1839	1942	1877	1910	1944	1980	1988	2030	2042	2046
	2062	2066	2082	2086	2118	2130	2134	2150	2154	2170	2174	2206	2218	2222	2238
	2242	2258	2262	2294	2306	2310	2326	2330	2346	2350	2384	2397	2400	2409	2424
	2428	2450	2454	2482	2495	2499	2508	2522	2527	2546	2550	2585	2600	2602	2606
	2610	2629	2632	2663	2671	2695	2702	2747	2750	2754	2780	2806	2808	2814	2841
	2863	2865	2867	2873	2880	2885	2902	2929	2955	2983	2990	3000	3010	3022	3030
	3042	3049	3067	3072	3090	3131	3137	3139	3141	3152	3155	3158	3161	3154	3166
	3176	3182	3184	3221	3223	3229	3234	3291	3295	3298	3301	3302	3303	3304	3305



3307	3310	3320	3354	3357	3360	3387	3388	3390	3391	3415	3416	3417	3418	3419		
3422	3424	3427	3430	3440	3454	3486	3496	3498	3502	3505	3532	3537	3543	3549		
3552	3560	3563	3565	3596	3600	3604	3605	3607	3608	3622	3658	3668	3676	3679		
3697	3714	3720	3733	3756	3772	3776	3779	3834	3837	3839	3843	3845	3849	3852		
3855	3861	3865	3868	3884	3886	3910	3914	3917	3924	3943	3947	3960	3962	3967		
3973	3980	4008	4011	4014	4095	4097	4099	4111	4116	4126	4144	4192	4194	4208		
4210	4215	4238	4281	4343	4352	4356	4358	4360	4362	4367	4369	4371	4373	4381		
4383																
\$LPCNT	1#	3144	3638	3736	4200	4240	4248									
\$OPADR	1#	1163	1166	1169	1172	1217	1220	1229	2885	3147	3182	3184	3298	3389	3606	
		3641	3739	3917	3967	3973	4203	4243	4251	4281	4371	4373	4383			
\$OPAND	1#	3230	4195													
\$OPCD1	1#	1163	1166	1169	1172	1217	1220	1229	1266	1283	1300	1342	1359	1376	1424	
		1441	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
		1825	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
		2428	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3147	3182
		3184	3230	3298	3310	3320	3360	3389	3427	3430	3454	3486	3498	3505	3532	3552
		3560	3563	3565	3596	3606	3622	3641	3677	3679	3697	3720	3739	3777	3779	3861
		3865	3868	3911	3917	3924	3944	3948	3967	3973	3980	4008	4011	4144	4195	4203
		4208	4210	4215	4243	4251	4281	4369	4371	4373	4383					
\$OPCD2	1#	4208	4215	4369												
\$OPCOO	1#	1163	1166	1169	1172	1217	1220	1229	1266	1283	1300	1342	1359	1376	1424	
		1441	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
		1825	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
		2428	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3147	3182
		3184	3230	3298	3310	3320	3360	3389	3427	3430	3454	3486	3498	3505	3532	3552
		3560	3563	3565	3596	3606	3622	3641	3677	3679	3697	3720	3739	3777	3779	3861
		3865	3868	3911	3917	3924	3944	3948	3967	3973	3980	4008	4011	4144	4195	4203
		4208	4210	4215	4243	4251	4281	4369	4371	4373	4383					
\$OPCOM	1#	4215														
\$OPDEF	1#	1145	1148	1152	1155	1157	1160	1162	1163	1165	1166	1168	1169	1171	1172	
		1174	1187	1190	1191	1192	1193	1194	1196	1200	1203	1208	1209	1217	1219	1220
		1222	1223	1227	1228	1229	1230	1231	1250	1253	1254	1264	1266	1269	1270	1280
		1283	1286	1287	1297	1300	1305	1306	1326	1329	1330	1340	1342	1345	1346	1356
		1359	1362	1363	1373	1376	1381	1382	1408	1411	1412	1422	1424	1427	1428	1438
		1441	1444	1445	1455	1458	1463	1464	1490	1492	1495	1496	1506	1509	1511	1512
		1522	1524	1527	1528	1548	1549	1551	1558	1561	1562	1572	1574	1577	1578	1588
		1591	1594	1595	1605	1608	1613	1614	1636	1639	1640	1650	1652	1655	1656	1666
		1669	1672	1673	1682	1686	1691	1692	1714	1717	1718	1728	1730	1733	1734	1744
		1747	1750	1751	1761	1764	1769	1770	1792	1795	1796	1806	1808	1811	1812	1822
		1825	1828	1829	1839	1842	1847	1848	1877	1879	1880	1910	1913	1914	1944	1946
		1947	1976	1977	1980	1988	1991	1992	2026	2027	2030	2042	2046	2049	2050	2062
		2066	2069	2070	2082	2086	2089	2090	2114	2115	2118	2130	2134	2137	2138	2150
		2154	2157	2158	2170	2174	2177	2178	2202	2203	2206	2218	2222	2225	2226	2238
		2242	2245	2246	2258	2262	2265	2266	2290	2291	2294	2306	2310	2313	2314	2326
		2330	2333	2334	2346	2350	2353	2354	2380	2381	2384	2397	2400	2403	2404	2405
		2406	2409	2412	2413	2424	2428	2430	2431	2439	2440	2450	2454	2456	2457	2478
		2479	2482	2495	2499	2502	2503	2504	2505	2508	2511	2512	2522	2527	2530	2531
		2532	2533	2535	2536	2546	2550	2553	2554	2555	2556	2558	2559	2581	2582	2585
		2600	2602	2606	2610	2613	2614	2615	2616	2618	2619	2629	2632	2635	2636	2637
		2638	2640	2641	2663	2671	2676	2677	2678	2679	2680	2681	2682	2686	2695	2702
		2707	2708	2719	2720	2721	2722	2723	2724	2725	2727	2747	2750	2754	2760	2761
		2762	2763	2764	2765	2766	2770	2780	2787	2788	2806	2808	2814	2819	2820	2821
		2822	2823	2824	2825	2828	2841	2844	2845	2863	2865	2867	2873	2877	2878	2880
		2882	2885	2889	2890	2891	2892	2895	2896	2902	2914	2915	2918	2919	2921	2922
		2929	2936	2941	2942	2955	2958	2959	2983	2990	2993	2994	2995	2996	3000	3003

