





B01

EOF102FFP8S82411  
DVDVCA.P11

08-AUG-77 09:16

0000000011-DVDVCA20 MACY11 27(108891008)AUG-77 09128DR2000VBASEQ

00010000

770920  
SEQ 0001

.REM 2

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVDVC-A-D  
PRODUCT NAME: DLV11-F OFFLINE TEST  
PRODUCT DATE: AUGUST, 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



35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72

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 |



73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
1261.0 GENERAL PROGRAM INFORMATION.  
-----

## 1.1 PROGRAM PURPOSE (ABSTRACT).

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE DLV11-F SERIAL LINE INTERFACE. 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-F SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

177560 -CONSOLE INTERFACE DEVICE ADDRESS  
175610 -FIRST SERIAL LINE ADDRESS OF 15 CONSECUTIVE SERIAL LINE DEVICES.

60 - VECTOR FOR CONSOLE DEVICE INTERFACE.  
300 - VECTOR FOR FIRST OF 15 DEVICES.

THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 4K OF MEMORY AND A DLV11-F (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.

## HARDWARE REQUIREMENTS:

ANY PDP-11 FAMILY PROCESSOR  
4K MEMORY - MINIMUM  
A SPECIAL WRAP CONNECTOR OR EQUIVALENT (OPTIONAL)

## 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.



127  
128  
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

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

### 1.5 ASSUMPTIONS.

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

### 2.0 OPERATING INSTRUCTIONS.

-----

#### 2.1 LOADING AND STARTING PROCEDURES.

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

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

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

#### 2.2 SPECIAL ENVIRONMENTS.

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

#### 2.3 OPERATIONAL SWITCH SETTINGS

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

#### CONTROL:

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

- 1) TYPE CONTROL G (<G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE



182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236

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 TIMEOUTS
- 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-F'S. IT REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT 'SBASE') AND ITS INTERRUPT VECTOR (STORED AT 'SVECT1'); AND WILL BE ABLE TO ADDRESS ANY DLV11-F STARTING AT THE SPECIFIED BASE ADDRESS UP TO 16 CONSECUTIVE DEVICES.

EXAMPLES:       SBASE: 175610  
                  SVECT1: 300

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

SBASE AND SVECT1 DEFAULT TO 175610 AND 300 RESPECTIVELY.



237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279

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#14 -- BASE ADDRESS + (160)  
BASE VECTOR + (160)

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

| BIT 15   | BIT 14   | - | - | - | BIT 1    | BIT 0    |
|----------|----------|---|---|---|----------|----------|
| ! CON- ! | ! UNIT ! |   |   |   | ! UNIT ! | ! UNIT ! |
| ! SOLE ! | ! 14 !   |   |   |   | ! #1 !   | ! #0 !   |

A BIT MAP CAN BE ENTERED AT '\$DEVN' PRIOR TO STARTING THE PROGRAM.

EXAMPLE:  
\$BASE: 175610  
\$VECTOR: 300  
\$DEVN: 100013

THE PROGRAM WILL TEST-

|         |        |     |
|---------|--------|-----|
| UNIT#0  | 175610 | 300 |
| UNIT#1  | 175620 | 310 |
| UNIT#3  | 175640 | 330 |
| CONSOLE | 177560 | 60  |

OPTIONS

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



280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330

THE DEFAULT VALUE OF SUSWR IS AS FOLLOWS:

| <u>BIT POSITION</u> | <u>DEFINITION</u>                        | <u>DEFAULT VALUE</u> |
|---------------------|--|----------------------|
| 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<br>(SEE FOLLOWING NOTE) | 02(8) = 110 BAUD     |
| 12                  | BREAK GENERATION ENABLED                 | 1 = YES              |
| 13                  | WRAP CONNECTOR INSTALLED                 | 0 = NO               |
| 14                  | MAINT JUMPER                             | 0 = NO               |
| 15                  | ERROR BITS ENABLED                       | 0 = NO               |

NOTE ON BITS <7:11>

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.(TEST #16)

NOTE ON BIT 14

THIS SWITCH WHEN ON WILL ALLOW THE DIAGNOSTIC TO TEST IN MAINTAINCE MODE. IT IS ASSUMED THAT THE MAINTAINCE JUMPER IS INSTALLED ON ALL OF THE DLV11-F MODULES WHEN THIS BIT IS SET.

2.5 EXECUTION TIMES.

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

FIRST PASS- 2 MINUTES  
ADDITIONAL PASSES 2 MINUTES  
ADDITIONAL DEVICES 2 MINUTES

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



331  
332  
333  
334  
335  
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

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:

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

WHERE ALL VALUES TYPED ARE OCTAL.  
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.



383  
384  
385  
386  
387  
388  
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

NOTE

THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.

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

5.0 "ENDPASS\*\*\*\*\*."  
DEVICE INFORMATION TABLES.  
-----



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-F RESPONDS TO THAT ADDRESS SPACE.



K01

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

THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS

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

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

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

TEST 5 RCVRIE - RCSR6 SET, CLEAR, RESET  
---- -

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

TEST 6 RCVRDONE - RCSR 7 - IS CLEARED BY INIT  
---- --

TEST 7 RCVRACT - RCSR 11 - 15 CLEARED BY INIT  
---- -

TEST 10 XMITRDY - TCSR 7 - IS SET BY INIT  
---- --

TEST 11 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED  
---- --  
WITH A CHARACTER AND THAT IT SETS WITHIN A  
REASONABLE AMOUNT OF TIME.

TEST 12 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 13 RCVRDONE IS CLEARED BY READING RBUF  
---- --



L01

484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536

TEST 14 RCVRACT - RCSR 11 - SETS WHEN A START BIT IS  
-----  
RECEIVED AND CLEARS WHEN RCVRDONE - RCSR 7 -  
SETS

TEST 15 OVERRUN BIT - RBUF 14  
-----

TEST 16 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS  
-----  
AVAILABLE A COMPARISON WILL BE MADE TO SEE IF  
NEW TIME IS LESS THAN PREVIOUS.

TEST 17 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 20 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL  
-----  
OF THE RECEIVER SIDE OF THE INTERRUPT LOGIC IN  
CHARACTER MODE.

TEST 21 TEST ACTUAL DATA TRANSFERED NON-INTERRUPT  
-----  
MAINTENANCE BIT SET

TEST 22 TEST DATA THROUGH WRAP  
-----

TEST 23 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE  
-----  
MODE.

TEST 24 TEST BREAK GENERATION LOGIC TRANSMIT KNOWN CHAR  
-----  
WITH BREAK SET AND COMPARE RECEIVED WITH 0.

TEST 25 NOT A TEST - SEND BACK TO LOOP  
-----

NOTE

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

ROUTINE:TIMER  
-----

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

INPUTS:

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

OUTPUT:

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

ROUTINE:DATLNG  
-----

THIS ROUTINE SETS UP A MASK FOR DATA, WITH -

INPUT:

NOTHING IS PASSED TO THIS ROUTINE BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:  
\$USWR-- THE WORD FOR SOFTWARE PARAMETERS  
DATA-- A MASK FOR THE LOCATION OF THE OCTAL NUMBER OF DATA BITS

OUTPUT----

MASK-- A MASK OF BINARY ZEROS RIGHT-JUSTIFIED THE NUMBER OF WHICH IS DEFINED IN \$USWR WORD.

ROUTINE:WAIT  
-----

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

SERVICE ROUTINE: INTSRV  
-----

THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT

537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590



NO1

591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604

'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-F UNDER TEST. ADRS +2 TO  
POINT TO THE VECTOR OF THE DLV11-F UNDER TEST.  
IT KEEPS TRACK OF THE CURRENT DEVICE AND BIT  
MASKS.

```

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
655
656
657
658
659
660

```

```

@
.TITLE MAINDEC-11-DVDVC-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

```

```

001100
000011
000012
000015
000200
177776
177774
177772
177570
177570
000000
000001
000002
000003
000004
000005
000006
000007
000006
000007
000000

```



661 000040  
662 000100  
663 000140  
664 000200  
665 000240  
666 000300  
667 000340

PR1= 40 ;: PRIORITY LEVEL 1  
PR2= 100 ;: PRIORITY LEVEL 2  
PR3= 140 ;: PRIORITY LEVEL 3  
PR4= 200 ;: PRIORITY LEVEL 4  
PR5= 240 ;: PRIORITY LEVEL 5  
PR6= 300 ;: PRIORITY LEVEL 6  
PR7= 340 ;: PRIORITY LEVEL 7

668  
669  
670 100000  
671 040000  
672 020000  
673 010000  
674 004000  
675 002000  
676 001000  
677 000400  
678 000200  
679 000100  
680 000040  
681 000020  
682 000010  
683 000004  
684 000002  
685 000001

