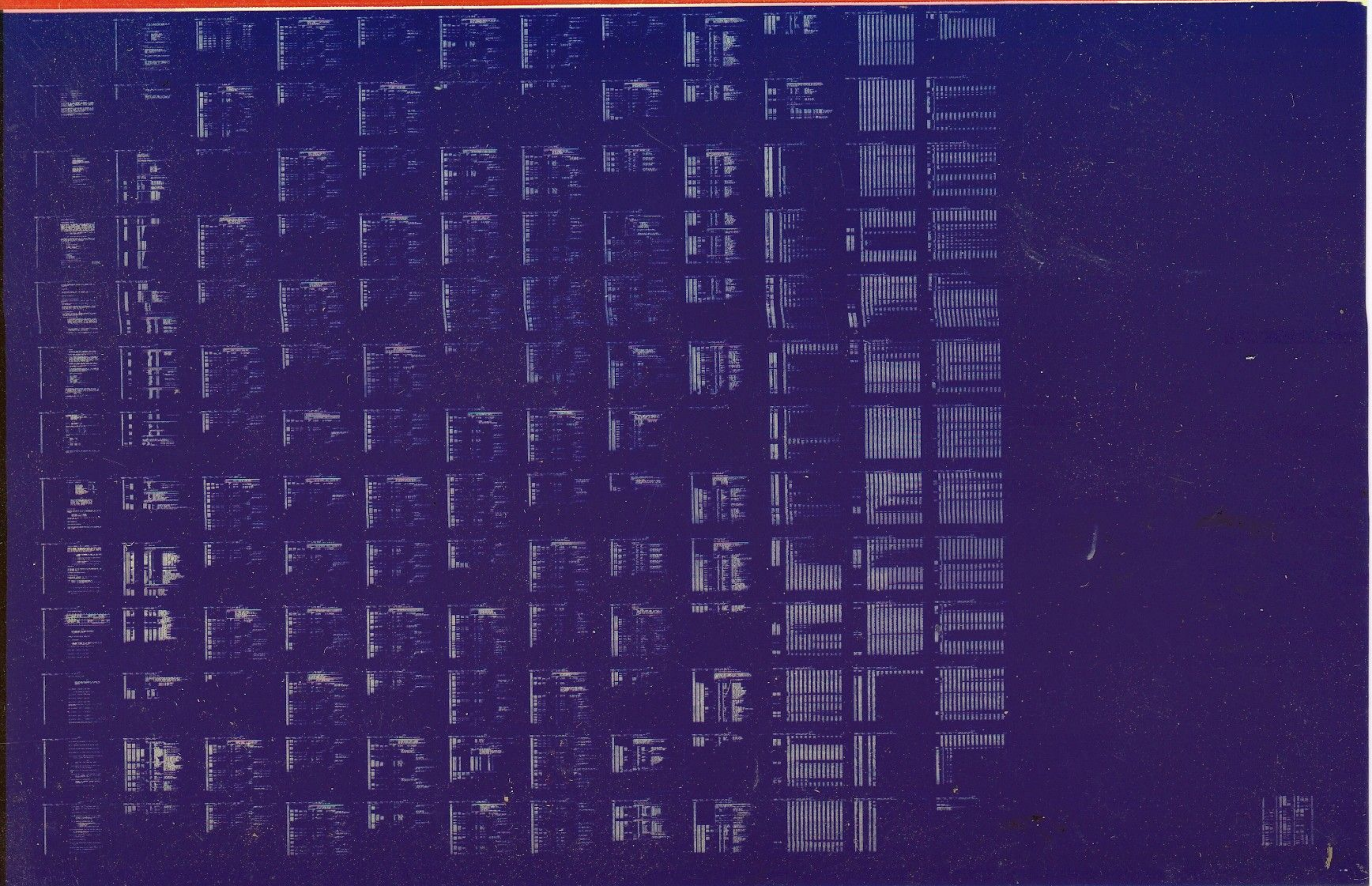


# DLV11-E

OFFLINE TEST  
MD-11-DVDVA-A

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

JUN 1977  
**digital**  
MADE IN USA





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

801

SEQ 0001

1  
2

.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.
6.0	SUMMARY OF TESTS AND SPECIAL SUBROUTINES



75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128

1.0 GENERAL PROGRAM INFORMATION.  
 -----

1.1 PROGRAM PURPOSE (ABSTRACT).

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

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

300 - VECTOR FOR FIRST OF 16 DEVICES.

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

1.2 SYSTEM REQUIREMENTS.

1. HARDWARE REQUIREMENTS:

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

SOFTWARE REQUIREMENTS:

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

STAND ALONE  
 WITH APT MONITOR  
 WITH ACT MONITOR  
 WITH XXDP MONITOR (CHAINABLE)

1.3 RELATED DOCUMENTS AND STANDARDS.

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS  
 APT  
 ACT  
 SYSMAC

175-003-009-02  
 MD-11-DZZMA  
 AUTOCAT-11-QZAUB  
 MD-11-DZQAC

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 'SDEVM' 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.



H01

SEQ 0007

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 '\$DEVN' 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 '\$DEVN' PRIOR TO STARTING THE PROGRAM.

EXAMPLE:

\$BASE: 175610  
\$VECTOR: 300  
\$DEVN: 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 PROCEDURE.

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 (RS) 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.

C02

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 17  
DVDVAA.P11 06-MAY-77 15:29

SEQ 0015

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

001100

000011  
000012  
000015  
000200  
177776  
177774  
177772  
177570  
177570

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
000006  
000007

000000

```

@
.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
    
```

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

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

.\*DATA BIT DEFINITIONS (BIT00 TO BIT15)

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



```

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

```

```

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

```

```

815 ;*****
816 ; RCSR REGISTER BIT NAMES
817 ;*****
818 DATAINT= BIT15 ; DATASET INTERRUPT
819 RING= BIT14 ; RINGING SIGNAL INDICATOR
820 CLRSEND= BIT13 ; CLEAR TO SEND FROM DATASET
821 CARDET= BIT12 ; CARRIER DETECT
822 RCVRACT= BIT11 ; RECEIVER ACTIVE INDICATOR
823 SECREC= BIT10 ; SECONDARY RECEIVE
824 ; UNUSED BIT09
825 ; UNUSED BIT08
826 RCVRDONE= BIT07 ; RECEIVER DONE
827 RCVRIE= BIT06 ; RECEIVER INTERRUPT ENABLE
828 DATAIE= BIT05 ; DATASET INTERRUPT ENABLE
829 ; UNUSED BIT04
830 SECXMIT= BIT03 ; SECONDARY TRANSMIT DATA
831 REQSEND= BIT02 ; REQUEST TO SEND
832 DTR= BIT01 ; DATA TERMINAL READY
833 ; UNUSED BIT00

```

836

837  
838  
839  
840  
841 100000  
842 040000  
843 020000  
844 010000  
845  
846  
847  
848  
849 000200  
850 000100  
851 000040  
852 000020  
853 000010  
854 000004  
855 000002  
856 000001  
857  
858  
859  
860  
861 100000  
862 040000  
863 020000  
864 010000  
865 004000  
866  
867  
868  
869  
870 000200  
871 000100  
872  
873  
874  
875 000004  
876  
877 000001  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891 000200  
892 000100

```
;;*****  
; RBUF REGISTER BIT NAMES  
;;*****  
ERROR= BIT15 ; ERROR INDICATOR  
ORERR= BIT14 ; OVERRUN ERROR  
FRERR= BIT13 ; FRAMING ERROR  
PERR= BIT12 ; PARITY ERROR  
; UNUSED BIT11  
; UNUSED BIT10  
; UNUSED BIT09  
; UNUSED BIT08  
RDATA7= BIT07 ; \  
RDATA6= BIT06 ; |  
RDATA5= BIT05 ; | RECEIVED DATA BITS  
RDATA4= BIT04 ; |  
RDATA3= BIT03 ; |  
RDATA2= BIT02 ; |  
RDATA1= BIT01 ; |  
RDATA0= BIT00 ; /
```

```
;;*****  
; TCSR REGISTER BIT NAMES  
;;*****  
PBAUD3= BIT15 ; \  
PBAUD2= BIT14 ; | PROGRAMMABLE BAUD  
PBAUD1= BIT13 ; | RATE BITS  
PBAUD0= BIT12 ; |  
PBAUDSET= BIT11 ; ENABLE SETTING OF  
; PROGRAMMABLE BAUDE RATE  
; UNUSED BIT10  
; UNUSED BIT09  
; UNUSED BIT08  
XMITRDY= BIT07 ; TRANSMITTER READY  
XMITIE= BIT06 ; TRANSMITTER INTERRUPT ENABLE  
; UNUSED BIT05  
; UNUSED BIT04  
; UNUSED BIT03  
MAINT= BIT02 ; MAINTENANCE SET BIT  
; UNUSED BIT01  
BREAK= BIT00 ; SEND BREAK (CONTINUOUS SPACE)
```

```
;;*****  
; TBUF REGISTER BIT NAMES  
;;*****  
; UNUSED BIT15  
; UNUSED BIT14  
; UNUSED BIT13  
; UNUSED BIT12  
; UNUSED BIT11  
; UNUSED BIT10  
; UNUSED BIT09  
; UNUSED BIT08  
TDATA7= BIT07 ; \  
TDATA6= BIT06 ; |
```



893 000040  
894 000020  
895 00001C  
896 000004  
897 000002  
898 000001  
899  
900  
901  
902  
903  
904 000017  
905 000020  
906 000040  
907 000100  
908 000200  
909  
910  
911  
912 007400  
913 010000  
914 020000  
915 040000  
916  
917  
918  
919 000000  
920  
921  
922  
923 000174  
924 000174 000000  
925 000176 000000  
926  
927 000200 000137 001336

TDATA5= BIT05  
TDATA4= BIT04  
TDATA3= BIT03  
TDATA2= BIT02  
TDATA1= BIT01  
TDATA0= BIT00  
: !  
: \ TRANSMITTER DATA BUFFER  
: /  
: !  
: /

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

DATA = 17  
PARITY = 20  
EVENODD = 40  
COMSPD = 100  
PBR = 200

; BAUDE MUST BE ON THE UPPER  
; BYTE BOUNDRY OF SUSWR.--4 BITS  
BAUD = 7400  
BRK = 10000  
CABLE = 20000  
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

. =174  
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER  
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER  
.SBTTL STARTING ADDRESS(ES)  
JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

928  
929  
930  
931  
932 000204  
933 000046  
934 000046 012214  
935 000052  
936 000052 000000  
937 000204  
938 001000  
939  
940  
941  
942  
943  
944 001000  
945 000024  
946 000024 000200  
947 000044  
948 000044 001000  
949 001000  
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 $ENDAD IN .SEOP
    =52
    .WORD 0         ;;2)SET LOC.52 TO ZERO
    =$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:
$SHBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;;ADDRESS 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)
$UNITM: .WORD 30. ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD SETEND-$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  
STSTNM: .BYTE 00  
SERFLG: .BYTE 00  
\$ICNT: .WORD 00  
\$LPADR: .WORD 00  
\$LPERR: .WORD 00  
\$ERTTL: .WORD 00  
\$ITEMB: .BYTE 0  
\$ERMAX: .BYTE 1  
\$ERRPC: .WORD 0  
\$GDADR: .WORD 00  
\$BDADR: .WORD 00  
\$GDAT: .WORD 00  
\$BDAT: .WORD 00  
; 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: ; APT MAILBOX  
\$MSGTY: .WORD AMSGTY ; MESSAGE TYPE CODE  
\$FATAL: .WORD AFATAL ; FATAL ERROR NUMBER  
\$TESTN: .WORD ATESTN ; TEST NUMBER  
\$PASS: .WORD APASS ; PASS COUNT  
\$DEVCT: .WORD ADEVCT ; DEVICE COUNT  
\$SUNIT: .WORD AUNIT ; I/O UNIT NUMBER  
\$MSGAD: .WORD AMSGAD ; MESSAGE ADDRESS