3004	3005	3006	3010	3014	3015	3022	3030	3032	3033	3042	3049	3053	3054	3067
3072	3075	3076	3077	3078	3081	3082	3090	3096	3097	3128	3129	3131	3137	3139
3141	3144	3145	3147	3149	3150	3152	3155	3158	3161	3164	3166	3169	3170	3172
3173	3176	3178	3182	3184	3189	3193	3194	3197	3201	3202	3203	3204	3207	3221
3223	3226	3229	3230	3231	3232	3234	3291	3295	3298	3301	3302	3303	3304	3305
3307	3310	3320	3324	3325	3326	3327	3331	3332	3335	3336	3341	3354	3357	3360
3368	3369	3370	3371	3373	3374	3387	3388	3389	3390	3391	3415	3416	3417	3418
3419	3422	3424	3427	3430	3440	3445	3446	3447	3448	3449	3450	3451	3454	3457
3458	3459	3460	3464	3465	3468	3469	3475	3486	3496	3498	3502	3505	3513	3514
3515	3516	3519	3520	3532	3537	3539	3540	3543	3549	3552	3560	3563	3565	3568
3569	3570	3571	3575	3576	3579	3580	3586	3596	3600	3604	3605	3606	3607	3608
3622	3632	3633	3634	3638	3639	3641	3643	3644	3653	3654	3658	3664	3665	3668
3676	3677	3679	3683	3684	3692	3697	3710	3711	3714	3720	3729	3730	3731	3733
3736	3737	3739	3741	3742	3751	3752	3756	3759	3760	3761	3762	3767	3768	3772
3776	3777	3779	3783	3784	3793	3814	3815	3816	3834	3837	3839	3843	3845	3849
3852	3855	3861	3865	3868	3875	3876	3880	3881	3894	3886	3910	3911	3914	3917
3920	3921	3924	3943	3944	3947	3948	3952	3953	3956	3957	3960	3962	3967	3973
3976	3977	3980	4008	4011	4014	4019	4020	4023	4024	4095	4097	4099	4108	4109
4111	4113	4116	4122	4123	4126	4131	4132	4133	4134	4139	4140	4141	4142	4144
4146	4154	4155	4177	4161	4162	4171	4173	4192	4194	4195	4196	4197	4200	4201
4203	4205	4206	4208	4210	4212	4215	4217	4222	4238	4240	4241	4243	4245	4246
4248	4249	4251	4253	4254	4256	4259	4267	4281	4323	4337	4338	4340	4341	4343
4345	4348	4352	4356	4358	4360	4362	4364	4367	4369	4371	4373	4377	4378	4381
4383	4385	4394												