.\*"SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000  
SW14= 40000  
SW13= 20000  
SW12= 10000  
SW11= 4000  
SW10= 2000  
SW09= 1000  
SW08= 400  
SW07= 200  
SW06= 100  
SW05= 40  
SW04= 20  
SW03= 10  
SW02= 4  
SW01= 2  
SW00= 1  
.EQUIV SW09,SW9  
.EQUIV SW08,SW8  
.EQUIV SW07,SW7  
.EQUIV SW06,SW6  
.EQUIV SW05,SW5  
.EQUIV SW04,SW4  
.EQUIV SW03,SW3  
.EQUIV SW02,SW2  
.EQUIV SW01,SW1  
.EQUIV SW00,SW0

686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698 100000  
699 040000  
700 020000  
701 010000  
702 004000  
703 002000  
704 001000  
705 000400  
706 000200  
707 000100  
708 000040  
709 000020  
710 000010  
711 000004  
712 000002  
713 000001  
714  
715  
716

.\*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000  
BIT14= 40000  
BIT13= 20000  
BIT12= 10000  
BIT11= 4000  
BIT10= 2000  
BIT09= 1000  
BIT08= 400  
BIT07= 200  
BIT06= 100  
BIT05= 40  
BIT04= 20  
BIT03= 10  
BIT02= 4  
BIT01= 2  
BIT00= 1  
.EQUIV BIT09,BIT9  
.EQUIV BIT08,BIT8  
.EQUIV BIT07,BIT7

```

717 .EQUIV BIT06,BIT6
718 .EQUIV BIT05,BIT5
719 .EQUIV BIT04,BIT4
720 .EQUIV BIT03,BIT3
721 .EQUIV BIT02,BIT2
722 .EQUIV BIT01,BIT1
723 .EQUIV BIT00,BIT0

```

```

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

```

```

726 000004
727 000010
728 000014
729 000014
730 000014
731 000020
732 000024
733 000030
734 000034
735 000060
736 000064
737 000240
738 000004
739 000001
740 000002
741 000003
742 000001
743 000002
744 000003
745 000002
746 000004
747 175610

```

```

ILLMEM= 4
ADRS= R1
GOOD= R2
BAD= R3
REGISTER=R1
BIT= R2
FUNCT= R3
LEAD= R2
FOLLOW= R4
DLADDR= 175610

```

```

: THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
SET= -1
CLR= 0

```

```

749 177777
750 000000

```

```

; *****
; RCSR REGISTER BIT NAMES
; *****

```

```

757 : UNUSED BIT15
758 : UNUSED BIT14
759 : UNUSED BIT13
760 : UNUSED BIT12
761 004000 RCVRACT= BIT11 ; RECEIVER ACTIVE INDICATOR
762 : UNUSED BIT10
763 : UNUSED BIT09
764 : UNUSED BIT08
765 000200 RCVRDONE= BIT07 ; RECEIVER DONE
766 000100 RCVRIE= BIT06 ; RECEIVER INTERRUPT ENABLE
767 : UNUSED BIT05
768 : UNUSED BIT04
769 : UNUSED BIT03
770 : UNUSED BIT02
771 : UNUSED BIT01
772 000001 RDRRUN= BIT00 ; READER RUN

```



773  
774  
775  
776  
777 100000  
778 040000  
779 020000  
780 010000  
781  
782  
783  
784  
785 000200  
786 000100  
787 000040  
788 000020  
789 000010  
790 000004  
791 000002  
792 000001  
793  
794  
795  
796  
797 100000  
798 040000  
799 020000  
800 010000  
801 004000  
802  
803  
804  
805  
806 000200  
807 000100  
808  
809  
810  
811 000004  
812  
813 000001  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827 000200  
828 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      ;
RDATA4=         BIT04      ;
RDATA3=         BIT03      ;
RDATA2=         BIT02      ;
RDATA1=         BIT01      ;
RDATA0=         BIT00      ;

```

RECEIVED DATA BITS

```

;*****
; TCSR REGISTER BIT NAMES
;*****
PBAUD3=         BIT15      ;
PBAUD2=         BIT14      ;
PBAUD1=         BIT13      ;
PBAUD0=         BIT12      ;
PBAUDSET=       BIT11      ;
; 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      ;

```

|     |        |         |       |   |                           |
|-----|--------|---------|-------|---|---------------------------|
| 829 | 000040 | TDATA5= | BIT05 | : | \ TRANSMITTER DATA BUFFER |
| 830 | 000020 | TDATA4= | BIT04 | : |                           |
| 831 | 000010 | TDATA3= | BIT03 | : |                           |
| 832 | 000004 | TDATA2= | BIT02 | : |                           |
| 833 | 000002 | TDATA1= | BIT01 | : |                           |
| 834 | 000001 | TDATA0= | BIT00 | : | /                         |

```

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

```

|     |        |           |     |
|-----|--------|-----------|-----|
| 840 | 000017 | DATA =    | 17  |
| 841 | 000020 | PARITY =  | 20  |
| 842 | 000040 | EVENODD = | 40  |
| 843 | 000100 | COMSPD =  | 100 |
| 844 | 000200 | PBR =     | 200 |

```

; BAUDE MUST BE ON THE UPPER
; BYTE BOUNDRY OF SUSMR.--4 BITS
BAUD = 7400
BRK = 10000
WRAP = 20000
MAINTJUMP = 40000
ERRBITS = 100000

```

```

;*****
.SBTTL TRAP CATCHER

```

|     |        |       |
|-----|--------|-------|
| 856 | 000000 | . = 0 |
| 857 |        | ;     |
| 858 |        | ;     |
| 859 |        | ;     |