K02

1017	001212	000000	\$MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
1018	001214		SETABLE:		:: APT ENVIRONMENT TABLE
1019	001214	000	SENV: .BYTE	AENV	:: ENVIRONMENT BYTE
1020	001215	000	SENV: .BYTE	AENV	:: ENVIRONMENT MODE BITS
1021	001216	000000	SSWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
1022	001220	071110	SUSWR: .WORD	AUSWR	:: USER SWITCHES
1023	001222	000000	SCPUOP: .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	SMTYP1: .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	\$MADR1: .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	SMTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
1040	001232	000000	\$MADR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
1041	001234	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
1042	001235	000	SMTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
1043	001236	000000	\$MADR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
1044	001240	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
1045	001241	000	SMTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
1046	001242	000000	\$MADR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
1047	001244	000300	SVECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
1048	001246	000000	SVECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
1049	001250	175610	SBASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
1050	001252	000001	SDEV: .WORD	ADEV	:: DEVICE MAP
1051	001254		SETEND:		
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 SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.  
;\*NOTE1: IF SITEMB 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

```

1079 001336
1080
1081
1082 001336 012706 001100
1083 001342 005026
1084 001344 022706 001140
1085 001350 001374
1086 001352 012706 001100
1087
1088 001356 012737 014132 000020
1089 001364 012737 000340 000022
1090 001372 012737 013732 000030
1091 001400 012737 000340 000032
1092 001406 012737 015064 000034
1093 001414 012737 000340 000036
1094 001422 012737 012250 000024
1095 001430 012737 000340 000026
1096 001436 016767 010520 010510
1097 001444 005067 177510
1098 001450 005067 177506
1099 001454 112767 000001 177433
1100 001462 012767 001462 177416
1101 001470 012767 001470 177412
1102
1103
1104 001476 013746 000004
1105 001502 012737 001536 000004
1106 001510 012767 177570 177422
1107 001516 012767 177570 177416
1108 001524 022777 177777 177406
1109 001532 001012
1110
1111 001534 000403
1112 001536 012716 001544 64$:
1113 001542 000002 RTI
1114 001544 012767 000176 177366 65$:
1115 001552 012767 000174 177362 MOV
1116 001560 012637 000004 66$:
1117 MOV
1118 001564 005067 177412 CLR $PASS
1119 001570 132767 000200 177417 BITB $APTSIZE,$ENVM
1120 001576 001403 BEQ 67$
1121 001600 012767 001216 177332 MOV $SSWREG,$SWR
1122 001606
1123
1124
1125 001606 005227 177777
1126 001612 001037
1127 001614 022737 012214 000042
1128 001622 001433
1129 001624 104401 001672
1130
1131 001630 005737 000042
1132 001634 001012
1133 001636 126727 177352 000001
1134 001644 001406

START:
.SBTTL INITIALIZE THE COMMON TAGS
;;CLEAR THE COMMON TAGS ($CMTAG) AREA
MOV $CMTAG,$R6 ;;FIRST LOCATION TO BE CLEARED
CLR ($R6)+ ;;CLEAR MEMORY LOCATION
CMP $SWR,$R6 ;;DONE?
BNE -6 ;;LOOP BACK IF NO
MOV $STACK,$SP ;;SETUP THE STACK POINTER
;;INITIALIZE A FEW VECTORS
MOV $SCOPE,$IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
MOV $340,$IOTVEC+2 ;;LEVEL 7
MOV $ERROR,$EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
MOV $340,$EMTVEC+2 ;;LEVEL 7
MOV $TRAP,$TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
MOV $340,$TRAPVEC+2 ;;LEVEL 7
MOV $SPWRN,$PWRVEC ;;POWER FAILURE VECTOR
MOV $340,$PWRVEC+2 ;;LEVEL 7
MOV $ENDCT,$SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
MOVB #1,$SERMAX ;;ALLOW ONE ERROR PER TEST
MOV $,$SLPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV $,$SLPERR ;;SETUP THE ERROR LOOP ADDRESS
;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV $ERRVEC,-($SP) ;;SAVE ERROR VECTOR
MOV $64,$ERRVEC ;;SET UP ERROR VECTOR
MOV $DSWR,$SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
MOV $DDISP,$DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
CMP #-1,$SWR ;;TRY TO REFERENCE HARDWARE SWR
BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
;;AND THE HARDWARE SWR IS NOT = -1
BR 65$ ;;BRANCH IF NO TIMEOUT
MOV $65,$($SP) ;;SET UP FOR TRAP RETURN
MOV $SWREG,$SWR ;;POINT TO SOFTWARE SWR
MOV $DISPREG,$DISPLAY
MOV ($SP)+,$ERRVEC ;;RESTORE ERROR VECTOR
CLR $PASS ;;CLEAR PASS COUNT
BITB $APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
BEQ 67$ ;;YES,USE NON-APT SWITCH
MOV $SSWREG,$SWR ;;NO,USE APT SWITCH REGISTER
67$:
.SBTTL TYPE PROGRAM NAME
;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
INC #-1 ;;FIRST TIME?
BNE 68$ ;;BRANCH IF NO
CMP $SENDAD,$#42 ;;ACT-11?
BEQ 68$ ;;BRANCH IF YES
TYPE 69$ ;;TYPE ASCIZ STRING
.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
TST $#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
BNE 70$ ;;BRANCH IF YES
CMPB $ENV,#1 ;;ARE WE RUNNING UNDER APT?
BEQ 70$ ;;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

SEQ 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$: .ASCIZ <CRLF>*MD-11-DVDVA-A*<CRLF>
1143 001712                      68$:
```

```

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

```

```

1177      ;*****
1178      ;*TEST 1      ADDRESSABILITY
1179      ;*          THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
1180      ;*          THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
1181      ;*          TO THAT ADDRESS SPACE
1182      ;*****
1183      002034 000004      TST1:  SCOPE
1184      002036 012767 000002 177114      MOV      #2, $TIMES      ;;DO 2 ITERATIONS
1185      002044 012767 000001 177126      MOV      #1, $TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1186      002052      LET      ADRS := DLADD
1187      002052 016701 177176      MOV      DLADD, ADRS
1188      ; SET UP INTERRUPT
1189      002056      SETVEC  #ILLMEM, #INTSRV, #PR7
1190      002056 010146      MOV      R1, -(SP)
1191      002060 012701 000004      MOV      #ILLMEM, R1
1192      002064 012721 011574      MOV      #INTSRV, (R1)+
1193      002070 012711 000340      MOV      #PR7, (R1)
1194      002074 012601      MOV      (SP)+, R1
1195      002076      LET      I := #0
1196      002076 005067 177170      CLR      I
1197      002102      REPEAT
1198      002102      $1:
1199      002102      BGNSUB
1200      002102 012767 002110 177000      MOV      #64$, $LPERR
1201      ; CLEAR FLAG
1202      002110      LET INTFLAG := #0
1203      002110 005067 007466      CLR      INTFLAG
1204      ; READ FLAG
1205      ; IF INTFLAG NE #0 THEN
1206      002114 005711      TST  @ADRS
1207      002116      IF INTFLAG NE #0 THEN
1208      002116 005767 007460      TST  INTFLAG
1209      002122 001401      BEQ  $2
1210      ; FATAL ERROR
1211      002124      ERRDF 1,,NODL
1212      002124 104001      ERROR  1
1213      ; ENDIF
1214      002126      $2:
1215      002126      ENDSUB
1216      002126      LET  I := I + #2
1217      002126 062767 000002 177136      ADD      #2, I
1218      002134      LET  ADRS := DLADD + I
1219      002134 016701 177114      MOV      DLADD, ADRS
1220      002140 066701 177126      ADD      I, ADRS
1221      ; UNTIL I EQ #8.
1222      002144 026727 177122 000010      CMP      I, #8.
1223      002152 001353      BNE  $1
1224      ; CLRVEC ILLMEM
1225      002154 010146      MOV      R1, -(SP)      ;;PUSH R1 ON STACK
1226      002156 010246      MOV      R2, -(SP)      ;;PUSH R2 ON STACK
1227      002160 012701 000004      MOV      #ILLMEM, R1
1228      002164 010102      MOV      R1, R2
1229      002166 062702 000002      ADD      #2, R2
1230      002172 010221      MOV      R2, (R1)+
1231      002174 005011      CLR      (R1)
1232      002176 012602      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  
ENDTST ;END OF TEST

```

1236
1237
1238
1239
1240
1241
1242
1243
1244 002202 000004
1245 002204 012767 000010 176746
1246 002212 012767 000002 176760
1247
1248
1249 002220
1250 002220 012767 002226 176662
1251
1252 002226
1253 002226 032777 000001 177030
1254 002234 001401
1255
1256 002236
1257 002236 104002
1258 002240
1259 002240
1260 002240
1261
1262
1263 002240
1264 002240 012767 002246 176642
1265 002246
1266 002246 052777 000001 177010
1267
1268 002254
1269 002254 032777 000001 177002
1270 002262 001001
1271
1272 002264
1273 002264 104003
1274 002266
1275 002266
1276 002266
1277
1278
1279 002266
1280 002266 012767 002274 176614
1281
1282 002274
1283 002274 042777 000001 176762
1284
1285 002302
1286 002302 032777 000001 176754
1287 002310 001401
1288
1289 002312
1290 002312 104004
1291 002314

```

```

*****
* THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
*****
*TEST 2      BREAK - TCSR0 SET, CLEAR, RESET
*           THIS BIT IS THE ONLY ONE IN THIS POSITION
*           THAT IS READ AND WRITE.
*****
TST2:  SCOPE
      MOV  #10,$TIMES      ;;DO 10 ITERATIONS
      MOV  #2,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                               ; SEE IF IT IS CLEAR
                               BGNSUB
      MOV  #64,$SLPERR
                               IF  #BREAK SETIN @TCSR THEN
      BIT  #BREAK,@TCSR
      BEQ  $3
                               ; BREAK DID NOT RESET IN TCSR
                               ERRHRD 2,,DIDNOT
      ERROR 2
                               ENDIF
$3:
                               ENDSUB
                               ; TRY TO SET BREAK BIT
                               BGNSUB
      MOV  #64,$SLPERR
      BIS  #BREAK,@TCSR
      LET  @TCSR := @TCSR SET.BY #BREAK
                               ; STUCK TO 0
      IF  #BREAK NOTSETIN @TCSR THEN
      BIT  #BREAK,@TCSR
      BNE  $4
                               ; BREAK DID NOT SET IN TCSR
                               ERRHRD 3,,DIDNOT
      ERROR 3
                               ENDIF
$4:
                               ENDSUB
                               ; TRY TO CLEAR A SET BIT
                               BGNSUB
      MOV  #64,$SLPERR
      LET  @TCSR := @TCSR CLR.BY #BREAK
      BIC  #BREAK,@TCSR
      IF  #BREAK SETIN @TCSR THEN
      BIT  #BREAK,@TCSR
      BEQ  $5
                               ; BREAK DID NOT CLEAR IN TCSR
                               ERRHRD 4,,DIDNOT
      ERROR 4
                               ENDIF

```

```

1292 002314 $5:
1293 002314 ENDSUB
1294
1295 ; NOW SEE IF RESET CLEARS IT
1296 002314 ;
1297 002314 012767 002322 176566 MOV #64$, $LPERR ;
1298 ;
1299 002322 LET @TCSR := @TCSR SET.BY #BREAK ;
1300 002322 052777 000001 176734 BIS #BREAK, @TCSR ; ISSUE BUS RESET
1301 ;
1302 002330 BRES@T ;
1303 002330 000005 RESET ;
1304 002332 IF #BREAK SETIN @TCSR THEN ;
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 ;
1311 002344 $6: ;
1312 002344 ENDSUB ;
1313 002344 ENDTST ;
1314
1315

```



```

1316
1317
1318
1319
1320 002344 000004
1321 002346 012767 000010 176604
1322 002354 012767 000003 176616
1323
1324
1325 002362
1326 002362 012767 002370 176520
1327
1328 002370
1329 002370 032777 000004 176666
1330 002376 001401
1331
1332 002400
1333 002400 104006
1334 002402
1335 002402
1336 002402
1337
1338
1339 002402
1340 002402 012767 002410 176500
1341 002410
1342 002410 052777 000004 176646
1343
1344 002416
1345 002416 032777 000004 176640
1346 002424 001001
1347
1348 002426
1349 002426 104007
1350 002430
1351 002430
1352 002430
1353
1354
1355 002430
1356 002430 012767 002436 176452
1357
1358 002436
1359 002436 042777 000004 176620
1360
1361 002444
1362 002444 032777 000004 176612
1363 002452 001401
1364
1365 002454
1366 002454 104010
1367 002456
1368 002456
1369 002456
1370
1371

```

```

*****
*****
*TEST 3      MAINT - TCSR2 SET, CLEAR, RESET
*****
↑ST3:  SCOPE
      MOV      #10,$TIMES      ;;DO 10 ITERATIONS
      MOV      #3,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
      ; SEE IF IT IS CLEAR
      BGNSUB
      MOV      #64,$SLPERR
      IF      #MAINT SETIN @TCSR THEN
      BIT      #MAINT,@TCSR
      BEQ     $7
      ; MAINT DID NOT RESET IN TCSR
      ERRHRD 6,,DIDNOT
      ENDIF
$7:
      ENDSUB
      ; TRY TO SET MAINT BIT
      BGNSUB
      MOV      #64,$SLPERR
      LET     @TCSR := @TCSR SET.BY #MAINT
      BIS     #MAINT,@TCSR
      IF ; STUCK TO 0
      ; #MAINT NOTSETIN @TCSR THEN
      BIT     #MAINT,@TCSR
      BNE     $10
      ; MAINT DID NOT SET IN TCSR
      ERRHRD 7,,DIDNOT
      ENDIF
$10:
      ENDSUB
      ; TRY TO CLEAR A SET BIT
      BGNSUB
      MOV      #64,$SLPERR
      LET     @TCSR := @TCSR CLR.BY #MAINT
      BIC     #MAINT,@TCSR
      IF ; SHOULD HAVE CLEARED
      ; #MAINT SETIN @TCSR THEN
      BIT     #MAINT,@TCSR
      BEQ     $11
      ; MAINT DID NOT CLEAR INTCSR
      ERRHRD 8,,DIDNOT
      ENDIF
$11:
      ENDSUB
      ; NOW SEE IF RESET CLEARS IT

```

H03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 35  
DVDVAA.P11 06-MAY-77 15:29 T3 MAINT - TCSR2 SET, CLEAR, RESET

SEQ 0033

```
1372 002456                                BGNSUB
1373 002456 012767 002464 176424          MOV    #64$, $LPERR
1374                                     LET    @TCSR := @TCSR SET.SY #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                                     ENDIF
1385 002504 104011          ERROR  9
1386 002506                                     ENDSUB
1387 002506          $12:                                     ENDTST
1388 002506
1389 002506
1390
1391
1392
```