SOPEQU 1#  
SOPMAN 1#  
SOPNEG 1#  
SOPNOR 1#  
SOPNOT 1#

2262	1283	1359	1441	1492	1524	1591	1669	1747	1825	2046	2086	2134	2174	2222
3552	2310	2350	2400	2454	2499	2550	2600	2632	3310	3360	3430	3486	3505	3532
	3560	3596	3677	3679	3697	3720	3777	3779	3911	3924	3944	3948	3980	
	1266	1300	1342	1376	1424	1458	1509	1574	1608	1652	1686	1730	1764	1808
	1842	2066	2154	2242	2330	2428	2527	2610	2747	2806	2863	3141	3320	3454
	3498	3563	3565	3622	3861	3865	3868	4008	4011	4210				

SOPOR 1#  
SOPROT 1#  
SOPRO 1#

1203	1145	1152	1155	1157	1160	1174	1187	1190	1191	1192	1193	1194	1196	1200
1455	1227	1230	1231	1250	1264	1280	1297	1326	1340	1356	1373	1408	1422	1438
1744	1490	1506	1522	1551	1558	1572	1588	1605	1636	1650	1666	1682	1714	1728
2118	1761	1792	1806	1822	1839	1877	1910	1944	1980	1988	2030	2042	2062	2082
2424	2130	2150	2170	2206	2218	2238	2258	2294	2306	2326	2346	2384	2397	2409
2750	2450	2482	2495	2508	2522	2546	2585	2602	2606	2629	2663	2671	2695	2702
3000	2754	2780	2808	2814	2841	2865	2867	2873	2880	2902	2929	2955	2983	2990
3158	3010	3022	3030	3042	3049	3067	3072	3090	3131	3137	3139	3144	3152	3155
3307	3161	3164	3166	3176	3221	3223	3234	3291	3295	3301	3302	3303	3304	3305
3502	3354	3357	3387	3390	3391	3415	3416	3417	3418	3419	3422	3424	3440	3496
3772	3537	3543	3549	3600	3604	3607	3608	3638	3658	3668	3714	3733	3736	3756
4095	3834	3837	3839	3843	3845	3849	3852	3855	3884	3886	3914	3960	3962	4014
4360	4097	4099	4111	4116	4126	4192	4200	4238	4240	4248	4343	4352	4356	4358

SOPR1 1#  
SOPR2 1#

1441	3147	3641	3739	4203	4215	4243	4251							
1825	1162	1165	1168	1171	1217	1219	1228	1266	1283	1300	1342	1359	1376	1424
2428	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
3229	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
3563	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3182	3184
	3298	3310	3320	3360	3388	3427	3430	3454	3486	3498	3505	3532	3552	3560
	3565	3596	3605	3622	3676	3679	3697	3720	3776	3779	3861	3865	3868	3910