```

;#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

```

|     |        |                  |  |
|-----|--------|------------------|--|
| 860 | 000174 | . = 174          |  |
| 861 | 000174 | DISPREG: .WORD 0 | ;; SOFTWARE DISPLAY REGISTER                   |
| 862 | 000176 | SWREG: .WORD 0   | ;; SOFTWARE SWITCH REGISTER                    |
| 863 |        | .SBTTL           | STARTING ADDRESS(ES)                           |
| 864 | 000200 | JMP              | 2#START ;; JUMP TO STARTING ADDRESS OF PROGRAM |

775



865  
866  
867  
868  
869 000204  
870 000046  
871 000046 011552  
872 000052  
873 000052 000000  
874 000204  
875 001000  
876  
877  
878  
879  
880  
881 001000  
882 000024  
883 000024 000200  
884 000044  
885 000044 001000  
886 001000  
887  
888  
889  
890  
891 001000  
892 001000 000000  
893 001002 001174  
894 001004 000005  
895 001006 000055  
896 001010 000036  
897 001012 000030

.SBTTL ACT11 HOOKS

;;\*\*\*\*\*  
;HOOKS REQUIRED BY ACT11

SSVPC=. ;SAVE PC  
.=46  
SENDAD ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP  
.=52  
.WORD 0 ;;2)SET LOC.52 TO ZERO  
.=SSVPC ;; RESTORE PC  
.=1000  
.SBTTL APT PARAMETER BLOCK

;;\*\*\*\*\*  
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
;\*\*\*\*\*

.\$X=. ;: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  
.=.\$X ;:RESET LOCATION COUNTER

;;\*\*\*\*\*  
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
;INTERFACE SPEC.

\$APTHD:  
\$HIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
\$MADR: .WORD \$MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)  
\$TSTM: .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  
SETEND-\$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

898  
899  
900  
901  
902  
903  
904  
905 001100  
906 001100 000000  
907 001102 000  
908 001103 000  
909 001104 000000  
910 001106 000000  
911 001110 000000  
912 001112 000000  
913 001114 000  
914 001115 001  
915 001115 000000  
916 001120 000000  
917 001122 000000  
918 001124 000000  
919 001126 000000  
920 001130 000000  
921 001132 000000  
922 001134 000  
923 001135 000  
924 001136 000000  
925 001140 177570  
926 001142 177570  
927 001144 177560  
928 001146 177562  
929 001150 177564  
930 001152 177566  
931 001154 000  
932 001155 002  
933 001156 012  
934 001157 000  
935 001160 000000  
936 001162 000000  
937 001164 177607 000377  
938 001170 077  
939 001171 015  
940 001172 000012  
941  
942  
943  
944  
945  
946 001174  
947 001174 000000  
948 001176 000000  
949 001200 000000  
950 001202 000000  
951 001204 000000  
952 001206 000000  
953 001210 000000

.SBTTL COMMON TAGS

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

SCMTAG: .=1100 ;;START OF COMMON TAGS  
STSTNM: .WORD 0 ;;CONTAINS THE TEST NUMBER  
SERFLG: .BYTE 0 ;;CONTAINS ERROR FLAG  
SICNT: .WORD 0 ;;CONTAINS SUBTEST ITERATION COUNT  
SLPADR: .WORD 0 ;;CONTAINS SCOPE LOOP ADDRESS  
SLPERR: .WORD 0 ;;CONTAINS SCOPE RETURN FOR ERRORS  
SERTTL: .WORD 0 ;;CONTAINS TOTAL ERRORS DETECTED  
SITEMB: .BYTE 0 ;;CONTAINS ITEM CONTROL BYTE  
SERMAX: .BYTE 1 ;;CONTAINS MAX. ERRORS PER TEST  
SERRPC: .WORD 0 ;;CONTAINS PC OF LAST ERROR INSTRUCTION  
SGDADR: .WORD 0 ;;CONTAINS ADDRESS OF 'GOOD' DATA  
SBDADR: .WORD 0 ;;CONTAINS ADDRESS OF 'BAD' DATA  
SGDDAT: .WORD 0 ;;CONTAINS 'GOOD' DATA  
SBDAT: .WORD 0 ;;CONTAINS 'BAD' DATA  
 ;;RESERVED--NOT TO BE USED  
SAUTOB: .BYTE 0 ;;AUTOMATIC MODE INDICATOR  
SINTAG: .BYTE 0 ;;INTERRUPT MODE INDICATOR  
SWR: .WORD DSWR ;;ADDRESS OF SWITCH REGISTER  
DISPLAY: .WORD DDISP ;;ADDRESS OF DISPLAY REGISTER  
STKS: 177560 ;;TTY KBD STATUS  
STKB: 177562 ;;TTY KBD BUFFER  
STPS: 177564 ;;TTY PRINTER STATUS REG. ADDRESS  
STPB: 177566 ;;TTY PRINTER BUFFER REG. ADDRESS  
SNLL: .BYTE 0 ;;CONTAINS NULL CHARACTER FOR FILLS  
SFILLS: .BYTE 2 ;;CONTAINS # OF FILLER CHARACTERS REQUIRED  
SFILLC: .BYTE 12 ;;INSERT FILL CHARS. AFTER A "LINE FEED"  
STPFLG: .BYTE 0 ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)  
STIMES: 0 ;;MAX. NUMBER OF ITERATIONS  
SESCAPE: 0 ;;ESCAPE ON ERROR ADDRESS  
SBELL: .ASCIZ <207><377><377> ;;CODE FOR BELL  
SQUES: .ASCII /?/ ;;QUESTION MARK  
SCRFL: .ASCII <15> ;;CARRIAGE RETURN  
SLF: .ASCIZ <12> ;;LINE FEED

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

\*\*\*\*\*  
.EVEN ;;APT MAILBOX  
SMAIL: ;;MESSAGE TYPE CODE  
SMSGTY: .WORD AMSGTY ;;FATAL ERROR NUMBER  
SFATAL: .WORD AFATAL ;;TEST NUMBER  
STESTN: .WORD ATESTN ;;PASS COUNT  
SPASS: .WORD APASS ;;DEVICE COUNT  
SDEVCT: .WORD ADEVCT ;;I/O UNIT NUMBER  
SUNIT: .WORD AUNIT ;;MESSAGE ADDRESS  
SMSGAD: .WORD AMSGAD



|     |        |        |                |        |  |
|-----|--------|--------|----------------|--------|--|
| 954 | 001212 | 000000 | \$MSGLG: .WORD | AMSGLG | :: MESSAGE LENGTH  |
| 955 | 001214 |        | SETABLE:       |        | :: APT ENVIRONMENT TABLE                                   |
| 956 | 001214 | 000    | SENV: .BYTE    | AENV   | :: ENVIRONMENT BYTE  |
| 957 | 001215 | 000    | SENVH: .BYTE   | AENVH  | :: ENVIRONMENT MODE BITS                                   |
| 958 | 001216 | 000000 | SSWREG: .WORD  | ASWREG | :: APT SWITCH REGISTER                                     |
| 959 | 001220 | 011110 | SUSWR: .WORD   | AUSWR  | :: USER SWITCHES   |
| 960 | 001222 | 000000 | SCPUOP: .WORD  | ACPUOP | :: CPU TYPE, OPTIONS                                       |
| 961 |        |        | ::             |        | BITS 15-11=CPU TYPE  |
| 962 |        |        | ::             |        | 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05           |
| 963 |        |        | ::             |        | 11/70=06, PDQ=07, Q=10                                     |
| 964 |        |        | ::             |        | BIT 10=REAL TIME CLOCK                                     |
| 965 |        |        | ::             |        | BIT 9=FLOATING POINT PROCESSOR                             |
| 966 |        |        | ::             |        | BIT 8=MEMORY MANAGEMENT                                    |
| 967 | 001224 | 000    | \$MAMS1: .BYTE | AMAMS1 | :: HIGH ADDRESS, M.S. BYTE                                 |
| 968 | 001225 | 000    | \$MTYP1: .BYTE | AMTYP1 | :: MEM. TYPE, BLK#1  |
| 969 |        |        | ::             |        | MEM. TYPE BYTE -- (HIGH BYTE)                              |
| 970 |        |        | ::             |        | 900 NSEC CORE=001  |
| 971 |        |        | ::             |        | 300 NSEC BIPOLAR=002                                       |
| 972 |        |        | ::             |        | 500 NSEC MOS=003   |
| 973 | 001226 | 000000 | \$MADR1: .WORD | AMADR1 | :: HIGH ADDRESS, BLK#1                                     |
| 974 |        |        | ::             |        | MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE |
| 975 | 001230 | 000    | \$MAMS2: .BYTE | AMAMS2 | :: HIGH ADDRESS, M.S. BYTE                                 |
| 976 | 001231 | 000    | \$MTYP2: .BYTE | AMTYP2 | :: MEM. TYPE, BLK#2  |
| 977 | 001232 | 000000 | \$MADR2: .WORD | AMADR2 | :: MEM. LAST ADDRESS, BLK#2                                |
| 978 | 001234 | 000    | \$MAMS3: .BYTE | AMAMS3 | :: HIGH ADDRESS, M.S. BYTE                                 |
| 979 | 001235 | 000    | \$MTYP3: .BYTE | AMTYP3 | :: MEM. TYPE, BLK#3  |
| 980 | 001236 | 000000 | \$MADR3: .WORD | AMADR3 | :: MEM. LAST ADDRESS, BLK#3                                |
| 981 | 001240 | 000    | \$MAMS4: .BYTE | AMAMS4 | :: HIGH ADDRESS, M.S. BYTE                                 |
| 982 | 001241 | 000    | \$MTYP4: .BYTE | AMTYP4 | :: MEM. TYPE, BLK#4  |
| 983 | 001242 | 000000 | \$MADR4: .WORD | AMADR4 | :: MEM. LAST ADDRESS, BLK#4                                |
| 984 | 001244 | 000300 | \$VECT1: .WORD | AVECT1 | :: INTERRUPT VECTOR#1, BUS PRIORITY#1                      |
| 985 | 001246 | 000000 | \$VECT2: .WORD | AVECT2 | :: INTERRUPT VECTOR#2, BUS PRIORITY#2                      |
| 986 | 001250 | 175610 | \$BASE: .WORD  | ABASE  | :: BASE ADDRESS OF EQUIPMENT UNDER TEST                    |
| 987 | 001252 | 100000 | \$DEVN: .WORD  | ADEVN  | :: DEVICE MAP  |
| 988 | 001254 |        | SETEND:        |        |  |
| 989 |        |        | .MEXIT         |        |  |

990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015

001254  
  
001254 175610  
001256 000300  
001260 175610  
001262 175612  
001264 175614  
001266 175615  
001270 175616  
001272 000000  
001274 000020  
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 (SERRPC).  
;;\*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

SERRTB:

;; 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



```

1016 001336
1017
1018
1019 001336 012706 001100
1020 001342 005026
1021 001344 022706 001140
1022 001350 001374
1023 001352 012706 001100
1024
1025 001356 012737 013474 000020
1026 001364 012737 000340 000022
1027 001372 012737 013274 000030
1028 001400 012737 000340 000032
1029 001406 012737 014426 000034
1030 001414 012737 000340 000036
1031 001422 012737 011606 000024
1032 001430 012737 000340 000026
1033 001436 016767 010056 010046
1034 001444 005067 177510
1035 001450 005067 177506
1036 001454 112767 000001 177433
1037 001462 012767 001462 177416
1038 001470 012767 001470 177412
1039
1040
1041 001476 013746 000004
1042 001502 012737 001536 000004
1043 001510 012767 177570 177422
1044 001516 012767 177570 177416
1045 001524 022777 177777 177406
1046 001532 001012
1047
1048 001534 000403
1049 001536 012716 001544 64$:
1050 001542 000002
1051 001544 012767 000176 177366 65$:
1052 001552 012767 000174 177362
1053 001560 012637 000004 66$:
1054
1055 001564 005067 177412
1056 001570 132767 000200 177417
1057 001576 001403
1058 001600 012767 001216 177332
1059 001606
1060
1061
1062 001606 005227 177777
1063 001612 001037
1064 001614 022737 011552 000042
1065 001622 001433
1066 001624 104401 001672
1067
1068 001630 005737 000042
1069 001634 001012
1070 001636 126727 177352 000001
1071 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 $SPWRDN,$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
RTI
MOV $SWREG,$SWR ;;POINT TO SOFTWARE SWR
MOV $DISPREG,$DISPLAY
MOV (SP)+,$ERRVEC ;;RESTORE ERROR VECTOR
CLR $PASS ;;CLEAR PASS COUNT
BITB $APTSIZE,$SENVM ;;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

```

L02

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 25  
DVDVCA.P11 08-AUG-77 09:16 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0024