```

1393
1394
1395
1396
1397 002506 000004
1398 002510 012767 000010 176442
1399 002516 012767 000004 176454
1400
1401 002524 012746 000340
1402 002530 012746 002536
1403 002534 000002
1404 002536
1405
1406
1407 002536
1408 002536 012767 002544 176344
1409
1410 002544
1411 002544 032777 000100 176512
1412 002552 001401
1413
1414 002554
1415 002554 104012
1416 002556
1417 002556
1418 002556
1419
1420
1421 002556
1422 002556 012767 002564 176324
1423 002564
1424 002564 052777 000100 176472
1425
1426 002572
1427 002572 032777 000100 176464
1428 002600 001001
1429
1430 002602
1431 002602 104013
1432 002604
1433 002604
1434 002604
1435
1436
1437 002604
1438 002604 012767 002612 176276
1439
1440 002612
1441 002612 042777 000100 176444
1442
1443 002620
1444 002620 032777 000100 176436
1445 002626 001401
1446
1447 002630
1448 002630 104014

```

```

*****
*****
*TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
*****
$T4: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #4,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
;; USE PRIORITY OF 7
MOV #PR7,-(SP) ;;PUT NEW PS ON STACK
MOV #64$,-(SP) ;;PUT NEW PC ON STACK
RTI ;;POP NEW PC AND PS
64$:
; SEE IF IT IS CLEAR
BGNSUB
MOV #65$,$LPERR
IF #XMITIE SETIN @TCSR THEN
BIT #XMITIE,@TCSR
BEQ $I3
; XMITIE DID NOT RESET IN TCSR
ERRHRD 10,,DIDNOT
ERROR 10
ENDIF
$I3:
ENDSUB
; TRY TO SET XMITIE BIT
BGNSUB
MOV #64$,$LPERR
LET @TCSR := @TCSR SET.BY #XMITIE
BIS #XMITIE,@TCSR
IF ; STUCK TO 0
; #XMITIE NOTSETIN @TCSR THEN
BIT #XMITIE,@TCSR
BNE $I4
; XMIT DID NOT RESET IN TCSR
ERRHRD 11,,DIDNOT
ERROR 11
ENDIF
$I4:
ENDSUB
; TRY TO CLEAR A SET BIT
BGNSUB
MOV #64$,$LPERR
LET @TCSR := @TCSR CLR.BY #XMITIE
BIC #XMITIE,@TCSR
IF ; SHOULD HAVE CLEARED
; #XMITIE SETIN @TCSR THEN
BIT #XMITIE,@TCSR
BEQ $I5
; XMIT DID NOT CLEAR IN TCSR
ERRHRD 12,,DIDNOT
ERROR 12

```



J03

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 37  
DVDVAA.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
1456
1457 002640
1458 002640 052777 000100 176416
1459
1460 002646
1461 002646 000005
1462 002650
1463 002650 032777 000100 176406
1464 002656 001401
1465
1466 002660
1467 002660 104015
1468 J02662
1469 002662
1470 002662
1471 002662
1472
1473
1474

```

\$15:

\$16:

ENDIF

ENDSUB

; NOW SEE IF RESET CLEARS IT

BGNSUB

MOV #64\$, \$LPERR

LET @TCSR := @TCSR SET.BY #XMITIE

BIS #XMITIE, @TCSR

: ISSUE BUS RESET  
BRESÉ

RESET

IF #XMITIE SETIN @TCSR THEN

BIT #XMITIE, @TCSR

BEQ \$16

: XMIT DID NOT RESET IN TCSR  
ERRHRD 13,,DIDNOT

ERROR 13

ENDIF

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.
*****
TST5:    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
                                        IF     ; STUCK TO 0
                                               #DTR SETIN @RCSR THEN
                                        ; DTR DID NOT CLEAR IN RCSR
                                        ERRHRD 14,,DIDNOT
                                        ENDIF
$17:                                        ENDSUB
                                        ; TRY TO SET DTR
                                        BGNSUB
         MOV        #64$,$LPERR
                                        LET     @RCSR := @RCSR SET.BY #DTR
         BIS        #DTR,@RCSR
                                        IF     #DTR NOTSETIN @RCSR THEN
         BIT        #DTR,@RCSR
         BNE        $20
                                        ; 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
                                        IF     ; SHOULD HAVE CLEARED IT
                                               #DTR SETIN @RCSR THEN
         BIT        #DTR,@RCSR
         BEQ        $21
                                        ; 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  
DVDVAAR.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 176204
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 JUMPER 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
                                   ENDIF
$22:
                                   ; SEE IF IT IS CLEAR
                                   BGNSUB
      MOV      #64,$$LPERR
                                   IF      #REQSEND SETIN @RCSR THEN
      BIT      #REQSEND,@RCSR
      BEQ      $23
                                   ; REQSEND DID NOT RESET IN RCSR
                                   ERRHRD 17,,DIDNOT
      ERROR   17
                                   ENDIF
$23:
                                   ENDSUB
                                   ; TRY TO SET REQSEND BIT
                                   BGNSUB
      MOV      #64,$$LPERR
      LET      @RCSR := @RCSR SET.BY #REQSEND
      BIS      #REQSEND,@RCSR
                                   ; STUCK TO 0
      IF      #REQSEND NOTSETIN @RCSR THEN
      BIT      #REQSEND,@RCSR
      BNE      $24
                                   ; REQSEND DID NOT SET IN RCSR
                                   ERRHRD 18,,DIDNOT
      ERROR   18
                                   ENDIF
$24:
                                   ENDSUB
                                   ; TRY TO CLEAR A SET BIT
                                   BGNSUB
      MOV      #64,$$LPERR
      LET      @RCSR := @RCSR CLR.BY #REQSEND
      BIC      #REQSEND,@RCSR
      IF      #REQSEND SETIN @RCSR THEN

```



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  
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  
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,$SLPERR  
      IF      #SECXMIT SETIN @RCSR THEN  
      BIT      #SECXMIT,@RCSR  
      BEQ     $27  
      ; SECXMIT DID NOT RESET IN RCSR  
      ERRHRD 21,,DIDNOT  
      ERROR   21  
      ENDIF  
$27:  ENDSUB  
      ; TRY TO SET SECXMIT BIT  
      BGNSUB  
      MOV      #64,$SLPERR  
      LET     @RCSR := @RCSR SET.BY #SECXMIT  
      BIS     #SECXMIT,@RCSR  
      IF     ; STUCK TO 0  
      #SECXMIT NOTSETIN @RCSR THEN  
      BIT     #SECXMIT,@RCSR  
      BNE     $30  
      ; SECXMIT DID NOT SET IN RCSR  
      ERRHRD 22,,DIDNOT  
      ERROR   22  
      ENDIF  
$30:  ENDSUB  
      ; TRY TO CLEAR A SET BIT  
      BGNSUB  
      MOV      #64,$SLPERR  
      LET     @RCSR := @RCSR CLR.BY #SECXMIT  
      BIC     #SECXMIT,@RCSR  
      IF     ; SHOULD HAVE CLEARED  
      #SECXMIT SETIN @RCSR THEN  
      BIT     #SECXMIT,@RCSR  
      BEQ     $31  
      ; 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
1700
1701
1702
                                ENDIF
                                ENDSUB
                                ENDTST

```

```

1703
1704
1705
1706
1707
1708
1709 003326 000004
1710 003330 012767 000010 175622
1711 003336 012767 000010 175634
1712
1713 003344
1714 003344 012767 003352 175536
1715
1716 003352
1717 003352 032777 000040 175700
1718 003360 001401
1719
1720 003362
1721 003362 104031
1722 003364
1723 003364
1724 003364
1725
1726
1727 003364
1728 003364 012767 003372 175516
1729 003372
1730 003372 052777 000040 175660
1731
1732 003400
1733 003400 032777 000040 175652
1734 003406 001001
1735
1736 003410
1737 003410 104032
1738 003412
1739 003412
1740 003412
1741
1742
1743 003412
1744 003412 012767 003420 175470
1745
1746 003420
1747 003420 042777 000040 175632
1748
1749 003426
1750 003426 032777 000040 175624
1751 003434 001401
1752
1753 003436
1754 003436 104033
1755 003440
1756 003440
1757 003440
1758

```

```

*****
*****
*TEST 10      DATAIE - RCSR3 SET, CLEAR, RESET
*              THIS BIT IS THE ONLY ONE IN THIS POSITION
*              THAT IS READ AND WRITE.
*****
↑ST10: SCOPE
      MOV      #10,$TIMES      ;;DO 10 ITERATIONS
      MOV      #10,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                               ; SEE IF IT IS CLEAR
                               BGNSUB
      MOV      #64,$SLPERR
      IF      #DATAIE SETIN @RCSR THEN
      BIT      #DATAIE,@RCSR
      BEQ     $33
                               ; DATAIE DID NOT RESET IN RCSR
                               ERRHRD 25,,DIDNOT
      ERROR   25
      ENDIF
$33:
      ENDSUB
                               ; TRY TO SET DATAIE BIT
                               BGNSUB
      MOV      #64,$SLPERR
      BIS      #DATAIE,@RCSR
      LET     @RCSR := @RCSR SET.BY #DATAIE
      IF      ; STUCK TO 0
      #DATAIE NOTSETIN @RCSR THEN
      ERROR   26
      ENDIF
$34:
      ENDSUB
                               ; TRY TO CLEAR A SET BIT
                               BGNSUB
      MOV      #64,$SLPERR
      BIC      #DATAIE,@RCSR
      LET     @RCSR := @RCSR CLR.BY #DATAIE
      IF      ; SHOULD HAVE CLEARED
      #DATAIE SETIN @RCSR THEN
      BIT      #DATAIE,@RCSR
      BEQ     $35
                               ; DATAIE DID NOT CLEAR IN RCSR
                               ERRHRD 27,,DIDNOT
      ERROR   27
      ENDIF
$35:
      ENDSUB

```

E04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 45  
DVDVAA.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    #64$, $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                                     $36:
1776 003470
1777 003470                               ENDSUB
1778
1779
1780                               ENDTST

```



```

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 MOV #64$,SLPERR BGNSUB
1792 MOV #64$,SLPERR
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 $37:
1802 ENDSUB
1803
1804 ; TRY TO SET RCVRIE BIT
1805 BGNSUB
1806 MOV #64$,SLPERR
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 $40:
1818 ENDSUB
1819
1820 ; TRY TO CLEAR A SET BIT
1821 BGNSUB
1822 MOV #64$,SLPERR
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  
DVDVAA.P11

06-MAY-77

MACY11 27(1006)  
15:29

16-MAY-77 10:36  
T11

PAGE 47  
RCVRIE - RCSR6 SET, CLEAR, RESET

SEQ 0045

; NOW SEE IF RESET CLEARS IT  
BGNSUB

1837  
1838 003602  
1839 003602 012767 003610 175300  
1840  
1841 003610  
1842 003610 052777 000100 175442  
1843  
1844 003616  
1845 003616 000005  
1846 003620  
1847 003620 032777 000100 175432  
1848 003626 001401  
1849  
1850 003630  
1851 003630 104040  
1852 003632  
1853 003632  
1854 003632  
1855 003632  
1856 003632  
1857  
1858  
1859  
1860

MOV #64\$, \$LPERR  
BIS #RCVRIE, @RCSR  
RESET  
BIT #RCVRIE, @RCSR  
BEQ \$42

ERROR 32

\$42:

LET @RCSR := @RCSR SET.BY #RCVRIE

: ISSUE BUS RESET  
BRESÉT

IF #RCVRIE SETIN @RCSR THEN

: RCVRIE DID NOT RESET IN RCSR  
ERRHRD 32,,DIDNOT

ENDIF

CKLOOP

ENDSUB

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

SEQ 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  
DVDVAA.P11

MACY11 27(1006)  
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

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

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 - 15 CLEARED BY INIT  
*****  
TST15: SCOPE  
MOV #10,$TIMES ;;DO 10 ITERATIONS  
MOV #15,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX  
  
IF #CABLE NOTSETIN $USWR THEN  
BIT #CABLE,$USWR  
BNE $46 ; CAN'T TEST WITHOUT BERG OR H315.  
EXIT TST  
MOV #1,$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  
:  
:TESTING EFFECT OF RESET ON BIT  
:RCVRACT DID NOT CLEAR IN RCSR  
:  
:ALLOW ANOTHER TRY  
BRESET  
ENDIF  
:  
:ALLOW LOOPING ON ERROR  
CKLOOP  
ENDSUB  
ENDTST
```

\$47:



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 RCVRCT - RCSR 11 - 15 CLEARED BY INIT

SEQ 0050

2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067

004052 000004  
004054 012767 000010 175076  
004062 012767 000016 175110  
004070  
004070 032767 020000 175122  
004076 001004  
004100  
004100 012767 000001 175052  
004106 000441  
004110  
004110  
004110  
004110 012767 004116 174772  
004116  
004116 042777 000002 175134  
004124  
004124 032777 010000 175126  
004132 001401  
004134  
004134 104045  
004136  
004136  
004136  
004136  
004136 012767 004144 174744  
004144  
004144 052777 000002 175106

```
*****
* 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,$$LPERR
BIC #DTR,$RCSR
LET ; CLEAR DTR
; RCSR := RCSR CLR.BY #DTR
IF ; CARDET SHOULD FOLLOW
#CARDET SETIN $RCSR THEN
; CARDET DID NOT
ERRHRD 37,,FORCE
; CLEAR WITH DTR
ENDIF
$S1:
ENDSUB
; SET
BGNSUB
MOV #64,$$LPERR
BIS #DTR,$RCSR
LET ; SET DTR
; RCSR := RCSR SET.BY #DTR
; CARDET SHOULD FOLLOW
```

M04

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 53  
 DVDVAA.P11 06-MAY-77 15:29 T16 TEST THAT CARDET SETS AND CLEARS

SEQ 0051