	3917	3924	3943	3947	3967	3973	3980	4008	4011	4144	4194	4208	4210	4281	4369
\$OPSHF	1#	4208	4383												
\$OPSLB	1#	4144													
\$OPSLB	1#														
\$OPXOR	1#														
\$OR	1#														
\$PUT	1#	2404	2503	2531	2554	2514	2636	2677	2720	2761	2820	2994	3004	3076	3325
\$STRUC	1#	3446	3458	3514	3569	3760	4140								
\$SUBON	1#	1214	1222	1259	1275	1292	1311	1335	1351	1368	1387	1417	1433	1450	1469
	1501	1517	1533	1554	1567	1583	1600	1619	1645	1661	1678	1697	1723	1739	1756
	1775	1801	1817	1834	1853	1890	1924	1959	1983	2006	2033	2057	2077	2097	2121
	2145	2165	2185	2209	2233	2253	2273	2297	2321	2341	2361	2387	2418	2436	2445
	2462	2485	2517	2541	2564	2588	2624	2646	2691	2713	2732	2775	2793	2833	2850
	2883	2887	2889	2905	2932	2934	2936	2937	2948	2964	3025	3045	3063	3093	3103
	3134	3148	3151	3179	3186	3189	3190	3198	3208	3216	3218	3226	3227	3342	3348
	3350	3379	3476	3481	3483	3525	3546	3587	3592	3594	3642	3645	3653	3664	3690
	3692	3693	3717	3740	3743	3751	3767	3790	3793	3794	3875	3891	3926	3964	3969
	3983	4019	4029	4114	4118	4128	4146	4147	4159	4170	4204	4207	4212	4213	4220
	4244	4247	4252	4255	4256	4257	4259	4260	4265	4321	4346	4354	4365	4375	4377
	4392														
\$THEY	1#	1208	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	1463	1495
	1511	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750	1769
	1795	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114	2137
	2157	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439	2456
	2478	2511	2535	2558	2581	2618	2640	2707	2787	2844	2877	2895	2918	2921	2941
	2958	3014	3032	3053	3081	3096	3128	3172	3193	3203	3331	3335	3373	3464	3468
	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956	3976	4023	4108	4122
	4154	4337	4340												
\$STILA	1#														
\$STILO	1#														
\$SUNTL2	1#	2889													
\$SUNTL3	1#														
\$SUNILE	1#	2913	3168												
\$SCHRE	961#														
\$SCHTM	961#														
\$SDEFA	1#														
\$SEND5	1#														
\$SERRO	1#														
\$SESCA	1#	802#													
\$SGEN	1#	1198	1214	1259	1275	1292	1311	1335	1351	1368	1387	1417	1433	1450	1469
	1501	1517	1533	1554	1567	1583	1600	1619	1645	1661	1678	1697	1723	1739	1756
	1775	1801	1817	1834	1853	1890	1924	1959	1983	2006	2033	2057	2077	2097	2121
	2145	2165	2185	2209	2233	2253	2273	2297	2321	2341	2361	2387	2418	2436	2445
	2462	2485	2517	2541	2564	2588	2624	2646	2691	2713	2732	2775	2793	2833	2850
	2875	2883	2887	2893	2905	2913	2932	2934	2937	2948	2964	3025	3045	3063	3093
	3103	3134	3146	3148	3168	3179	3186	3190	3198	3208	3216	3218	3227	3342	3348
	3350	3379	3476	3481	3483	3525	3546	3587	3592	3594	3640	3642	3650	3661	3690
	3693	3717	3738	3740	3748	3764	3790	3794	3873	3891	3926	3964	3969	3983	4016
	4029	4070	4105	4114	4118	4128	4147	4159	4170	4172	4177	4202	4204	4213	4220
	4221	4226	4242	4244	4250	4252	4257	4260	4265	4266	4287	4321	4322	4325	4335
	4346	4354	4365	4375	4392	4393									
\$SGETS	1#	1214	1222	1259	1275	1292	1311	1335	1351	1368	1387	1417	1433	1450	1469
	1501	1517	1533	1554	1567	1583	1600	1619	1645	1661	1678	1697	1723	1739	1756
	1775	1801	1817	1834	1853	1890	1924	1959	1983	2006	2033	2057	2077	2097	2121