```
1072 001646 026727 177266 000176      CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1073 001654 001005                      BNE     71$            ;;BRANCH IF NO
1074 001656 104406                      GTSWR                      ;;GET SOFT-SWR SETTINGS
1075 001660 000403                      BR      71$
1076 001662 112767 000001 177244 70$:   MOVB   #1,$AUTOB      ;;SET AUTO-MODE INDICATOR
1077 001670                      71$:
1078 001670 000410                      BR      68$            ;;GET OVER THE ASCIZ
1079                      ;;69$: .ASCIZ <CRLF>*MD-11-DVDVC-A*<CRLF>
1080 001712                      68$:
```



```

1081 001712                               WHILE SDEVN EQ #0 DO
1082 001712
1083 001712 005767 177334          $1:   TST   SDEVN
1084 001716 001101                BNE   S2
1085 001720                        TYPTXT <<CRLF>!I HAVE NO DEVICE TO TEST.!)
1086 001762                        TYPTXT <<CRLF>!SET UP SDEVN TO INDICATE ACTUAL CONFIGURATION.!)
1087 002050                        TYPTXT <<CRLF>!TYPE PROCEED (P) TO CONTINUE.!)
1088 002116 000000                HALT
1089 002120
1090 002120 000674                BR    S1                               ENDDO
1091 002122
1092 002122          $2:
1093 002122 012767 000001 007122    MOV   #1,INITFLAG
1094 002130                LET   INITFLAG := #1
1095 002130 012767 100000 007112    MOV   #BIT15,BITMASK
1096 002136                LET   BITMASK := #BIT15 ; START AT CONSOLE
1097 002136          LOOP:
1098 002136 004767 006666          CALL  CYCLE ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1099                JSR   PC,CYCLE ;
1100                ; ADDR+2 -> NEXT VECTOR
1101 002142                ; GET UNIT ADDRESS
1102 002142 012167 177106          MOV   (ADRS)+,DLADD
1103                ; GET UNIT VECTOR
1104 002146                LET   DLVEC := (ADRS)
1105 002146 011167 177104          MOV   (ADRS),DLVEC
1106 002152                LET   ADRS := DLADD
1107 002152 016701 177076          MOV   DLADD,ADRS
1108                ; RCSR = DLADD + 0
1109 002156                LET   RCSR := DLADD
1110 002156 016767 177072 177074    MOV   DLADD,RCSR
1111 002164                LET   RBUF := DLADD + #2
1112 002164 016767 177064 177070    MOV   DLADD,RBUF
1113 002172 062767 000002 177062    ADD   #2,RBUF
1114 002200                LET   TCSR := DLADD + #4
1115 002200 016767 177050 177056    MOV   DLADD,TCSR
1116 002206 062767 000004 177050    ADD   #4,TCSR
1117 002214                LET   TCSRHI := DLADD + #5
1118 002214 016767 177034 177044    MOV   DLADD,TCSRHI
1119 002222 062767 000005 177036    ADD   #5,TCSRHI
1120 002230                LET   TBUF := DLADD + #6
1121 002230 016767 177020 177032    MOV   DLADD,TBUF
1122 002236 062767 000006 177024    ADD   #6,TBUF
1123 002244                LET   R5 := #RSSTACK
1124 002244 012705 001334          MOV   #RSSTACK,R5
1125                ; ;BRESET
1126 002250 000005                RESET
    
```

```

1127 .....
1128 .....
1129 .....
1130 .....
1131 .....
1132 .....
1133 002252 000004 .....
1134 002254 012767 000002 176676 .....
1135 002262 012767 000001 176710 .....
1136 002270 .....
1137 002270 016701 176760 .....
1138 .....
1139 002274 .....
1140 002274 010146 .....
1141 002276 012701 000004 .....
1142 002302 012721 011020 .....
1143 002306 012711 000340 .....
1144 002312 012601 .....
1145 002314 .....
1146 002314 005067 176752 .....
1147 002320 .....
1148 002320 .....
1149 002320 .....
1150 002320 012767 002326 176562 .....
1151 .....
1152 002326 .....
1153 002326 005067 006474 .....
1154 .....
1155 .....
1156 002332 005711 .....
1157 002334 .....
1158 002334 005767 006466 .....
1159 002340 001401 .....
1160 .....
1161 002342 .....
1162 002342 104001 .....
1163 002344 .....
1164 002344 .....
1165 002344 .....
1166 002344 .....
1167 002344 062767 000002 176720 .....
1168 002352 .....
1169 002352 016701 176676 .....
1170 002356 066701 176710 .....
1171 .....
1172 002362 026727 176704 000010 .....
1173 002370 001353 .....
1174 002372 .....
1175 002372 010146 .....
1176 002374 010246 .....
1177 002376 012701 000004 .....
1178 002402 010102 .....
1179 002404 062702 000002 .....
1180 002410 010221 .....
1181 002412 005011 .....
1182 002414 012602 .....

```

```

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

```



B03

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 28  
DVDVCA.P11 08-AUG-77 09:16 TI ADDRESSABILITY

SEQ 0027

1183 002416 012601  
1184  
1185 002420

MOV (SP)+,R1

;;POP STACK INTO R1  
ENDTST ;END OF TEST

```

1186 .....*****
1187 .....* THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
1188 .....*****
1189 .....*****
1190 .....*TEST 2          BREAK - TCSR0 SET, CLEAR, RESET
1191 .....*          THIS BIT IS THE ONLY ONE IN THIS POSITION
1192 .....*          THAT IS READ AND WRITE.
1193 .....*****
1194 002420 000004          TST2:  SCOPE
1195 002422 012767 000010 176530      MOV      #10,$TIMES          ;;DO 10 ITERATIONS
1196 002430 012767 000002 176542      MOV      #2,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
1197 .....
1198 .....          IF #BRK NOTSETIN SUSMR THEN
1199 002436 032767 010000 176554      BIT      #BRK,$USMR
1200 002444 001004          BNE      $5
1201 002446          .....          EXIT TEST ; BREAK NOT INSTALLED
1202 002446 012767 000001 176504      MOV      #1,$TIMES
1203 002454 000452          BR       TST3          ;;EXIT THIS TEST
1204 002456          .....          ENDIF
1205 002456          $5:
1206 .....
1207 .....          ; SEE IF IT IS CLEAR
1208 002456          .....          BGNSUB
1209 002456 012767 002464 176424      MOV      #64,$SLPERR
1210 .....
1211 .....          IF #BREAK SETIN @TCSR THEN
1212 002464 032777 000001 176572      BIT      #BREAK,@TCSR
1213 002472 001401          BEQ      $6
1214 .....
1215 .....          ; BREAK DID NOT RESET IN TCSR
1216 002474 104002          ERROR   2          ERRHRD 2,,DIDNOT
1217 002476          .....          ENDIF
1218 002476          $6:
1219 002476          .....          ENDSUB
1220 .....
1221 .....          ; TRY TO SET BREAK BIT
1222 002476          .....          BGNSUB
1223 002476 012767 002504 176404      MOV      #64,$SLPERR
1224 002504          .....          LET @TCSR := @TCSR SET.BY #BREAK
1225 002504 052777 000001 176552      BIS      #BREAK,@TCSR
1226 .....
1227 .....          IF ; STUCK TO 0
1228 002512 032777 000001 176544      BIT      #BREAK,@TCSR
1229 002520 001001          BNE      $7
1230 .....
1231 .....          ; BREAK DID NOT SET IN TCSR
1232 002522 104003          ERROR   3          ERRHRD 3,,DIDNOT
1233 002524          .....          ENDIF
1234 002524          $7:
1235 002524          .....          ENDSUB
1236 .....
1237 .....          ; TRY TO CLEAR A SET BIT
1238 002524          .....          BGNSUB
1239 002524 012767 002532 176356      MOV      #64,$SLPERR
1240 .....
1241 002532          .....          LET @TCSR := @TCSR CLR.BY #BREAK

```





E03

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 31  
DVDVCA.P11 08-AUG-77 09:16

T2 BREAK - TCSR0 SET, CLEAR, RESET

SEQ 0030