```

2068 004152
2069 004152 032777 010000 175100 BIT #CARDET, @RCSR
2070 004160 001001 BNE $52
2071 ; CARDET DID NOT SET
2072 004162 ; ERRHRD 38,,FORCE
2073 004162 104046 ERROR 38
2074
2075 ; WITH DTR
2076 004164 ENDIF
2077 004164 $52: ENDSUB
2078 004164
2079
2080 ; CLEAR
2081 004164 BGNSUB
2082 004164 012767 004172 174716 MOV #64$, $LPERR
2083
2084 ; CLEAR DTR
2085 004172 LET @RCSR := @RCSR CLR.BY #DTR
2086 004172 042777 000002 175060 BIC #DTR, @RCSR
2087 ; CARDET SHOULD FOLLOW
2088 004200 IF #CARDET SETIN @RCSR THEN
2089 004200 032777 010000 175052 BIT #CARDET, @RCSR
2090 004206 001401 BEQ $53
2091 ; CARDET DID NOT
2092 004210 ; ERRHRD 39,,FORCE
2093 004210 104047 ERROR 39
2094
2095 ; CLEAR WITH DTR
2096 004212 ENDIF
2097 004212 $53: ENDSUB
2098 004212
2099 004212 ENDTST
2100
2101
2102
2103
  
```

```

2104 .....
2105 .....
2106 *TEST 17 TEST THAT CLREND SETS AND CLEARS
2107 * AS DTR SETS AND CLEARS
2108 .....
2109 †ST17: SCOPE
2110 004212 000004 MOV #10,$TIMES ;;DO 10 ITERATIONS
2111 004214 012767 000010 174736 MOV #17,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2112 004222 012767 000017 174750 IF ; CAN WE USE THE WRAPAROUND??
2113 004230 ; CANBLE NOTSETIN $USWR THEN
2114 004230 032767 020000 174762 BIT #CABLE,$USWR
2115 004236 001004 BNE $54
2116 ; CAN'T TEST WITHOUT BERG OR H315.
2117 004240 EXIT TST
2118 004240 012767 000001 174712 MOV #1,$TIMES
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 LET ; CLEAR DTR
2133 004256 042777 000002 174774 BIC #DTR,@RCSR @RCSR := @RCSR CLR.BY #DTR
2134 004256 042777 000002 174774 IF ; CLREND SHOULD FOLLOW
2135 004264 ; #CLREND SETIN @RCSR THEN
2136 004264 032777 020000 174766 BIT #CLREND,@RCSR
2137 004264 001401 BEQ $55
2138 004272
2139 ; CLREND DID NOT
2140 004274 ERRHRD 40,,FORCE
2141 004274 104050 ERROR 40
2142
2143 ENDIF ; CLEAR WITH DTR
2144 004276
2145 004276 $55:
2146 004276 ENDSUB
2147
2148 ; SET
2149 004276 BGNSUB
2150 004276 012767 004304 174604 MOV #64,$SLPERR
2151
2152 LET ; SET DTR
2153 004304 @RCSR := @RCSR SET.BY #DTR
2154 004304 052777 000002 174746 BIS #DTR,@RCSR
2155 004312 ; CLREND SHOULD FOLLOW
2156 004312 032777 020000 174740 BIT #CLREND,@RCSR
2157 004312 001001 BNE $56 #CLREND NOTSETIN @RCSR THEN
2158 004320 ; CLREND DID NOT SET
2159

```





```

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,$SUSWR ;CABLE NOTSETIN $SUSWR THEN
2202 004370 032767 020000 174622 BIT #CABLE,$SUSWR
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

```





E05

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 58  
DVDVAA.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 ENDIF
2297 004550 $64:
2298
2299 ; SECXMIT AND
2300 ; SECREC ARE CONNECTED
2301
2302 ; BY THE H315 OR EQUIV.
2303
2304 ; CLEAR
2305 004550 BGNSUB
2306 004550 012767 004556 174332 MOV #64,$LPERR
2307
2308 ; CLEAR SECXMIT
2309 004556 LET @RCSR := @RCSR CLR.BY #SECXMIT
2310 004556 042777 000010 174474 BIC #SECXMIT,@RCSR
2311 ; SECREC SHOULD FOLLOW
2312 004564 IF #SECREC SETIN @RCSR THEN
2313 004564 032777 002000 174466 BIT #SECREC,@RCSR
2314 004572 001401 BEQ $65
2315 ; SECREC DID NOT
2316 004574 ERRHRD 46,,FORCE
2317 004574 104056 ERROR 46
2318
2319 ; CLEAR WITH SECXMIT
2320 004576 ENDIF
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 ; SECREC SHOULD FOLLOW
2332 004612 IF #SECREC NOTSETIN @RCSR THEN
2333 004612 032777 002000 174440 BIT #SECREC,@RCSR
2334 004620 001001 BNE $66
2335 ; SECREC DID NOT SET

```



2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423

004652 000004  
004654 012767 000010 174276  
004662 012767 000022 174310  
004670  
004670 032767 020000 174322  
004676 001004  
004700  
004700 012767 000001 174252  
004706 000463  
004710  
004710  
004710 012746 000340  
004714 012746 004722  
004720 000002  
004722  
004722  
004722 012767 004730 174160  
004730  
004730 042777 000002 174322  
004736  
004736 010546  
004740 012745 000010  
004744 004767 004544  
004750 012605  
004752  
004752 017703 174302  
004756  
004756 032777 100000 174274  
004764 001401  
004766 104061  
004770  
004770  
004770  
004770

```
*****  
*****  
*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 CLSEND BY THE H315  
*****  
TST22: SCOPE  
MOV #10,$TIMES ;;DO 10 ITERATIONS  
MOV #22,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX  
IF #CABLE NOTSETIN $USWR THEN ; CAN WE USE THE WRAPAROUND??  
BIT #CABLE,$USWR  
BNE $70 ; CABLE NOTSETIN $USWR THEN  
EXIT 'TST ; CAN'T TEST WITHOUT BERG OR H315.  
MOV #1,$TIMES  
BR TST23 ;;EXIT THIS TEST  
ENDIF  
$70:  
MOV #PR7,-(SP) ;MAKE SURE NOTHING UNEXPECTED HAPPENS  
MOV #64$,-(SP) ;;PUT NEW PS ON STACK  
RTI ;;PUT NEW PC ON STACK  
;;POP NEW PC AND PS  
64$:  
;READ TWICE - CLEARS  
BGNSUB  
MOV #65$,$LPERR  
LET ; CLEAR DTR  
RCSR := RCSR CLR.BY #DTR  
BIC #DTR,RCSR  
;WAIT 10 MICRO-SEC FOR CABLE  
WAITUS 10  
MOV R5,-(SP)  
MOV #10,-(R5)  
JSR PC,WAIT  
MOV (SP)+,R5  
LET ; READ RCSR - TO CLEAR DATAINT  
R3 := RCSR  
; READ RCSR AGAIN  
IF #DATAINT SETIN RCSR THEN  
BIT #DATAINT,RCSR  
BEQ $71  
; READING RCSR DID NOT CLEAR DATAINT  
ERRHRD 49,EDATAINT  
ERROR 49  
ENDIF  
$71:  
ENDSUB  
; DTR SETTING SETS DATAINT  
BGNSUB
```



# H05

MAINDEC-11-DVDVA-A    MACY11 27(1006)    16-MAY-77 10:36    PAGE 61  
 DVDVAA.P11    06-MAY-77 15:29    T22    TEST THAT DATAINT (RCSR-15) SETS

SEQ 0059

2424	004770	012767	004776	174112	MOV	#64\$, \$LPERR	
2425							
2426							
2427	004776						;SET DTR
2428	004776	052777	000002	174254	BIS	#DTR, @RCSR	LET @RCSR := @RCSR SET.BY #DTR
2429	005004						IF #DATAINT NOTSETIN @RCSR THEN
2430	005004	032777	100000	174246	BIT	#DATAINT, @RCSR	
2431	005012	001001			BNE	\$72	
2432							:SETTING DTR DID NOT SET DATAINT
2433	005014						ERRHRD 50,, E2DATA
2434	005014	104062			ERROR	50	
2435	005016						ENDIF
2436	005016					\$72:	
2437							
2438	005016						IF #DATAINT SETIN @RCSR THEN
2439	005016	032777	100000	174234	BIT	#DATAINT, @RCSR	
2440	005024	001401			BEQ	\$73	
2441							:READING RCSR DID NOT CLEAR DATAINT
2442	005026						ERRHRD 51, E2DATA
2443	005026	104063			ERROR	51	
2444	005030						ENDIF
2445	005030					\$73:	
2446	005030						ENDSUB
2447							
2448							; DTR CLEARING SETS DATAINT
2449	005030						BGNSUB
2450	005030	012767	005036	174052	MOV	#64\$, \$LPERR	
2451							
2452							
2453	005036						;CLEAR DTR
2454	005036	042777	000002	174214	BIC	#DTR, @RCSR	LET @RCSR := @RCSR CLR.BY #DTR
2455	005044						IF #DATAINT NOTSETIN @RCSR THEN
2456	005044	032777	100000	174206	BIT	#DATAINT, @RCSR	
2457	005052	001001			BNE	\$74	
2458							:CLEARING DTR DID NOT SET DATAINT
2459	005054						ERRHRD 52,, E2DATA
2460	005054	104064			ERROR	52	
2461	005056						ENDIF
2462	005056					\$74:	
2463	005056						ENDSUB
2464	005056						ENDTST
2465							
2466							
2467							

```

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,$TIMES ;;DO 10 ITERATIONS
2475 005056 012767 000010 174072 MOV #23,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2476 005056 012767 000023 174104 IF ;CAN WE USE THE WRAPAROUND??
2477 005074 ;CABLE NOTSETIN SUSWR THEN
2478 005074 032767 020000 174116 BIT #CABLE,$USWR
2479 005102 001004 BNE $75
2480 ;CAN'T TEST WITHOUT BERG OR H315.
2481 ;EXIT TST
2482 005104 012767 000001 174046 MOV #1,$TIMES
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 BGNSUB
2495 005126 012767 005134 173754 MOV #65$,$LPERR
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
2503 005144 010546 MOV R5,-(SP)
2504 005150 012745 000010 MOV #10,-(R5)
2505 005154 004767 004340 JSR PC,WAIT
2506 005154 012605 MOV (SP)+,R5
2507 005156
2508 005156 017703 174076 MOV @RCSR,R3
2509 ;READ ONCE
2510 005162 LET R3 := @RCSR
2511 005162 032777 100000 174070 BIT #DATAINT,@RCSR
2512 005170 001401 BEQ $76 ;READ TWICE
2513 ;DATAINT SETIN @RCSR THEN
2514 005172 ;READING RCSR DID NOT CLEAR DATAINT
2515 005172 104065 ERROR 53 ERRHRD 53, EDATAINT
2516 005174 ENDF
2517 005174 $76:
2518 005174 ENDSUB
2519
2520 ; SET RING --> SET DATAINT
2521 005174 BGNSUB
2522 005174 012767 005202 173706 MOV #64$,$LPERR
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 005302 000004
2577 005304 012767 000010 173646
2578 005312 012767 000024 173660
2579
2580 005320
2581 005320 032767 020000 173672
2582 005326 001004
2583
2584 005330
2585 005330 012767 000001 173622
2586 005336 000454
2587 005340
2588 005340
2589
2590
2591 005340 012746 000340
2592 005344 012746 005352
2593 005350 000002
2594 005352
2595
2596
2597
2598
2599 005352
2600 005352 042777 000010 173700
2601 005360
2602 005360 017703 173674
2603
2604
2605 005364
2606 005364 012767 005372 173516
2607
2608
2609 005372
2610 005372 052777 000010 173660
2611
2612 005400
2613 005400 010546
2614 005402 012745 000010
2615 005406 004767 004102
2616 005412 012605
2617 005414
2618 005414 032777 100000 173636
2619 005422 001001
2620
2621 005424
2622 005424 104124
2623 005426
2624 005426
2625 005426
2626
2627

```

```

*****
*****
*TEST 24 TEST THAT DATAINT SETS WHEN SECURE CHANGES STATE
*****
TST24: 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
;CAN'T TEST WITHOUT BERG OR H315.
EXIT TST
;;EXIT THIS TEST
ENDIF

$101:
MOV #PR7,-(SP) ;;PUT NEW PS ON STACK
MOV #64$,-(SP) ;;PUT NEW PC ON STACK
RTI ;;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
;SETTING SECURE DID NOT SET DATAINT
ERRHRD 84,, E2DATA
ERROR 84
ENDIF
ENDSUB
;CLEAR SECURE --> DATAINT SET