	2145	2165	2185	2209	2233	2253	2273	2297	2321	2341	2361	2387	2418	2436	2445
	2462	2485	2517	2541	2564	2588	2624	2646	2691	2713	2732	2775	2793	2833	2850
	2882	2883	2887	2889	2905	2932	2934	2936	2937	2948	2964	3025	3045	3063	3093
	3103	3134	3148	3151	3178	3179	3186	3189	3190	3197	3198	3207	3208	3216	3218
	3226	3227	3341	3342	3348	3350	3379	3475	3476	3481	3483	3525	3546	3586	3587
	3592	3594	3642	3645	3653	3664	3690	3692	3693	3717	3740	3743	3751	3767	3790
	3793	3794	3875	3891	3926	3964	3969	3983	4019	4029	4113	4114	4118	4128	4131
	4146	4147	4159	4170	4204	4207	4212	4213	4220	4244	4247	4252	4118	4256	4257
	4259	4260	4265	4321	4345	4346	4354	4364	4365	4375	4377	4392	4255		
\$\$GETT	1#	2882	3178	3197	3207	3341	3475	3586	4113	4131	4345	4364			
\$\$LPCN	1#	3147	3641	3739	4203	4243	4251								
\$\$NEWT	1#	802#	1177	1239	1317	1394	1476	1540	1626	1704	1782	1865	1899	1933	1967
	2017	2105	2193	2281	2369	2469	2573	2653	2736	2797	2853	2975	3118	3278	3403
\$\$POP	1#	3613	3703	3804	3998	4041									
	1501	1214	1222	1259	1275	1292	1311	1335	1351	1368	1387	1417	1433	1450	1469
	1775	1517	1533	1554	1567	1583	1600	1619	1645	1661	1678	1697	1723	1739	1756
	2145	1801	1817	1834	1853	1890	1924	1959	1983	2006	2033	2057	2077	2097	2121
	2462	2165	2185	2209	2233	2253	2273	2297	2321	2341	2361	2387	2418	2436	2445
	2883	2485	2517	2541	2564	2588	2624	2646	2691	2713	2732	2775	2793	2833	2850
	3134	2887	2889	2905	2932	2934	2936	2937	2948	2964	3025	3045	3063	3093	3103
	3350	3148	3151	3179	3186	3189	3190	3198	3208	3216	3218	3226	3227	3342	3348
	3692	3379	3476	3481	3483	3525	3546	3587	3592	3594	3642	3645	3653	3664	3690
	3983	3693	3717	3740	3743	3751	3767	3790	3793	3794	3875	3891	3926	3964	3969
	4244	4019	4029	4114	4118	4128	4146	4147	4159	4170	4204	4207	4212	4213	4220
	4392	4247	4252	4255	4256	4257	4259	4260	4265	4321	4346	4354	4365	4375	4377
\$\$PUSH	1#	1198	1199	1208	1210	1253	1255	1269	1271	1286	1288	1305	1307	1329	1331
	1345	1347	1362	1364	1381	1383	1411	1413	1427	1429	1444	1446	1463	1465	1495
	1497	1511	1513	1527	1529	1548	1550	1561	1563	1577	1579	1594	1596	1613	1615
	1639	1641	1655	1657	1672	1674	1691	1693	1717	1719	1733	1735	1750	1752	1769
	1771	1795	1797	1811	1813	1828	1830	1847	1849	1879	1881	1913	1915	1946	1948
	1976	1978	1991	1993	2026	2028	2049	2051	2069	2071	2089	2091	2114	2116	2137
	2139	2157	2159	2177	2179	2202	2204	2225	2227	2245	2247	2265	2267	2290	2292
	2313	2315	2333	2335	2353	2355	2380	2382	2412	2414	2430	2432	2439	2441	2456
	2452	2478	2480	2511	2513	2535	2537	2558	2560	2581	2583	2618	2620	2640	2642
	2686	2687	2707	2709	2727	2728	2770	2771	2787	2789	2828	2829	2844	2846	2875
	2876	2877	2879	2884	2895	2897	2913	2914	2916	2918	2920	2921	2923	2936	2941
	2943	2958	2960	3014	3016	3032	3034	3053	3055	3081	3083	3096	3098	3128	3130
	3144	3146	3147	3151	3168	3169	3171	3172	3174	3180	3189	3193	3195	3199	3201
	3205	3209	3331	3333	3335	3337	3343	3373	3375	3464	3466	3468	3470	3477	3519
	3521	3539	3541	3575	3577	3579	3581	3588	3638	3640	3641	3645	3650	3651	3661
	3662	3683	3685	3710	3712	3736	3738	3739	3743	3748	3749	3764	3765	3783	3785
	3873	3874	3880	3882	3920	3922	3952	3954	3956	3958	3976	3978	4016	4017	4023
	4025	4071	4105	4106	4108	4110	4115	4122	4124	4146	4154	4156	4178	4200	4202
	4203	4207	4227	4240	4242	4243	4247	4248	4250	4251	4255	4288	4326	4335	4336
	4337	4339	4340	4342	4347	4366									
\$\$SELE	1#														
\$\$SET	5039#	5048	5049	5050	5051	505C	5055	5056	5057						
\$\$SETH	1118#														
\$\$SETS	1#	1198	1199	1208	1210	1253	1255	1269	1271	1286	1288	1305	1307	1329	1331
	1345	1347	1362	1364	1381	1383	1411	1413	1427	1429	1444	1446	1463	1465	1495
	1497	1511	1513	1527	1529	1548	1550	1561	1563	1577	1579	1594	1596	1613	1615
	1639	1641	1655	1657	1672	1674	1691	1693	1717	1719	1733	1735	1750	1752	1769
	1771	1795	1797	1811	1813	1828	1830	1847	1849	1879	1881	1913	1915	1946	1948
	1976	1978	1991	1993	2026	2028	2049	2051	2069	2071	2089	2091	2114	2116	2137
	2139	2157	2159	2177	2179	2202	2204	2225	2227	2245	2247	2265	2267	2290	2292



N12

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 160  
DVDVAA.P11 06-MAY-77 15:29 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0156

. ABS. 015144 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DVDVAA/CRF/NL: TOC/SOL=DVDVAA.SML, SPMAC.SML, DVDVAA.P11

RUN-TIME: 91 97 6 SECONDS  
RUN-TIME RATIO: 4641/195=23.7  
CORE USED: 43K (85 PAGES)

B13