```

1275 ;:*****
1276 ;:*****
1277 ;:*****
1278 ;:TEST 3 MAINT - TCSR2 SET, CLEAR, RESET
1279 ;:*****
1279 002602 000004 TST3: SCOPE
1280 002604 012767 000010 176346 MOV #10,$TIMES ;;DO 10 ITERATIONS
1281 002612 012767 000003 176360 MOV #3,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1282
1283 002620 IF #MAINTJUMP NOTSETIN $USWR THEN
1284 002620 032767 040000 176372 BIT #MAINTJUMP,$USWR
1285 002626 001004 BNE $12
1286 002630 EXIT TEST
1287 002630 012767 000001 176322 MOV #1,$TIMES
1288 002636 000452 BR TST4 ;;EXIT THIS TEST
1289 002640 ENDF
1290 002640 $12:
1291
1292 ; SEE IF IT IS CLEAR
1293 002640 ; BGNSUB
1294 002640 012767 002646 176242 MOV #64,$SLPERR
1295
1296 002646 IF #MAINT SETIN @TCSR THEN
1297 002646 032777 000004 176410 BIT #MAINT,@TCSR
1298 002654 001401 BEQ $13
1299
1300 ; MAINT DID NOT RESET IN TCSR
1301 002656 104006 ERROR 6 ERRHRD 6,,DIDNOT
1302 002660 ENDF
1303 002660 $13:
1304 002660 ENDSUB
1305
1306 ; TRY TO SET MAINT BIT
1307 002660 ; BGNSUB
1308 002660 012767 002666 176222 MOV #64,$SLPERR
1309 002666 LET @TCSR := @TCSR SET.BY #MAINT
1310 002666 052777 000004 176370 BIS #MAINT,@TCSR
1311 ; STUCK TO 0
1312 002674 IF #MAINT NOTSETIN @TCSR THEN
1313 002674 032777 000004 176362 BIT #MAINT,@TCSR
1314 002702 001001 BNE $14
1315
1316 ; MAINT DID NOT SET IN TCSR
1317 002704 104007 ERROR 7 ERRHRD 7,,DIDNOT
1318 002706 ENDF
1319 002706 $14:
1320 002706 ENDSUB
1321
1322 ; TRY TO CLEAR A SET BIT
1323 002706 ; BGNSUB
1324 002706 012767 002714 176174 MOV #64,$SLPERR
1325
1326 002714 LET @TCSR := @TCSR CLR.BY #MAINT
1327 002714 042777 000004 176342 BIC #MAINT,@TCSR
1328 ; SHOULD HAVE CLEARED
1329 002722 IF #MAINT SETIN @TCSR THEN
1330 002722 032777 000004 176334 RIT #MAINT,@TCSR

```



```

1331 002730 001401          BEQ      $15
1332                                     ; MAINT DID NOT CLEAR INTCSR
1333 002732                                     ERRHRD 10,,DIDNOT
1334 002732 104010          ERROR    10
1335 002734                                     ENDIF
1336 002734          $15:                                     ENDSUB
1337 002734                                     ; NOW SEE IF RESET CLEARS IT
1338                                     BGNSUB
1339                                     ;
1340 002734 012767 002742 176146      MOV      @64$,SLPERR
1341 002734 052777 000004 176314      BIS      @MAINT,@TCSR
1342                                     LET      @TCSR := @TCSR SET.BY @MAINT
1343 002742 052777 000004 176314      BIS      @MAINT,@TCSR
1344                                     ; ISSUE BUS RESET
1345 002750                                     BRESÉT
1346 002750 000005          RESET
1347 002752                                     IF      @MAINT SETIN @TCSR THEN
1348 002752 032777 000004 176304      BIT      @MAINT,@TCSR
1349 002760 001401          BEQ      $16
1350                                     ; MAINT DID NOT RESET IN TCSR
1351                                     ERRHRD 11,,DIDNOT
1352 002762                                     ENDIF
1353 002762 104011          ERROR    11
1354                                     ENDSUB
1355 002764          $16:                                     ENDTST
1356 002764
1357 002764
1358
1359
1360

```

```

1361
1362
1363
1364
1365 002764 000004
1366 002766 012767 000010 176164
1367 002774 012767 000004 176176
1368
1369 003002 012746 000340
1370 003006 012746 003014
1371 003012 000002
1372 003014
1373
1374
1375 003014
1376 003014 012767 003022 176066
1377
1378 003022
1379 003022 032777 000100 176234
1380 003030 001401
1381
1382 003032
1383 003032 104012
1384 003034
1385 003034
1386 003034
1387
1388
1389 003034
1390 003034 012767 003042 176046
1391 003042
1392 003042 052777 000100 176214
1393
1394 003050
1395 003050 032777 000100 176206
1396 003056 001001
1397
1398 003060
1399 003060 104013
1400 003062
1401 003062
1402 003062
1403
1404
1405 003062
1406 003062 012767 003070 176020
1407
1408 003070
1409 003070 042777 000100 176166
1410
1411 003076
1412 003076 032777 000100 176160
1413 003104 001401
1414
1415 003106
1416 003106 104014

```

```

*****
*****
:TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
*****
TST4: 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 $17
; XMITIE DID NOT RESET IN TCSR
ERRHRD 12,,DIDNOT
ENDIF
$17:
ENDSUB
; TRY TO SET XMITIE BIT
BGNSUB
MOV #64$,$LPERR
LET @TCSR := @TCSR SET.BY #XMITIE
BIS #XMITIE,@TCSR
IF ; STUCK TO 0
#XMITIE NOTSETIN @TCSR THEN
; XMIT DID NOT RESET IN TCSR
ERRHRD 13,,DIDNOT
ENDIF
$20:
ENDSUB
; TRY TO CLEAR A SET BIT
BGNSUB
MOV #64$,$LPERR
LET @TCSR := @TCSR CLR.BY #XMITIE
BIC #XMITIE,@TCSR
IF ; SHOULD HAVE CLEARED
#XMITIE SETIN @TCSR THEN
; XMIT DID NOT CLEAR IN TCSR
ERRHRD 14,,DIDNOT

```



H03

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 34  
DVDVCA.P11 08-AUG-77 09:16 T4 XMITIE - TCSR6 SET, CLEAR, RESET

SEQ 0033

```

1417 003110                                ENDIF
1418 003110                                ENDSUB
1419 003110                                ; NOW SEE IF RESET CLEARS IT
1420                                     BGNSUB
1421
1422 003110 012767 003116 175772  MOV  #64$,SLPERR
1423 003110 052777 000100 176140  BIS  #XMITIE,@TCSR
1424
1425 003116                                LET  @TCSR := @TCSR SET.BY #XMITIE
1426 003116                                ; ISSUE BUS RESET
1427                                     BRESÉT
1428 003124                                IF  #XMITIE SETIN @TCSR THEN
1429 003124 000005  RESET
1430 003126                                ; XMIT DID NOT RESET IN TCSR
1431 003126 032777 000100 176130  BIT  #XMITIE,@TCSR
1432 003134 001401  BEQ  $22
1433                                     ERRHRD 15,,DIDNOT
1434 003136                                ENDIF
1435 003136 104015  ERROR  15
1436 003140                                ENDSUB
1437 003140                                ENDTST
1438 003140
1439 003140
1440
1441
1442

```

\$21:

\$22:

```

1443 ;*****
1444 ;*****
1445 ;TEST 5 RCVRIE - RCSR6 SET, CLEAR, RESET
1446 ; THIS BIT IS THE ONLY ONE IN THIS POSITION
1447 ; THAT IS READ AND WRITE.
1448 ;*****
1449 STS: SCOPE
1450 MOV #10,$TIMES ;DO 10 ITERATIONS
1451 MOV #5,$TESTN ;SET TEST NUMBER IN APT MAIL BOX
1452 ; SEE IF IT IS CLEAR
1453 BGNSUB
1454 MOV #64,$SLPERR
1455
1456 IF #RCVRIE SETIN @RCSR THEN
1457 BIT #RCVRIE,@RCSR
1458 BEQ $23
1459 ; RCVRIE DID NOT RESET IN RCSR
1460 ; ERRHRD 35,,DIDNOT
1461 ERROR 35
1462 ENDIF
1463 $23:
1464 ENDSUB
1465
1466 ; TRY TO SET RCVRIE BIT
1467 BGNSUB
1468 MOV #64,$SLPERR
1469 LET @RCSR := @RCSR SET.BY #RCVRIE
1470 BIS #RCVRIE,@RCSR
1471 ; STUCK TO 0
1472 IF #RCVRIE NOTSETIN @RCSR THEN
1473 BIT #RCVRIE,@RCSR
1474 BNE $24
1475 ; RCVRIE DID NOT SET IN RCSR
1476 ; ERRHRD 36,,DIDNOT
1477 ERROR 36
1478 ENDIF
1479 $24:
1480 ENDSUB
1481
1482 ; TRY TO CLEAR A SET BIT
1483 BGNSUB
1484 MOV #64,$SLPERR
1485
1486 LET @RCSR := @RCSR CLR.BY #RCVRIE
1487 BIC #RCVRIE,@RCSR
1488 ; SHOULD HAVE CLEARED
1489 IF #RCVRIE SETIN @RCSR THEN
1490 BIT #RCVRIE,@RCSR
1491 BEQ $25
1492 ; RCVRIE DID NOT CLEAR IN RCSR
1493 ; ERRHRD 37,,DIDNOT
1494 ERROR 37
1495 ENDIF
1496 $25:
1497 ENDSUB
1498

```



J03

MAINDEC-11-DVDVC-A  
DVDVCA.P11

MACY11 27(1006)  
08-AUG-77 09:16

08-AUG-77 09:20 PAGE 36  
T5 RCVRIE - RCSR6 SET, CLEAR, RESET

SEQ 0035

; NOW SEE IF RESET CLEARS IT  
BGNSUB

1499  
1500 003252  
1501 003252 012767 003260 175630  
1502  
1503 003260  
1504 003260 052777 000100 175772  
1505  
1506 003266  
1507 003266 000005  
1508 003270  
1509 003270 032777 000100 175762  
1510 003276 001401  
1511  
1512 003300  
1513 003300 104040  
1514 003302  
1515 003302  
1516 003302  
1517 003302  
1518 003302  
1519  
1520  
1521  
1522

MOV #64\$, \$LPERR

BIS #RCVRIE, @RCSR

RESET

BIT #RCVRIE, @RCSR  
BEQ \$26

ERROR 40

\$26:

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

; ISSUE BUS RESET  
BRES@T

IF #RCVRIE SET IN @RCSR THEN

; RCVRIE DID NOT RESET IN RCSR  
ERRHRD 40,, DIDNOT

ENDIF

CKLOOP

ENDSUB

ENDTST

1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558

\*\*\*\*\*  
\* THE FOLLOWING 4 TESTS VERIFY  
\* THAT RESET (INIT) INITIALIZES READ ONLY BITS.  
\*\*\*\*\*  
\*\*\*\*\*  
\*TEST 6 TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT  
\*\*\*\*\*

TST6: SCOPE  
MOV #10,\$TIMES ;;DO 10 ITERATIONS  
MOV #6,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX

003302 000004  
003304 012767 000010 175646  
003312 012767 000006 175660

003320  
003320 012767 003326 175562  
003326  
003326 032777 000200 175724  
003334 001402

BGNSUB  
MOV #64,\$LPERR  
IF #RCVRDONE SETIN @RCSR THEN  
BIT #RCVRDONE,@RCSR  
BEQ \$27

;RCVRDONE SHOULD HAVE CLEARED BY INIT  
;RCVRDONE DID NOT CLEAR IN RCSR  
ERRHRD 41,MRESET, DIDNOT

003336  
003336 104041  
003340  
003340 000005  
003342  
003342

ERROR 41  
RESET

;REISSUE RESET  
BRESET

\$27:

ENDIF  
;ALLOW LOOPING AFTER ERROR  
CKLOOP  
ENDSUB  
ENDTST



```

1559
1560
1561
1562
1563 003342 000004
1564 003344 012767 000010 175606
1565 003352 012767 000007 175620
1566
1567
1568
1569
1570 003360
1571 003360 026727 005703 000001
1572 003366 001001
1573
1574 003370
1575 003370 000416
1576 003372
1577 003372
1578 003372 032767 020000 175620
1579 003400 001401
1580
1581 003402
1582 003402 000411
1583 003404
1584 003404
1585 003404 032767 000004 175606
1586 003412 001401
1587
1588 003414
1589 003414 000404
1590 003416
1591 003416
1592 003416 012767 000001 175534
1593 003424 000414
1594 003426
1595 003426
1596 003426
1597 003426
1598 003426
1599 003426
1600
1601 003426
1602 003426 012767 003434 175454
1603
1604 003434
1605 003434 032777 004000 175616
1606 003442 001405
1607
1608
1609 003444
1610 003444 042777 000004 175612
1611 003452
1612 003452 104044
1613
1614

```

```

*****
*****
*TEST 7 TEST THAT RCVRACT - RCSR 11 - IS CLEARED BY INIT
*****
TST7: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #7,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX

IF CONSOLE EQ #TRUE THEN
CMP CONSOLE,#TRUE
BNE $30
ELSE ;; EXECUTE TEST
BR $31
$30:
IF #WRAP SETIN SUSWR THEN
BIT #WRAP,$USWR
BEQ $32
ELSE ;; EXECUTE TEST
BR $33
$32:
IF #MAINT SETIN $USWR THEN
BIT #MAINT,$USWR
BEQ $34
ELSE ;;EXECUTE TEST
BR $35
$34:
EXIT TEST ; LINE MUST BE TERMINATED
MOV #1,$TIMES
BR TST10
;;EXIT THIS TEST
ENDIF
$35:
ENDIF
$33:
ENDIF
$31:

BGNSUB
IF #RCVRACT SETIN @RCSR THEN
;RESET SHOULD HAVE CLEARED RCVRACT
LET @TCSR := @TCSR CLR.BY #MAINT
ERRHRD 44, HRESET, DIDNOT
;TESTING EFFECT OF RESET ON BIT

```

M03

MAINDEC-11-DVDVC-A  
DVDVCA.P11

MACY11 27(1006)  
08-AUG-77 09:16

08-AUG-77 09:20 PAGE 39  
T7

TEST THAT RCVRACT - RCSR 11 - IS CLEARED BY INIT

SEQ 0038

1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627

003454  
003454 000005  
003456  
003456  
003456  
003456  
003456

RESET  
\$36:

;RCVRACT DID NOT CLEAR IN RCSR  
:ALLOW ANOTHER TRY  
BRESET  
ENDIF  
:ALLOW LOOPING ON ERROR  
CKLOOP  
ENDSUB  
ENDTST



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

003456 000004  
003460 012767 000010 175472  
003466 012767 000010 175504  
  
003474  
003474 012767 003502 175406  
  
003502  
003502 032777 000200 175554  
003510 001002  
  
003512  
003512 104042  
  
003514  
003514 000005  
003516  
003516  
003516  
003516

```
*****  
*****  
*TEST 10 TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT  
*****  
TST10: SCOPE  
MOV #10,STIMES ;;DO 10 ITERATIONS  
MOV #10,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
```

```
BGNSUB  
MOV #645,SLPERR  
IF #XMITRDY NOTSETIN @TCSR THEN  
BIT #XMITRDY,@TCSR  
BNE $37  
  
:RESET SHOULD HAVE SET BIT.  
:XMITRDY DID NOT SET IN TCSR (AFTER RESE  
ERRHRD 42,HRESET,DIDNOT  
:ISSUE ANOTHER RESET  
BRESET  
ENDIF  
:ALLOW LOOPING ON ERROR  
CKLOOP  
ENDSUB  
ENDTST
```

\$37:

```

1662 ;*****
1663 ;*****
1664 ;*TEST 11 TEST THAT XMIT RDY - TCSR 7 - CLEARS
1665 ;* WHEN TBUF IS LOADED WITH A CHARACTER
1666 ;* AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
1667 ;*****
1668 003516 000004 TST11: SCOPE
1669 003520 012767 000010 175432 MOV #10,$TIMES ;;DO 10 ITERATIONS
1670 003526 012767 000011 175444 MOV #11,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1671
1672 003534 IFB APTCON EQ #TRUE THEN
1673 003534 126727 005526 000001 CMPB APTCON,#TRUE
1674 003542 001004 BNE $40
1675 003544 EXIT TEST
1676 003544 012767 000001 175406 MOV #1,$TIMES
1677 003552 000513 BR TST12 ;;EXIT THIS TEST
1678 003554 ENDIF
1679 003554 $40:
1680
1681 003554 LET PASS := #1 ;INIT COUNT OF TIMES THRU
1682 003554 012767 000001 000212 MOV #1,PASS
1683 003562 LOOP ; START OF LOOP
1684 003562 $41:
1685 ; MAX OF 2 TIMES THRU
1686 003562 LET ERRORFLAG := #CLR
1687 003562 012767 000000 000206 MOV #CLR,ERRORFLAG
1688 003570 LET EXITFLAG := #CLR
1689 003570 012767 000000 000202 MOV #CLR,EXITFLAG
1690 ; LOAD TBUF WITH ONE CHARACTER
1691 ; WAIT FOR READY TO SET
1692 ; (SHOULD BE VERY SHORT WAIT
1693 ; SINCE UART DOUBLE BUFFERS ITS INPUT)
1694
1695 ; SEND A CHARACTER
1696 003576 LET #TBUF :B= #0
1697 003576 105077 175466 CLRB #TBUF
1698 ; WAIT A MAXIMUM
1699 ; OF 50 MSEC FOR
1700 ; XMIT RDY TO SET IN TCSR
1701 CALL TIMER IN (<#5,#XMITRDY,TCSR,#SET)
1702 003602 MOV R5,-(SP)
1703 003604 010546 177777 MOV #SET,-(R5)
1704 003610 016745 175450 MOV TCSR,-(R5)
1705 003614 012745 000200 MOV #XMITRDY,-(R5)
1706 003620 012745 000005 MOV #5,-(R5)
1707 003624 004767 004632 JSR PC,TIMER
1708 003630 012605 MOV (SP)+,R5
1709
1710 ; TIMER RETURNS AN ERROR IF BIT DID
1711 ; NOT MEET CONDITION WITHIN TIME LIMIT
1712 003632 103001 BCC $43 IF.ERROR THEN
1713
1714 ; XMIT RDY DID NOT SET IN TCSR
1715 003634 104066 ERROR 66 ERRHRD 66,,DIDNOT
1716 003636 ENDIF
1717 003636 $43:

```



```

1718
1719
1720
1721
1722
1723
1724 003636
1725 003636 105077 175426
1726 003642 000240
1727
1728
1729 003644
1730 003644 032777 000200 175412
1731 003652 001404
1732
1733 003654
1734 003654 012767 177777 000114
1735
1736
1737 003662
1738 003662 000416
1739 003664
1740
1741
1742
1743 003664
1744 003664 010546
1745 003666 012745 177777
1746 003672 016745 175366
1747 003676 012745 000200
1748 003702 012745 000005
1749 003706 004767 004550
1750 003712 012605
1751 003714
1752 003714 103001
1753
1754 003716
1755 003716 104070
1756 003720
1757 003720
1758 003720
1759 003720
1760 003720
1761 003720 026727 000052 177777
1762 003726 001011
1763 003730
1764 003730 026727 000040 000001
1765 003736 003404
1766
1767 003740
1768 003740 104067
1769 003742
1770 003742 012767 177777 000030
1771 003750
1772 003750
1773 003750

```

```

; LOAD TBUF WITH A SECOND CHARACTER
; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
; AND THEN WAIT FOR IT TO SET

; SEND SECOND CHARACTER
LET @TBUF :B= #0

; GIVE IT TIME TO CLEAR
; XMITRDY SHOULD HAVE CLEARED UPON
; RECEIPT OF A CHARACTER
IF #XMITRDY SET IN @TCSR THEN

; XMITRDY DID NOT CLEAR IN TCSR
LET ERRORFLAG := #SET
; DEFER ERROR TYPEOUT

ELSE

; WAIT A MAXIMUM
; OF 50 MSEC FOR
; XMIT RDY TO SET IN TCSR
CALL TIMER IN (&#5, #XMITRDY, TCSR, #SET)

IF .ERROR THEN
; XMIT RDY DID NOT SET IN TCSR
ERRHRD 70,,DIDNOT
ENDIF
ENDIF ; OF DEFERED ERROR CALL
IF ERRORFLAG EQ #SET THEN
IF PASS GT #1 THEN
; CALL ERROR IF 2ND TRY
ERRHRD 67,,DIDNOT
LET EXITFLAG := #SET
ENDIF
ELSE ; NO ERROR

```

```

$44: BR $45
$46: BCC $46
$47: BNE $47
$48: MOV #SET, -(R5)
$49: MOV TCSR, -(R5)
$50: MOV #XMITRDY, -(R5)
$51: MOV #5, -(R5)
$52: JSR PC, TIMER
$53: MOV (SP)+, R5
$54: MOV R5, -(SP)
$55: MOV #SET, -(R5)
$56: MOV TCSR, -(R5)
$57: MOV #XMITRDY, -(R5)
$58: MOV #5, -(R5)
$59: JSR PC, TIMER
$60: MOV (SP)+, R5
$61: BCC $46
$62: ERROR 70
$63: BNE $47
$64: CMP ERRORFLAG, #SET
$65: BLE $50
$66: CMP PASS, #1
$67: ERROR 67
$68: MOV #SET, EXITFLAG
$69: BNE $46
$70: BCC $46
$71: BNE $47
$72: MOV #SET, EXITFLAG
$73: BNE $46

```

004

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 43  
DVDVCA.P11 08-AUG-77 09:16 T11 TEST THAT XMIT RDY - TCSR 7 - CLEARS

SEQ 0042

```
1774 003750 000403          BR      $51
1775 003752          $47:          LET EXITFLAG := #SET
1776 003752          MOV      #SET,EXITFLAG          ENDIF
1777 003752 012767 177777 000020          EXIF      EXITFLAG EQ #SET
1778 003760          $51:          ENDLOOP
1779 003760          CMP      EXITFLAG,#SET
1780 003760 026727 000014 177777          BEQ      $42
1781 003766 001401
1782 003770          BR      $41
1783 003770 000674          $42:
1784 003772          BR      TST12          ;;EXIT : SKIP AROUND FLAG WORDS
1785 003772 000403          ;;EXIT THIS TEST
1786 003774 000000          PASS:          0
1787 003776 000000          ERRORFLAG:    0
1788 004000 000000          EXITFLAG:      0
1791 004002          ENDTST
```



E04

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 44  
DVDVCA.P11 08-AUG-77 09:16 T11

TEST THAT XMIT RDY - TCSR 7 - CLEARS

SEQ 0043

```

1792
1793
1794
1795
1796
1797
1798 004002 000004
1799 004004 012767 000010 175146
1800 004012 012767 000012 175160
1801
1802 004020
1803 004020 032767 040000 175172
1804 004026 001404
1805 004030 126727 005232 000001
1806 004036 001004
1807 004040
1808 004040
1809 004040 012767 000001 175112
1810 004046 000442
1811 004050
1812 004050
1813
1814
1815 004050
1816 004050 052777 000004 175206
1817
1818 004056
1819 004056 012767 004064 175024
1820
1821
1822 004064
1823 004064 105077 175200
1824
1825
1826
1827
1828 004070
1829 004070 010546
1830 004072 012745 177777
1831 004076 016745 175156
1832 004102 012745 000200
1833 004106 012745 000005
1834 004112 004767 004344
1835 004116 012605
1836
1837
1838 004120
1839 004120 103004
1840
1841
1842 004122
1843 004122 042777 000004 175134
1844 004130
1845 004130 104071
1846 004132
1847 004132

```

```

*****
*****
TEST 12      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.
*****
TST12: SCOPE
        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
        MOV     #12,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                IF #MAINTJUMP NOTSETIN SUSWR ORB APTCON EQ #TRUE
        BIT     #MAINTJUMP,$USWR
        BEQ     $52
        CMPB   APTCON,#TRUE
        DNE    $53
$52:
                                EXIT TEST
        MOV     #1,$TIMES
        BR     TST13           ;;;EXIT THIS TEST
                                ENDIF
$53:
                                ; SET THE MAINTENANCE BIT
                                LET @TCSR := @TCSR SET.BY #MAINT
                                BGNSUB
        MOV     #64,$SLPERR
                                ; SEND A CHARACTER AND LET IT WRAP AROUND
                                LET @TBUF :B= #0
        CLRB   @TBUF
                                ; WAIT A MAXIMUM OF 50 MSEC
                                ; FOR RCVR DONE TO SET IN
                                ; RCSR
                                CALL TIMER IN (<#5,#RCVRDONE,RCSR,#SET)
        MOV     RS,-(SP)
        MOV     #SET,-(RS)
        MOV     RCSR,-(RS)
        MOV     #RCVRDONE,-(RS)
        MOV     #5,-(RS)
        JSR    PC,TIMER
        MOV     (SP)+,RS
                                ;DIDN'T SET IN TIME
                                IF.ERROR THEN
        BCC    $54
                                ; RCVRDONE DID NOT SET IN RCSR
                                ; CAN NOT LEAVE WITH MAINT SET
        LET   @TCSR := @TCSR CLR.BY #MAINT
                                ERRHRD 71,,DIDNOT
        ERROR  71
        ENDIF
$54:

```

F04

MAINDEC-11-DVDVC-A  
DVDVCA.P11

MACY11 27(1006)  
08-AUG-77 09:16

08-AUG-77 09:20 PAGE 45  
T12

TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)

SEQ 0044