```



# M05

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

T24    TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE

SEQ 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    CLR    @TBUF
2672    ;    WAIT A MAXIMUM
2673    ;    OF 50 MSEC FOR
2674    ;    XMIT RDY TO SET IN TCSR
2675    CALL    TIMER IN <#5,#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    ;    TIMER RETURNS AN ERROR IF BIT DID
2684    ;    NOT MEET CONDITION WITHIN TIME LIMIT
2685    IF.ERROR THEN
2686    BCC    $104
2687    ;    XMIT RDY DID NOT SET IN TCSR
2688    ERRHRD 54,,DIDNOT
2689    ERROR    54
2690    ENDIF
2691    $104:
2692    ENDSUB
2693    BGNSUB
2694    MOV    #64,$SLPERR
2695    ;    LOAD TBUF WITH A SECOND CHARACTER
2696    ;    CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
2697    ;    AND THEN WAIT FOR IT TO SET
2698
2699    ;    SEND SECOND CHARACTER
2700    LET    @TBUF :B= #0
2701    CLR    @TBUF
2702    ;    GIVE IT TIME TO CLEAR
2703    ;    XMITRDY SHOULD HAVE CLEARED UPON
2704    ;    RECEIPT OF A CHARACTER
2705    IF    #XMITRDY SET IN @TCSR THEN
2706    BIT    #XMITRDY,@TCSR
2707
  
```



# N05

MAINDEC-11-DVDVA-A    MACY11 27(1006)    16-MAY-77 10:36 PAGE 67  
 DVDVA.A.P11    06-MAY-77 15:29    T25    TEST THAT XMIT RDY - TCSR 7 - CLEARS    SEQ 0065

2708	005576	001401		BEQ	\$105		
2709						: XMITRDY DID NOT CLEAR IN TCSR	
2710	005600					ERRHRD 55,,DIDNOT	
2711	005600	104067		ERROR	55		
2712	005602					ENDIF	
2713	005602		\$105:				
2714							
2715						: WAIT A MAXIMUM	
2716						: OF 50 MSEC FOR	
2717						: XMIT RDY TO SET IN TCSR	
2718	005602					CALL TIMER IN (<#5,#XMITRDY,TCSR,#SET>	
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					IF.ERROR THEN	
2727	005632	103001		BCC	\$106		
2728						: XMIT RDY DID NOT SET IN TCSR	
2729	005634					ERRHRD 56,,DIDNOT	
2730	005634	104070		ERROR	56		
2731	005636					ENDIF	
2732	005636		\$106:				
2733	005636					ENDSUB	
2734	005636					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

        ; 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
        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  
DVDVAA.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 R0 :B= @RBUF

IF #RCVRDONE SETIN @RCSR THEN

;RCVRDONE DID NOT CLEAR IN RCSR
ERRHRD 60,, DIDNOT
ENDIF

$111:
$112:
ENDTST

```

E06

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 71  
DVDVA.A.P11 06-MAY-77 15:29

T27 TEST THAT RCVRDONE IS CLEARED BY READING RBUF

SEQ 0069

```

2852 ;*****
2853 ;*****
2854 ;*TEST 30 TEST THAT RCVRACT - RCSR 11 - SETS
2855 ;* WHEN A START BIT IS RECEIVED AND
2856 ;* CLEARS WHEN RCVRDONE - RCSR 7 - SETS
2857 ;*****
2858 006062 000004 TST30: SCOPE
2859 006064 012767 000010 173066 MOV #10,$TIMES ;;DO 10 ITERATIONS
2860 006072 012767 000030 173100 MOV #30,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2861
2862 006100 LET @TCSR := @TCSR SET.BY #MAINT
2863 006100 052777 000004 173156 BIS #MAINT,@TCSR
2864 006106 LET FLAG :B= #CLR
2865 006106 112767 000000 003272 MOVB #CLR,FLAG
2866 006114 LET COUNT := #0
2867 006114 005067 000172 CLR COUNT
2868 ;LOAD A CHARACTER INTO TBUF
2869 ;WAIT FOR RCVRACT TO SET
2870
2871 ;SEND A CHARACTER
2872 006120 LET @TBUF :B= #0
2873 006120 105077 173144 CLRB @TBUF
2874 006124 REPEAT
2875 006124 $113: IF #RCVRACT SET IN @RCSR THEN
2876 006124 IF #RCVRACT SET IN @RCSR THEN
2877 006124 032777 004000 173126 BIT #RCVRACT,@RCSR
2878 006132 001404 BEQ $114
2879 006134 LET FLAG :B= #SET
2880 006134 112767 177777 003244 MOVB #SET,FLAG
2881 006142 ELSE
2882 006142 000402 BR $115
2883 006144 $114: LET COUNT := COUNT + #1
2884 006144
2885 006144 005267 000142 INC COUNT
2886 006150 ENDIF
2887 006150 $115: UNTIL FLAG EQ #SET OR COUNT HI MAX
2888 006150
2889 006150 026727 003232 177777 CMP FLAG,#SET
2890 006156 001404 BEQ $116
2891 006160 026767 000126 000122 CMP COUNT,MAX
2892 006166 101756 BLOS $113
2893 006170 $116: IF COUNT HI MAX THEN
2894 006170
2895 006170 026767 000116 000112 CMP COUNT,MAX
2896 006176 101405 BLOS $117
2897 ;IT NEVER SET
2898 ;RCVRACT DID NOT SET IN RCSR.
2899 006200 ERRHRD 61,, DIDNOT
2900 006200 104075 ERROR 61
2901 006202 EXIT TEST
2902 006202 012767 000001 172750 MOV #1,$TIMES
2903 006210 000441 BR TST31 ;;EXIT THIS TEST
2904 006212
2905 006212 $117: ENDF
2906
2907

```

```

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    $123
2923
2924                                     ;RCVRDONE AND RCVRACT
2925 006242                                     ;BOTH SET
2926 006242 104076          ERROR    62          ERRHRD 62, DONEACT
2927
2928                                     ;NO USE CONTINUING
2929 006244                                     EXIT TST
2930 006244 012767 000001 172706          MOV    #1,$TIMES
2931 006252 000420          BR      TST31          ;;EXIT THIS TEST
2932                                     ENDIF
2933 006254                                     $123:
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
2944                                     ;RCVRDONE DID NOT SET IN RCSR
2945 006266 104077          ERROR    63          ERRHRD 63,,DIDNOT
2946                                     ;SET IT BACK.
2947 006270                                     ENDIF
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
2961                                     ;RCVRDONE DID NOT CLEAR IN RCSR
2962 006304 104100          ERROR    64          ERRHRD 64,,DIDNOT
2963 006306                                     ENDIF
    
```



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 11 - 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 RCVRACT - 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
      WAIT    50
                              ;OUTPUT 2ND CHARACTER
      LET    @TBUF :B= #0
                              ;LET OVERRUN HAPPEN
      WAIT    50
                              ;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$, $LPERR
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, ,DIDNOT
3038
3039
3040
3041 006442                                ; -WHEN ORERR SET.
3042 006442 012767 000001 172510        MOV    #1, $TIMES
3043 006450 000465                        BR     TST32                ;;EXIT THIS TEST
3044 006452                                ENDIF
3045 006452                                $127:
3046 006452                                ENDSUB
3047
3048 006452                                BGNSUB
3049 006452 012767 006460 172430        MOV    #64$, $LPERR
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
    
```

;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.  
 IF #ORERR NOTSETIN RBUF THEN  
 BIT #ORERR, RBUF  
 BNE \$130  
 :READING RBUF CLEARED ORERR.  
 ERRHRD 67, ITCLRED  
 ;SKIP REST OF TEST  
 EXIT  
 ;;EXIT THIS TEST  
 ENDIF  
 ENDSUB  
 BGNSUB  
 MOV #64\$, \$LPERR  
 ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED  
 ;SEND A CHARACTER AROUND.  
 LET RBUF :B= #0  
 ;LET IT CIRCULATE  
 WAITUS 50  
 MOV R5, -(SP)  
 MOV #50, -(R5)  
 JSR PC, WAIT  
 MOV (SP)+, R5  
 IF #ORERR SETIN RBUF THEN  
 BIT #ORERR, RBUF  
 BEQ \$131  
 :ORERR DID NOT CLEAR IN RBUF  
 ERRHRD 68, ,DIDNOT





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

006624 000004  
006626 012767 000010 172324  
006634 012767 000032 172336  
006642  
006642 032767 000200 172350  
006650 001004  
006652  
006652 012767 000001 172300  
006660 000542  
006662  
006662  
006662 012767 177777 000272  
006670  
006670 012767 177777 000266  
006676 052777 000004 172360  
006704  
006704 005003  
006706 000401  
006710  
006710 005203  
006712  
006712 020327 000017  
006716 003060  
006720  
006720 017700 172336  
006724  
006724 116377 007104 172334  
006732  
006732 005002  
006734  
006734 005077 172330  
006740  
006740 005067 000212  
006744  
006744 005067 000210  
006750  
006750  
006750 005702  
006752 001014  
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:  
MOV #-1,OLD  
LET OLD+2 := #-1  
MOV #-1,OLD+2  
LET @TCSR := @TCSR SET.BY #MAINT  
BIS #MAINT,@TCSR  
;EACH BAUD RATE  
INCR R3 FROM #0 TO #15. BY #1  
CLR R3  
BR $134  
$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 032777 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:
    
```



```

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 CORROSPONDING
; ** BAUD TO BE SELECTED IN THE DLV11-E UPON
; ** COMPLETION OF THIS TEST.

```

					BAUD	OFFSET INTO TABLE
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 007156
3271 007156 000000 000000 .EVEN
3272 007162 000000 000000 NEW: 0,0
3273 007166 OLD: 0,0

```

ENDTST

3274  
3275  
3276

```

3277      ;*****
3278      ;*****
3279      ;TEST 33      TRANSMITTER INTERRUPT LOGIC TEST
3280      ;          LOGICALLY THIS IS 4 SEPARATE TESTS
3281      ;          A) DOES TRANSMITTER INTERRUPT LOGIC WORK
3282      ;          B) AT PRIORITY OF 0
3283      ;          C) AND ONLY ONCE
3284      ;          D) BUT NOT WITH INTERRUPT ENABLE CLEAR
3285      ;*****
3286      007166 000004      ST33: SCOPE
3287      007170 012767 000010 171762      MOV      #10,$TIMES      ;;DO 10 ITERATIONS
3288      007176 012767 000033 171774      MOV      #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3289      ;          ;CLEAR 'INTERRUPT OCCURED' FLAG
3290      007204      CLR      INTFLAG      LET INTFLAG := #0
3291      007204 005067 002372
3292
3293      ;          ;GET VECTOR ADDRESS
3294      007210      MOV      DLVEC,R3      LET R3 := DLVEC
3295      007210 016703 172042
3296      ;          ;FOR THE TRANSMITTER
3297      007214      ADD      #4,R3      LET R3 := R3 + #4
3298      007214 062703 000004
3299      ;          ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
3300      007220      SETVEC R3, #INTSRV, #PR7
3301      007220 010146      MOV      R1,-(SP)
3302      007222 010301      MOV      R3,R1
3303      007224 012721 011574      MOV      #INTSRV,(R1)+
3304      007230 012711 000340      MOV      #PR7,(R1)
3305      007234 012601      MOV      (SP)+,R1
3306      007236      BGNSUB
3307      007236 012767 007244 171644      MOV      #64,$SLPERR
3308      ;          ;CLEAR INTERRUPT ENABLE
3309      007244      LET #ATCSR := #ATCSR CLR.BY #XMITIE
3310      007244 042777 000100 172012      BIC      #XMITIE,#ATCSR
3311
3312      ;          ;SET IT TO 0
3313      007252 012746 000000      MOV      #PRO,-(SP)      ;;PUT NEW PS ON STACK
3314      007256 012746 007264      MOV      #65$,-(SP)      ;;PUT NEW PC ON STACK
3315      007262 000002      RTI      ;;POP NEW PC AND PS
3316      007264      65$:
3317
3318      ;          ;NOW SET I.E. BIT
3319      007264      LET #ATCSR := #ATCSR SET.BY #XMITIE
3320      007264 052777 000100 171772      BIS      #XMITIE,#ATCSR
3321
3322      ;          ;LET INTERRUPT HAVE TIME TO OCCUR
3323      007272      WAITUS 10
3324      007272 010546      MOV      R5,-(SP)
3325      007274 012745 000010      MOV      #10,-(R5)
3326      007300 004767 002210      JSR      PC,WAIT
3327      007304 012605      MOV      (SP)+,R5
3328
3329      ;          ;DID EXACTLY 1 INTERRUPT OCCUR
3330      007306      IF INTFLAG NE #1 THEN
3331      007306 026727 002270 000001      CMP      INTFLAG,#1
3332      007314 001406      BEQ      $147
    
```

```

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          $147:
3351 007332                                     ENDSUB
3352                                     ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
3353 007332                                     BGNSUB
3354 007332 012767 007340 171550      MOV     #64$, $LPERR
3355                                     ;CLEAR 'INTERRUPT OCCURED' FLAG
3356 007340                                     LET INTFLAG := #0
3357 007340 005067 002236          CLR     INTFLAG
3358                                     ;CLEAR INTERRUPT ENABLE
3359 007344                                     LET @TCSR := @TCSR CLR.BY #XMITIE
3360 007344 042777 000100 171712      BIC     #XMITIE, @TCSR
3361                                     ;NO INTERRUPTS SHOULD OCCUR.
3362 007352 012746 000000          MOV     #PRO, -(SP)      ;;PUT NEW PS ON STACK
3363 007356 012746 007364          MOV     #65$, -(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
    
```





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 005067 002072  
  
007510  
007510 052777 000004 171546  
  
007516  
007516 042777 000100 171534  
  
007524 012746 000000  
007530 012746 007536  
007534 000002  
007536  
  
007536  
007536 105077 171526  
  
007542  
007542 010546  
007544 012745 177777  
007550 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  
CLR INTFLAG  
;;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,-(R5)
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                                ENDIF
3481 007640              $155:
3482 007640                                ENDIF
3483 007640              $153:
3484                                ;RESET MAINT. BIT.
3485 007640                                LET @TCSR := @TCSR CLR.BY #MAINT
3486 007640 042777 000004 171416      BIC    #MAINT,@TCSR
3487 007646                                ENDSUB
3488
3489
3490
3491
3492
3493
3494                                ; INTERRUPT WITHOUT I E SET.
3495 007646                                BGNSUB
3496 007646 012767 007654 171234      MOV    #64$,$LPERR
3497 007654                                LET @TCSR := @TCSR SET.BY #MAINT
3498 007654 052777 000004 171402      BIS    #MAINT,@TCSR
3499
3500                                ;CLEAR INTERRUPT FLAG
3501 007662                                LET INTFLAG := #0
3502 007662 005067 001714      CLR    INTFLAG
3503                                ;CLEAR INTERRUPT
3504 007666                                LET @RCSR := @RCSR CLR.BY #RCVRIE
3505 007666 042777 000100 171364      BIC    #RCVRIE,@RCSR
3506                                ;DON'T LET THEM IN
3507 007674 012746 000340      MOV    #PR7,-(SP)      ;;PUT NEW PS ON STACK
3508 007700 012746 007706      MOV    #65$,-(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    R5,-(SP)

```



```

3514 007710 012745 000010      MOV    #10,-(RS)
3515 007714 004767 001574      JSR    PC,WAIT
3516 007720 012605              MOV    (SP)+,RS
3517                               ;DID IT HAPPEN?
3518 007722              IF INTFLAG NE #0 THEN
3519 007722 005767 001654      TST    INTFLAG
3520 007726 001401              BEQ    $156
3521                               ; INTERRUPT OCCURED AT PRIORITY 7
3522 007730              ERRHRD 77,NOTENAB
3523 007730 104115              ERROR  77
3524 007732              ENDIF
3525 007732              ;CLEAR THE WORLD
3526 007732              BRESÉT
3527 007732 000005              RESET
3528 007732
3529
3530                               ;RESET MAINT. BIT.
3531 007734              LET @TCSR := @TCSR CLR.BY #MAINT
3532 007734 042777 000004 171322  BIC    #MAINT,@TCSR
3533 007742              ENDSUB
3534
3535                               ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3536 007742              BGNSUB
3537 007742 012767 007750 171140  MOV    #64$,$LPERR
3538 007750              IF #CABLE NOTSETIN $USWR THEN
3539 007750 032767 020000 171242  BIT    #CABLE,$USWR
3540 007756 001004              BNE    $157
3541                               ;CAN'T TEST WITHOUT A CABLE
3542 007760              EXIT TST
3543 007760 012767 000001 171172  MOV    #1,$TIMES
3544 007766 000463              BR     TST35
3545 007770              ;;;EXIT THIS TEST
3546 007770              ENDIF
3547                               ; CLEAR 'INTFLAG'
3548 007770              LET INTFLAG := #0
3549 007770 005067 001606      CLR    INTFLAG
3550
3551                               ;CLEAR INTERRUPTS
3552 007774 042777 000040 171256  BIC    #DATAIE,@RCSR
3553                               ;CHANGE PRIORITY
3554                               ; TO 0
3555 010002 012746 000000      MOV    #PRO,-(SP)
3556 010006 012746 010014      MOV    #64$,-(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                               ;:POP NEW PC AND PS
3562 010022              LET @RCSR := @RCSR CLR.BY #REQSEND
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                               ;LET IT COME IN.
3568 010036              WAITUS 10
3569 010040 012745 000010      MOV    RS,-(SP)
3569 010040              MOV    #10,-(RS)
    
```

\$156:

\$157:

64\$:

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

SEQ 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
3578 010062                    ; NONE OCCURED
3579 010062 005767 001514      YST    INTFLAG
3580 010066 001002              BNE    $161
3581
3582 010070                    ; DATAINT DID NOT INTERRUPT IN TIME
3583 010070 104113              ERROR  75
3584
3585 010072                    ; TWICE OR MORE
3586 010072 000401              BR     $162
3587 010074                    ELSE
3588
3589 010074                    ; DATAINT INTERRUPTED TWICE
3590 010074 104114              ERROR  76
3591 010076
3592 010076                    ERRHRD 76,,TWICE
3593 010076
3594 010076                    ENDIF
3595 010076                    ENDIF
3596 010076 042777 000040 171154  BIC    #DATAIE,ARCSR
3597 010104                    LET ARCSR := ARCSR CLR.BY #DATAIE
3598
3599 010104                    ENDSUB
3600 010104 017704 171146      MOV    @DLVEC,R4
3601 010110                    LET R4 := @DLVEC
3602 010110 010146              MOV    R1,-(SP)
3603 010112 010246              MOV    R2,-(SP)
3604 010114 012701 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
3610 010134 012601              MOV    (SP)+,R1
3611 010136

```

```

;; PUSH R1 ON STACK
;; PUSH R2 ON STACK
;; POP STACK INTO R2
;; POP STACK INTO R1
ENDTST

```

H07

```

3612
3613
3614
3615
3616
3617 010136 000004
3618 010140 012767 000001 171012
3619 010146 012767 000035 171024
3620
3621 010154
3622 010154 052777 000004 171102
3623
3624
3625
3626 010162 012746 000000
3627 010166 012746 010174
3628 010172 000002
3629 010174
3630
3631 010174
3632 010174 162705 000002
3633 010200 004767 001210
3634 010204 012501
3635
3636
3637 010206
3638 010206 005002
3639 010210 000401
3640 010212
3641 010212 005202
3642 010214
3643 010214 020227 000377
3644 010220 003023
3645
3646
3647
3648
3649 010222
3650 010222
3651
3652 010222
3653 010222 032777 000200 171034
3654 010230 001774
3655
3656
3657 010232
3658 010232 110277 171032
3659
3660 010236
3661 010236
3662
3663 010236
3664 010236 032777 000200 171014
3665 010244 001774
3666
3667 010246

```

```

*****
*****
*TEST 35      TEST ACTUAL DATA TRANSFERED
*              NON-INTERRUPT MAINTENANCE BIT SET
*****
TST35: SCOPE
        MOV      #1,$TIMES      ;;DO 1 ITERATION
        MOV      #35,$TESTN    ;;SET TEST NUMBER IN APT MAIL BOX
                                ;;SET MAINT. BIT
                                LET @TCSR := @TCSR SET.BY #MAINT
                                ;CHANGE PRIORITY
                                ;...TO 0
                                ;;PUT NEW PS ON STACK
                                ;;PUT NEW PC ON STACK
                                ;;POP NEW PC AND PS
                                ;GET DATA MASK.
                                CALL DATLNG OUT <R1>
                                ;ALL BINARY CHAR.
                                INCR R2 FROM #0 TO #377 BY #1
                                ;TRANSMIT CHAR IN R2
                                REPEAT
                                ;IS IT READY?
                                UNTIL #XMITRDY SETIN @TCSR
                                ;TRANSMIT IT
                                LET @TBUF :B= R2
                                REPEAT
                                ;WAIT TILL ITS HERE
                                UNTIL #RCVRDONE SETIN @RCSR
                                ;AND SAVE IT
                                LET R3 := @RBUF

```

```

$164:
$163:
$165:
$166:
$167:

```



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					LET R3 := R3 CLR.BY R1
3679	010256	040103		BIC	R1,R3	
3680						
3681						;RECEIVED DATA
3682	010260					IF R4 NE R3 THEN
3683	010260	020403		CMP	R4,R3	
3684	010262	001401		BEQ	\$170	
3685						;DATA COMPARE ERROR
3686	010264					ERRHRD 78,COMP,SBWAS
3687	010264	104116		ERROR	78	
3688						; <TRANSMITTED> <RECEIVED>
3689	010266					ENDIF
3690	010266		\$170:			ENDINC ; R2
3691	010266					
3692	010266	000751		BR	\$164	
3693	010270		\$165:			
3694						
3695						;RESET MAINT. BIT.
3696	010270					LET @TCSR := @TCSR CLR.BY #MAINT
3697	010270	042777	000004	170766	BIC	#MAINT,@TCSR
3698	010276					ENDTST
3699						
3700						
3701						

```

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 NOTSETIN 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 RO := @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					WAITUS 10
3759	010422	010546		MOV	R5, -(SP)	
3760	010424	012745	000010	MOV	#10, -(R5)	
3761	010430	004767	001060	JSR	PC, WAIT	
3762	010434	012605		MOV	(SP)+, R5	
3763	010436					REPEAT
3764	010436					; IS IT HERE
3765						UNTIL #RCVRDONE SETIN #RCSR
3766	010436					
3767	010436	032777	000200 170614	BIT	#RCVRDONE, #RCSR	
3768	010444	001774		BEQ	\$176	
3769						; RETRIEVE
3770						LET R3 := #RBUF
3771	010446					
3772	010446	017703	170610	MOV	#RBUF, R3	
3773						; STRIP OFF JUNK ON BOTH
3774						LET R4 := R2 CLR. BY R1
3775	010452					
3776	010452	010204		MOV	R2, R4	
3777	010454	040104		BIC	R1, R4	
3778	010456					LET R3 := R3 CLR. BY R1
3779	010456	040103		BIC	R1, R3	
3780						
3781						; WE HAVE TROUBLE
3782	010460					IF R4 NE R3 THEN
3783	010460	020403		CMP	R4, R3	
3784	010462	001401		BEQ	\$177	
3785						; DATA COMPARE ERROR
3786	010464					ERRHRD 79, COMP, SBWAS
3787	010464	104117		ERROR	79	
3788						; <R2> <R3>
3789	010466					ENDIF
3790	010466					
3791						ENDINC ; R2
3792	010466					
3793	010466	000743		BR	\$173	
3794	010470					
3795						
3796						
3797						
3798	010470					ENDTST
3799						
3800						
3801						
3802						











```

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          RTI
3992
3993 011004
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  
RHL D:0  
ENDSRV  
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,STIMES ;;DO 10 ITERATIONS
MOV #40,STESTN ;;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 SETIN @RCSR

IFB @RBUF NE #0 THEN
;;BREAK DID NOT EQUAL 0
ERRHRD 81 ,BADBRK
ENDIF
BRESET ;CLEAN UP
EXIT
;;;EXIT THIS TEST
BADBRK: .ASCIZ /BREAK DID NOT EQUAL 0/

$206:
$207:
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
    
```



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

```

```

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, -(RS)
4141 011350 004767 000140                    JSR   PC, WAIT
4142 011354 012605                    MOV   (SP)+,RS
4143 011356                                     LET   TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4144 011356 005367 000022                    DEC   TIMSAV
4145 011362                                     ENDL00P          ; CONTINUED AT THE TOP
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 $USWR WORD.
*****
```

```
*****
LET MASK(R5) := #0 ; START
CLR MASK(R5)
LET NUMBR := $USWR AND #DATA
MOV $USWR, 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
BIS #1, MASK(R5)
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 $230:
4243 011534 062702 000001
4244 011540 $227:
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
BR $227
$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 011574  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280 011574  
4281 011574 005267 000002  
4282 011600  
4283 011600 000002  
4284 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
RTI
;THAT'S ALL
INTFLAG: 0
```

```

4285
4286 011604
4287 011604
4288
4289 011604 104401 011612
4290 011610 000405
4291
4292 011624
4293 011624 016746 167350
4294 011630 104402
4295 011632 104401 011640
4296 011636 000405
4297
4298 011652
4299 011652 116767 167236 167316
4300 011660 016746 167312
4301 011664 104405
4302 011666 104401 011674
4303 011672 000404
4304
4305 011704
4306 011704 016746 167206
4307 011710 104402
4308 011712 104401 011720
4309 011716 000404
4310
4311 011730
4312 011730 016746 167320
4313 011734 104402
4314 011736 104401 011744
4315 011742 000405
4316
4317 011756
4318 011756 016746 167274
4319 011762 104402
4320 011764
4321 011764
4322 011764
4323 011764 000207

ROUTINE MYTYPE
MYTYPE:
;*****
        TYPE      65$      ;;TYPE ASCIZ STRING
        BR        64$      ;;GET OVER THE ASCIZ
;;65$: .ASCIZ  <CRLF>*TEST * *
64$:
        MOV      $TESTN,-(SP)  ;;SAVE $TESTN FOR TYPEOUT
        TYPOC
        TYPE      67$      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        66$      ;;TYPE ASCIZ STRING
;;67$: .ASCIZ  *,ERROR * *
66$:
        MOV      $ITEMB,$FATAL  ;;APT FATAL ERROR NUMBER
        MOV      $FATAL,-(SP)  ;;SAVE $FATAL FOR TYPEOUT
        TYPOD
        TYPE      69$      ;;GO TYPE--DECIMAL ASCII WITH SIGN
        BR        68$      ;;TYPE ASCIZ STRING
;;69$: .ASCIZ  *,PC = *
68$:
        MOV      $ERRPC,-(SP)  ;;SAVE $ERRPC FOR TYPEOUT
        TYPOC
        TYPE      71$      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        70$      ;;TYPE ASCIZ STRING
;;71$: .ASCIZ  *,CSR: *
70$:
        MOV      $LADD,-(SP)  ;;SAVE $LADD FOR TYPEOUT
        TYPOC
        TYPE      73$      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR        72$      ;;TYPE ASCIZ STRING
;;73$: .ASCIZ  *,VECTOR: *
72$:
        MOV      $LVEC,-(SP)  ;;SAVE $LVEC FOR TYPEOUT
        TYPOC
        ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
ENDRTN
$235:
$236:
RTS      PC

```



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:

MOV #1, BITMASK

LET BITMASK := #1

MOV #1, \$DEVCT

LET \$DEVCT := #1

MOV \$BASE, ADDRESS

LET ADDRESS := \$BASE

MOV \$VECT1, VECTOR

LET VECTOR := \$VECT1

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

```

4380 012102
4381 012102 012701 012120      MOV  #ADDRESS,ADRS      LET ADRS := #ADDRESS
4382 012106
4383 012106 005267 167072      INC  $DEVCT           LET $DEVCT := $DEVCT + #1
4384 012112
4385 012112 000404      BR   $237            RETURN
4386 012114 000000      BITMASK:             0
4387 012116 000001      INITFLAG:            1
4388 012120 000000      ADDRESS:              0
4389 012122 000000      VECTOR:               0
4390
4391 012124
4392 012124      $237:
4393 012124      $240:
4394 012124 000207      RTS   PC            ENDRTN
4395
4396

```

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 #XXXX" (WHERE XXXX IS A DECIMAL NUMBER)
;IF THERES A MONITOR GO TO IT
;IF THERE ISN'T JUMP TO SPECIALADDRESS

SEOP:
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?

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

SENDCT: .WORD 1
SEOPCT
TYPE $SENDMG ;: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 2#42,RO ;:GET MONITOR ADDRESS
BEQ $DOAGN ;:BRANCH IF NO MONITOR
RESET ;:CLEAR THE WORLD
SENDAD: JSR PC,(RO) ;: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
$SENDMG: .ASCIZ <15><12>/END PASS #/

```



```

4436          .SBTTL  POWER DOWN AND UP ROUTINES
4437
4438          ::*****
4439          :POWER DOWN ROUTINE
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          :POWER UP ROUTINE
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    SPOWER      ;; REPORT THE POWER FAILURE
4471 012404 012416      $PWRMG: .WORD    SPOWER ;; 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 001011  
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 $STPFLG ;; IS THERE A TERMINAL?
BPL 1$ ;; BR IF YES
HALT ;; HALT HERE IF NO TERMINAL
BR 3$ ;; LEAVE
1$: MOV RO, -(SP) ;; SAVE RO
MOV 32(SP), RO ;; GET ADDRESS OF ASCIZ STRING
CMPB #APTENV, SENV ;; RUNNING IN APT MODE
BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
BITB #APT$SPOOL, SENVM ;; SPOOL MESSAGE TO APT
BEQ 62$ ;; 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$: BITB #APTCSUP, SENVM ;; APT CONSOLE SUPPRESSED
BNE 60$ ;; 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
524: CLRB $SCHARCNT ;; CLEAR CHARACTER COUNT
BR 2$ ;; GET NEXT CHARACTER
55: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
65: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
BNE 2$ ;; IF NO GO GET NEXT CHAR.
MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
AND THE NULL CHAR.
75: DECB 1($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 $SCHARCNT ;; DO NOT COUNT AS A COUNT
```

```

4535 012612 000770          BR      7$          ;;LOOP
4536
4537          ;HORIZONTAL TAB PROCESSOR
4538
4539 012614 112716 000040    6$:   MOVB   #' (SP)          ;; REPLACE TAB WITH SPACE
4540 012620 004767 000014    9$:   JSR    PC,$TYPEC          ;; TYPE A SPACE
4541 012624 132767 000007 000052    BITB  #',$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    $TYPEC: TSTB  2$TPS          ;; WAIT UNTIL PRINTER IS READY
4546 012644 100375          BPL   $TYPEC
4547 012646 116677 000002 166276    MOVB  2(SP),2$TPB          ;; 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$: ADD #6,SP ;: IGNORE PREVIOUS INPUT
4598 013036 000757 BR 19$ ;: 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   @STKS      ;; WAIT FOR
4642 013206 100375          BPL      1$          ;; A CHARACTER
4643 013210 117766 165732 000004  MOVB    @STKB,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$: TSTB   @STKS      ;; WAIT FOR A CHARACTER
4648 013240 100375          BPL      2$          ;; LOOP UNTIL ITS THERE
4649 013242 117746 165700  MOVB    @STKB,-(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          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      $TTYIN,R3 ;; GET ADDRESS
4669 013320 022703 013430 2$: CMP      $TTYIN+8.,R3 ;; BUFFER FULL?

```

\*\*\*\*\*

THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY

\*CALL:

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

\*\*\*\*\*

THIS ROUTINE WILL INPUT A STRING FROM THE TTY

\*CALL:

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

4670	013324	101405				BLOS	4S	:: 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		10S:	CMPB	#177, (R3)	:: IS IT A RUBOUT
4674	013336	001003				BNE	3S	:: SKIP IF NOT
4675	013340	104401	001170		4S:	TYPE	\$QUES	:: TYPE A '?'
4676	013344	000763				BR	1S	:: CLEAR THE BUFFER AND LOOP
4677	013346	111367	000044		3S:	MOVB	(R3), 9S	:: ECHO THE CHARACTER
4678	013352	104401	013416			TYPE	9S	
4679	013356	122723	000015			CMPB	#15, (R3)+	:: CHECK FOR RETURN
4680	013362	001356				BNE	2S	:: 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			9S:	.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					



```

.SBTTL APT COMMUNICATIONS ROUTINE
*****
4697 013464 112767 000001 000236 $ATY1: MOV  #1,$FFLG ;; TO REPORT FATAL ERROR
4698 013472 112767 000001 000226 $ATY3: MOV  #1,$MFLG ;; TO TYPE A MESSAGE
4699 013500 000403 BR $ATYC
4700 013502 112767 000001 000220 $ATY4: MOV  #1,$FFLG ;; TO ONLY REPORT FATAL ERROR
4701 013510 010046 MOV  RO,-(SP) ;; PUSH RO ON STACK
4702 013512 010146 MOV  R1,-(SP) ;; PUSH R1 ON STACK
4703 013514 105767 000206 TSTB $MFLG ;; SHOULD TYPE A MESSAGE?
4704 013520 001450 BEQ  5$ ;; IF NOT: BR
4705 013522 122767 000001 165464 CMPB #APTENV,$ENV ;; OPERATING UNDER APT?
4706 013530 001031 BNE  3$ ;; IF NOT: BR
4707 013532 132767 000100 165455 BITB #APTSPool,$ENVM ;; SHOULD SPOOL MESSAGES?
4708 013540 001425 BEQ  3$ ;; IF NOT: BR
4709 013542 017600 000004 MOV  #4(SP),RO ;; GET MESSAGE ADDR.
4710 013546 062766 000002 000004 ADD  #2,4(SP) ;; BUMP RETURN ADDR.
4711 013554 005767 165414 1$: TST  $MSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
4712 013560 001375 BNE  1$ ;; IF NOT: WAIT
4713 013562 010067 165422 MOV  RO,$MSGAD ;; PUT ADDR IN MAILBOX
4714 013566 105720 2$: TSTB (RO)+ ;; FIND END OF MESSAGE
4715 013570 001376 BNE  2$
4716 013572 166700 165412 SUB  $MSGAD,RO ;; SUB START OF MESSAGE
4717 013576 006200 ASR  RO ;; GET MESSAGE LNTH IN WORDS
4718 013600 010067 165406 MOV  RO,$MSGLGT ;; PUT LENGTH IN MAILBOX
4719 013604 012767 000004 165362 MOV  #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
4720 013612 000413 BR  5$
4721 013614 017667 000004 000016 3$: MOV  #4(SP),4$ ;; PUT MSG ADDR IN JSR LINKAGE
4722 013622 062766 000002 000004 ADD  #2,4(SP) ;; BUMP RETURN ADDRESS
4723 013630 016746 164142 MOV  177776,-(SP) ;; PUSH 177776 ON STACK
4724 013634 004767 176566 JSR  PC,$TYPE ;; CALL TYPE MACRO
4725 013640 000000 4$: .WORD 0
4726 013642 5$:
4727 013642 105767 000062 10$: TSTB $FFLG ;; SHOULD REPORT FATAL ERROR?
4728 013646 001416 BEQ  12$ ;; IF NOT: BR
4729 013650 005767 165340 TST  $ENV ;; RUNNING UNDER APT?
4730 013654 001413 BEQ  12$ ;; IF NOT: BR
4731 013656 005767 165312 11$: TST  $MSGTYPE ;; FINISHED LAST MESSAGE?
4732 013662 001375 BNE  11$ ;; IF NOT: WAIT
4733 013664 017667 000004 165304 MOV  #4(SP),$FATAL ;; GET ERROR #
4734 013672 062766 000002 000004 ADD  #2,4(SP) ;; BUMP RETURN ADDR.
4735 013700 005267 165270 INC  $MSGTYPE ;; TELL APT TO TAKE ERROR
4736 013704 105067 000020 12$: CLRB $FFLG ;; CLEAR FATAL FLAG
4737 013710 105067 000013 CLRB $LFLG ;; CLEAR LOG FLAG
4738 013714 105067 000006 CLRB $MFLG ;; CLEAR MESSAGE FLAG
4739 013720 012601 MOV  (SP)+,R1 ;; POP STACK INTO R1
4740 013722 012600 MOV  (SP)+,RO ;; POP STACK INTO RO
4741 013724 000207 RTS  PC ;; RETURN
4742 013726 000 .BYTE 0 ;; MESSG. FLAG
4743 013727 000 .BYTE 0 ;; LOG FLAG
4744 013730 000 .BYTE 0 ;; FATAL FLAG
4745 013732 .EVEN
4746 000200 APTSIZE=200
4747 000001 APTENV=001
4748 000100 APTSPool=100
4749
4750
4751
4752

```

H09

MRINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 113  
DVDVAA.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 165104  
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
      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
      JSR      PC, SATY4      ;; 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
      BIT      #BIT09, @SWR      ;; LOOP ON ERROR SWITCH SET?
      BEQ      4$      ;; BR IF NO
      MOV      $LPERR, (SP)      ;; FUDGE RETURN FOR LOOPING
      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  
014202  
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

$SCOPE:
CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
BNE $OVER ;;YES IF SW14=1
;*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
;;THIS INSTRUCTION TO A "NOP" (NOP=240)
MOV 2$ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
MOV 5$,$ERRVEC ;;SET FOR TIMEOUT
TST 2$177060 ;;TIME OUT ON XOR?
MOV (SP)+,2$ERRVEC ;;RESTORE THE ERROR VECTOR
BR $SVLAD ;;GO TO THE NEXT TEST
5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
MOV (SP)+,2$ERRVEC ;;RESTORE THE ERROR VECTOR
BR 7$ ;;LOOP ON THE PRESENT TEST
6$;*****END OF CODE FOR THE XOR TESTER*****
BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
BEQ 2$ ;;BR IF NO
CMPB 2$SWR,$STNM ;;ON THE RIGHT TEST? SWR<7:0>
BEQ $OVER ;;BR IF YES
2$: 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
7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
BR $OVER
4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
BR 1$ ;;ESCAPE TO THE NEXT TEST
3$: BIT #BIT11,$SWR ;;INHIBIT ITERATIONS?
BNE 1$ ;;BR IF YES
TST $PASS ;;IF FIRST PASS OF PROGRAM
BEQ 1$ ;;INHIBIT ITERATIONS
INC $ICNT ;;INCREMENT ITERATION COUNT
CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
BGE $OVER ;;BR IF MORE ITERATION REQUIRED
1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
$SVLAD: INCB $STNM ;;COUNT TEST NUMBERS
MOV $STNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
```

K09

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

SEQ 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	SOVER: 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  
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

014412  
014412 010046  
014414 010146  
014416 010246  
014420 010346  
014422 010546  
014424 012746 020200  
014430 016605 000020  
014434 100004  
014436 005405  
014440 112766 000055 000001  
014446 005000  
014450 012703 014626  
014454 112723 000040  
014460 005002  
014462 016001 014616  
014466 160105  
014470 002402  
014472 005202  
014474 000774  
014476 060105  
014500 005702  
014502 001002  
014504 105716  
014506 100407  
014510 106316  
014512 103003  
014514 116663 000001 177777  
014522 052702 000060  
014526 052702 000040  
014532 110223  
014534 005720  
014536 020027 000010  
014542 002746  
014544 003002  
014546 010502  
014550 000764  
014552 105726  
014554 100003  
014556 116663 177777 177776  
014564 105013  
014566 012605  
014570 012603  
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?  
BPL 9\$ ;;BR IF NO  
9\$: MOVB -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING  
CLRB (R3) ;;SET THE TERMINATOR  
MOV (SP)+,R5 ;;POP STACK INTO R5  
MOV (SP)+,R3 ;;POP STACK INTO R3  
MOV (SP)+,R2 ;;POP STACK INTO R2



4928	014574	012601			MOV	(SP)+,R1	::POP STACK INTO R1
4929	014576	012600			MOV	(SP)+,R0	::POP STACK INTO R0
4930	014600	104401	014626		TYPE	\$DBLK	::NOW TYPE THE NUMBER
4931	014604	016666	000002	000004	MOV	2(SP),4(SP)	::ADJUST THE STACK
4932	014612	012616			MOV	(SP)+,(SP)	
4933	014614	000002			RTI		::RETURN TO USER
4934	014616	023420			\$DTBL:	10000.	
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  
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

014636	017646	000000	
014642	116667	000001	000211
014650	112667	000207	
014654	062716	000002	
014660	000406		
014662	112767	000001	000171
014670	112767	000006	000165
014676	112767	000005	000154
014704	010346		
014706	010446		
014710	010546		
014712	116704	000145	
014716	005404		
014720	062704	000006	
014724	110467	000132	
014730	116704	000125	
014734	016605	000012	
014740	005003		
014742	006105		
014744	000404		
014746	006105		
014750	006105		
014752	006105		
014754	010503		
014756	006103		
014760	105367	000076	
014764	100016		
014766	042703	177770	
014772	001002		
014774	005704		
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    N              ;;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    N              ;;CALL FOR TYPEOUT
*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    N              ;;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
*1$:     ROL    R5             ;;ROTATE MSB INTO "C"
*        BR    3$             ;;GO DO MSB
*2$:     ROL    R5             ;;FORM THIS DIGIT
*        ROL    R5
*        ROL    R5
*        ROL    R5
*3$:     MOV    R5,R3          ;;GET LSB OF THIS DIGIT
*        ROL    R3            ;;TYPE THIS DIGIT?
*        DECB   SOMODE         ;;BR IF NO
*        BPL   7$             ;;GET RID OF JUNK
*        BIC   #177770,R3     ;;TEST FOR 0
*        BNE   4$             ;;SUPPRESS THIS 0?
*        TST   R4             ;;BR IF YES
*        BEQ   5$
    
```

B10

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 120  
DVDVA.A.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		MOVB	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  
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

.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	
ACDW1 = 000000	1008			
ACDW2 = 000000	1008			
ACPUOP = 000000	1008	1023		
ADDRES 012120	4360*	4371*	4381	4388#
ADDW0 = 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 = 000000	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		
AMSLG = 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#		
APTENV = 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		
AUSWR = 071110	3#	1008	1022	
AVECT1 = 000300	3#	1008	1047	
AVECT2 = 000000	1008	1048		
BADBRK 011070	4034#			
BAUD = 007400	912#			
BAUDRA 007124	3265#			



















K10

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	4192#	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#												
SERROR	013732												
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	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	3333	3337	3375	3466
	3470	3521	3541	3577	3581	3685	3712	3785	3882	3922	3954	3958	3978
SFSBAD=	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
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	3792	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

M10

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

MACY11 27(1006)  
06-MAY-77 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						
\$GDADR 001120	979#												
\$GDOAT 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#
	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#	2887#	2895#	2905#	2918#	2934#	2941#	2948#	2958#
	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	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#	2887	2895#	2905	2918#	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
	4354	4375#											
\$ISK1 = 000001	2921#	2932	3201#	3216	3335#	3348	3468#	3481	3579#	3592	3956#	3964	4340#
\$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	2485	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	3690	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							
	38	1145	1146	1148	1149	1152	1153	1155	1156	1157	1158	1160
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

\$LSTIN= 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	2726	2727	2728	2747	2748	2750	2751	2754	2755	2760	2761
2762	2763	2764	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	3369	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	4196	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

\$LSTST= 177777

\$LSTTA= 000000



CROSS REFERENCE TABLE -- USER SYMBOLS

		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	3690	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#											
SMSGAD	001210	1016#	4717*	4720										
SMSGLG	001212	1017#	4722*											
SMSGTY	001174	1010#	4715	4723*	4735	4739*								
SMSWR	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#





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

MACY11 27(1006)  
15:29

16-MAY-77 10:36 PAGE 139  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0136

SSAVLE= 177777

3# 2936# 2937# 3147# 3151# 3189# 3190# 3641# 3645# 3739# 3743# 4146# 4147#  
4203# 4207# 4243# 4247# 4251# 4255#

SSAVRE= \*\*\*\*\* U

5058  
4449# 4457 4458# 4459# 4475#

SSAVR6 012414

1088 4822#

SSCOPE 014132

1079# 1087 1088 1090 1092 1094 1096 1097 1098 1100 1127 1130 4408

SSETUP= 000137

4563 4697 4769 4795 4803 4823

SSSKO = 000233

2936# 2937 3147# 3151 3189# 3190 3641# 3645 3739# 3743 4146# 4147 4203#

SSTUP = 177777

1079#  
4833 4862#

SSVLAD 014340

932# 937

SSVPC = 000204

3# 680 684 685 686 687 688 689 690 998 999 1000 1097

SSWR = 167400

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 4792 4796 4808

4814 4815 4816 4817 4818 4824 4836 4838 4839 4842 4843 4844 4851

SSWREG 001216

4852 4853 4865 4868 4871

SSWRMK= 000000

1021# 1121 4818 4819 4840

STAGLE= 177777

690 691 1199# 1210# 1214# 1222# 1255# 1259# 1271# 1275# 1288# 1292# 1307# 1311#

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#

STAGNU= 000246

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# 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#



# 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#	1335#	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#	4118#	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	4643	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											
\$TRP = 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#



	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											
STYPOB 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#	2242#	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#						
SSARGC= 000000	4071#	4178#	4227#	4288#	4326#								
SSBYTE= 000403	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#	2767#	2844#	2877#	2895#	2914#	2918#	2921#	2941#	2958#	3014#	3032#



K11

MAINDEC-11-DVDVA-A  
DVDVAA.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  
SSLOC = 012100

SSLOCN= 000000  
SSRETU= 000000  
SSRTN1= 000237  
SSRTN2= 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#	3174#	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#	4288#	4321#	4326#	4385#	4392#
4071#	4162#	4172#	4178#	4221#	4227#	4266#	4288#	4322#	4326#	4393#		

\$\$\$SRC = 000027  
\$\$TO = 000000

3230#	4195#												
1148#	1149	2403#	2406	2502#	2505	2530#	2533	2553#	2556	2613#	2616	2635#	
2638	2676#	2682	2719#	2725	2760#	2766	2819#	2825	2993#	2996	3003#	3006	
3075#	3078	3324#	3327	3368#	3371	3445#	3451	3457#	3460	3513#	3516	3568#	
3571	3632#	3633	3634	3729#	3730	3731	3759#	3762	3814#	3815	3816	4139#	
4142	4348#	4349											
4965*	4969*	4979	5014#										
1198#	1223												
1346	1351#												
2559	2564#												
2582	2588#												
2619	2624#												
2641	2646#												
2686	2691#												
2708	2713#												
2727	2732#												
2770	2775#												
1363	1368#												
2788	2793#												
2828	2833#												
2845	2850#												
2875#	2892												
2878	2883#												
2882	2887#												
2890	2893#												
2896	2905#												
1382	1387#												
2913#	2936												
2915	2937#												
2919	2934#												
2922	2932#												
2942	2948#												
2959	2964#												
3015	3025#												
3033	3045#												
1412	1417#												
3054	3063#												
3082	3093#												
3097	3103#												
3129	3134#												
3145	3148#												
3146#	3226												
3150	3227#												
3168#	3189												
1428	1433#												
3170	3190#												
3173	3179#												
3178	3186#												
3194	3198#												
3197	3218#												
3202	3204	3208#											
3207	3216#												
3332	3350#												
1445	1450#												
3336	3342#												
3341	3348#												

\$OFILL 015061  
 \$1 002102  
 \$10 002430  
 \$100 005300  
 \$101 005340  
 \$102 005426  
 \$103 005470  
 \$104 005554  
 \$105 005602  
 \$106 005636  
 \$107 005730  
 \$11 002456  
 \$110 005752  
 \$111 006044  
 \$112 006062  
 \$113 006124  
 \$114 006144  
 \$115 006150  
 \$116 006170  
 \$117 006212  
 \$12 002506  
 \$120 006212  
 \$121 006256  
 \$122 006254  
 \$123 006254  
 \$124 006270  
 \$125 006306  
 \$126 006424  
 \$127 006452  
 \$13 002556  
 \$130 006474  
 \$131 006544  
 \$132 006556  
 \$133 006662  
 \$134 006712  
 \$135 006710  
 \$136 007060  
 \$137 006750  
 \$14 002604  
 \$140 007004  
 \$141 006772  
 \$142 007002  
 \$143 007016  
 \$144 007042  
 \$145 007040  
 \$146 007042  
 \$147 007332  
 \$15 002632  
 \$150 007330  
 \$151 007332

\$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#



N11

MAINDEC-11-DVDVA-A MACY11 27(1006) 16-MAY-77 10:36 PAGE 146  
DVDVA.P11 06-MAY-77 15:29 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0143

\$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#														
BGNHRD	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											
COMMENT	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#													
ENDDEC	1#														
ENDDO	1#	2935	3188												
ENDHRD	3#														
ENDHW	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#														
ERRDF	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	2181	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	1228	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	1728	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	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
	3155	3010	3022	3030	3042	3049	3067	3072	3090	3131	3137	3139	3144	3147	3152
	3305	3158	3161	3164	3166	3176	3221	3223	3234	3291	3295	3301	3302	3303	3304
	3496	3307	3354	3357	3387	3390	3391	3415	3416	3417	3418	3419	3422	3424	3440
	3736	3502	3537	3543	3549	3600	3604	3607	3608	3638	3641	3658	3668	3714	3733
	3960	3739	3756	3772	3834	3837	3839	3843	3845	3849	3852	3855	3884	3886	3914
	4243	3962	4014	4095	4097	4099	4111	4116	4126	4192	4200	4203	4215	4238	4240
\$CKOP2	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	1441	1162	1165	1168	1171	1217	1219	1228	1266	1283	1300	1342	1359	1376	1424
	1825	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
	2428	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
	3229	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3182	3184
	3563	3298	3310	3320	3360	3388	3427	3430	3454	3486	3498	3505	3532	3552	3560
	3917	3565	3596	3605	3622	3676	3679	3720	3720	3776	3779	3861	3865	3868	3910
	4371	3924	3943	3947	3967	3973	3980	4008	4011	4144	4194	4208	4210	4281	4369
\$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	1208	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	1463	1495
	1795	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750	1769
	2157	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114	2137
	2478	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439	2456
	2941	2511	2535	2558	2581	2618	2640	2707	2787	2844	2877	2895	2914	2918	2921
	3373	2958	3014	3032	3053	3081	3096	3128	3169	3172	3193	3201	3203	3331	3335
	4023	3464	3468	3519	3539	3575	3579	3683	3710	3783	3880	3920	3952	3956	3976
\$COMPA	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	1511	1208	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	1463	1495
	1795	1527	1548	1561	1577	1594	1613	1639	1655	1672	1691	1717	1733	1750	1769
	2157	1811	1828	1847	1879	1913	1946	1976	1991	2026	2049	2069	2089	2114	2137
	2478	2177	2202	2225	2245	2265	2290	2313	2333	2353	2380	2412	2430	2439	2456
	2895	2511	2535	2558	2581	2618	2640	2686	2707	2727	2770	2787	2828	2844	2877
	3193	2914	2918	2921	2941	2958	3014	3032	3053	3081	3096	3128	3144	3169	3172
	3783	3201	3331	3335	3373	3464	3468	3519	3539	3575	3579	3638	3683	3710	3736
\$COUNT	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
	3324	3880	3920	3952	3956	3976	4023	4108	4122	4154	4200	4240	4248	4337	4340
\$DO	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	4131													
	1770	1209	1223	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	2138	1512	1528	1549	1562	1578	1595	1614	1640	1656	1673	1692	1718	1734	1751
	2457	1796	1812	1829	1848	1880	1914	1947	1977	1992	2027	2050	2070	2090	2115
	2878	2158	2178	2203	2226	2246	2266	2291	2314	2334	2354	2381	2413	2431	2440
		2479	2512	2536	2559	2582	2619	2641	2686	2708	2727	2770	2788	2828	2845
		2882	2890	2892	2896	2915	2919	2922	2936	2942	2959	3015	3033	3054	3082

\$GENTA	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
	4241	4246	4249	4254	4256	4259	4338	4341	4345	4364	4378	4385			
	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	4105	4114	4118	4128	4147	4159	4170	4172	4202	4204	4213	4220	4221	4242
	4244	4250	4252	4257	4260	4265	4266	4321	4322	4335	4346	4354	4365	4375	4392
	4393														
\$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	1842	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



CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0151

	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														
SLPCNT	1#	3144	3638	3736	4200	4240	4248								
SOPADD	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			
SOPAND	1#	3230	4195												
SOPCD1	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					
SOPCD2	1#	4208	4215	4369											
SOPCOD	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					
SOPCOM	1#	4215													
SOPDEF	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	3884	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	4157	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#  
SOPNAN 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	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	3152	3155
3158	3161	3164	3166	3176	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	3658	3668	3714	3733	3736	3756
3772	3834	3837	3839	3843	3845	3849	3852	3855	3884	3886	3914	3960	3962	4014
4095	4097	4099	4111	4116	4126	4192	4200	4238	4240	4248	4343	4352	4356	4358
4360	4362	4367	4381											

SOPR1 1#  
SOPR2 1#

1441	1162	1165	1168	1171	1217	1219	1228	1266	1283	1300	1342	1359	1376	1424
1825	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
2428	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
3229	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3182	3184
3563	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	4371	4373	4383												
\$OPSLB	1#	4208													
\$OPSMB	1#	4144													
\$OPXOR	1#														
\$OR	1#														
\$PUT	1#	2404	2503	2531	2554	2614	2636	2677	2720	2761	2820	2994	3004	3076	3325
\$STRUC	3369	3446	3458	3514	3569	3760	4140								
\$SUBON	1#														
	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
\$THEN	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#														
\$SMILE	1#	2913	3168												
\$SCHRE	961#														
\$SCHTM	961#														
\$SDEFA	1#														
\$SENDS	1#														
\$SERRO	1#														
\$SESCA	1#														
\$SGEN	1#	802#													
	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	2361	2387	2418	2436	2445
	2875	2485	2517	2541	2564	2588	2624	2646	2691	2713	2732	2775	2793	2833	2850
	3103	2883	2887	2893	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
	4221	4070	4105	4114	4118	4128	4147	4159	4170	4172	4177	4202	4204	4213	4220
	4346	4226	4242	4244	4250	4252	4257	4260	4265	4266	4287	4321	4322	4325	4335
\$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	4255	4256	4257
	4259	4260	4265	4321	4345	4346	4354	4364	4365	4375	4377	4392			
\$\$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#	3703	3804	3998	4041										
	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															
\$\$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
	2458	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															
\$\$SELE	1#														
\$\$SET	5039#	5048	5049	5050	5051	5053	5055	5056	5057						
\$\$SETM	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