```

1848
1849 004132                                ENDSUB
1850
1851 004132                                BGNSUB
1852 004132 012767 004140 174750          MOV    #64$,$LPERB
1853                                     ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
1854                                     ; THIS ALSO WILL CLEAR THE MAINT. BIT
1855                                     BRESET
1856 004140 000005                          RESET
1857
1858 004142                                IF #RCVRDONE SETIN #RCSR THEN
1859 004142 032777 000200 175110          BIT    #RCVRDONE,#RCSR
1860 004150 001401                          BEQ    $55
1861                                     ; RCVRDONE DID NOT RESET IN RCSR.
1862                                     ERRHRD 72,,DIDNOT
1863 004152 104072                          ERROR  72
1864
1865 $55:
1866 004154                                ENDSUB
1867 004154                                ENDTST

```



```

1868
1869
1870
1871
1872 004154 000004
1873 004156 012767 000010 174774
1874 004164 012767 000013 175006
1875
1876 004172
1877 004172 032767 040000 175020
1878 004200 001404
1879 004202 126727 005060 000001
1880 004210 001004
1881 004212
1882 004212
1883 004212 012767 000001 174740
1884 004220 000451
1885 004222
1886 004222
1887
1888
1889 004222
1890 004222 052777 000004 175034
1891 004230
1892 004230 012767 004236 174652
1893
1894
1895
1896
1897 004236
1898 004236 105077 175026
1899
1900
1901
1902 004242
1903 004242 010546
1904 004244 012745 177777
1905 004250 016745 175004
1906 004254 012745 000200
1907 004260 012745 000050
1908 004264 004767 004172
1909 004270 012605
1910
1911 004272
1912 004272 103007
1913
1914
1915 004274
1916 004274 042777 000004 174762
1917 004302
1918 004302 104073
1919
1920 004304
1921 004304 052777 000004 174752
1922 004312
1923 004312

```

```

*****
*****
*TEST 13 TEST THAT RCVRDONE IS CLEARED BY READING RBUF
*****
TST13: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #13,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
IF #MAINTJUMP NOTSETIN SUSWR ORB APTCON EQ #TRUE
BIT #MAINTJUMP,SUSWR
BEQ $S6
CMPB APTCON,#TRUE
BNE $S7
$S6:
EXIT TEST
MOV #1,$TIMES
BR TST14 ;;EXIT THIS TEST
$S7:
ENDIF
; SET MAINT. BIT
LET @TCSR := @TCSR SET.BY #MAINT
BGNSUB
MOV #64,$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
; CAN NOT LEAVE WITH MAINT SET
LET @TCSR := @TCSR CLR.BY #MAINT
ERRHRD 73,, DIDNOT
; SET IT BACK TO CONTINUE
LET @TCSR := @TCSR SET.BY #MAINT
ENDIF
$S6:

```

H04

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 47  
DVDVCA.P11 08-AUG-77 09:16

T13 TEST THAT RCVRDONE IS CLEARED BY READING RBUF

SEQ 0046

ENDSUB

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

```

1924 004312
1925
1926
1927
1928
1929
1930 004312
1931 004312 117700 174744      MOVB   @RBUF,R0
1932
1933 004316
1934 004316 032777 000200 174734  BIT    #RCVRDONE,@RCSR
1935 004324 001407                BEQ    $61
1936
1937
1938 004326
1939 004326 042777 000004 174730  BIC    #MAINT,@TCSR
1940 004334
1941 004334 104074                ERROR  74
1942
1943 004336
1944 004336 052777 000004 174720  BIS    #MAINT,@TCSR
1945 004344
1946 004344
1947 004344

```

: READ BUFFER  
LET R0 :B= @RBUF

IF #RCVRDONE SETIN @RCSR THEN

: RCVRDONE DID NOT CLEAR IN RCSR  
: CAN NOT LEAVE WITH MAINT SET  
LET @TCSR := @TCSR CLR.BY #MAINT

ERRHRD 74,,DIDNOT

: SET IT BACK TO CONTINUE  
LET @TCSR := @TCSR SET.BY #MAINT

ENDIF

\$61:

ENDTST

```

1948 .....*****
1949 .....*****
1950 .....*****
1951 .....*TEST 14 TEST THAT RCVRACT - RCSR 11 - SETS
1952 .....* WHEN A START BIT IS RECEIVED AND
1953 .....* CLEARS WHEN RCVRDONE - RCSR 7 - SETS
1954 .....*****
1954 004344 000004 TST14: SCOPE
1955 004346 012767 000010 174604 MOV #10,STIMES ;;DO 10 ITERATIONS
1956 004354 012767 000014 174616 MOV #14,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
1957 004362 126727 004700 000000 CMPB APTCON,#FALSE IFB APTCON EQ #FALSE AND #MAINTJUMP SETIN SUSWR
1958 004362 126727 004700 000000 BNE $62
1959 004370 001005 BIT #MAINTJUMP,SUSWR
1960 004372 032767 040000 174620 BEQ $62
1961 004400 001401
1962 .....
1963 004402 ELSE ;; EXECUTE TEST
1964 004402 000404 BR $63
1965 004404 $62:
1966 004404 EXIT TEST
1967 004404 012767 000001 174546 MOV #1,STIMES
1968 004412 000526 BR TST15 ;;EXIT THIS TEST
1969 004414 ENDIF
1970 004414 $63:
1971 004414 LET @TCSR := @TCSR SET.BY #MAINT
1972 004414 052777 000004 174642 BIS #MAINT,@TCSR
1973 004422 LET FLAG :B= #CLR
1974 004422 112767 000000 004202 MOVB #CLR,FLAG
1975 004430 LET COUNT := #0
1976 004430 005067 000232 CLR COUNT
1977 .....;LOAD A CHARACTER INTO TBUF
1978 .....;WAIT FOR RCVRACT TO SET
1979 .....
1980 .....:SEND A CHARACTER
1981 004434 LET @TBUF :B= #0
1982 004434 105077 174630 CLRB @TBUF
1983 004440 REPEAT
1984 004440 $64:
1985 004440 IF #RCVRACT SETIN @RCSR THEN
1986 004440 032777 004000 174612 BIT #RCVRACT,@RCSR
1987 004446 001404 BEQ $65
1988 004450 LET FLAG :B= #SET
1989 004450 112767 177777 004154 MOVB #SET,FLAG
1990 004456 ELSE
1991 004456 000402 BR $66
1992 004460 $65:
1993 004460 LET COUNT := COUNT + #1
1994 004460 005267 000202 INC COUNT
1995 004464 ENDIF
1996 004464 $66:
1997 004464 UNTILB FLAG EQ #SET OR COUNT HI MAX
1998 004464 126727 004142 177777 CMPB FLAG,#SET
1999 004472 001404 BEQ $67
2000 004474 026767 000166 000162 CMP COUNT,MAX
2001 004502 101756 BLOS $64
2002 004504 $67:
2003 004504 IF COUNT HI MAX THEN

```





K04

```

2060 004616                                LET  @TCSR := @TCSR CLR.BY #MAINT
2061 004616 042777 000004 174440          BIC  #MAINT,@TCSR
2062 004624                                ERRHRD 77,,DIDNOT
2063 004624 104077                          ERROR 77
2064 004626                                EXIT TEST
2065 004626 012767 000001 174324          MOV  #1,@TIMES
2066 004634 000415                          BR   TST15          ;;EXIT THIS TEST
2067 004636                                ENDIF
2068 004636                                $75:
2069                                     ;TEST THAT READING THE RECEIVER
2070                                     ;BUFFER CLEARS RCVRDONE
2071
2072
2073                                     ;READ CHAR.
2074 004636                                LET  @R0 := @RBUF
2075 004636 017700 174420                    MOV  @RBUF,@R0
2076
2077                                     IF  @RCVRDONE SETIN @RCSR THEN
2078 004642 032777 000200 174410            BIT  #RCVRDONE,@RCSR
2079 004650 001404                          BEQ  $76
2080                                     ;RCVRDONE DID NOT CLEAR IN RCSR
2081                                     ;CAN NOT LEAVE WITH MAINT SET
2082 004652                                LET  @TCSR := @TCSR CLR.BY #MAINT
2083 004652 042777 000004 174404          BIC  #MAINT,@TCSR
2084 004660                                ERRHRD 100,,DIDNOT
2085 004660 104100                          ERROR 100
2086 004662                                ENDIF
2087 004662                                $76:
2088
2089                                     EXIT
2090 004662 000402                          BR   TST15          ;;EXIT THIS TEST
2091 004664 070000                          MAX:70000
2092 004666 000000                          COUNT: 0
2093
2094 004670                                ENDTST
2095

```

```

2096
2097
2098
2099
2100
2101 004670 000004
2102 004672 012767 000010 174260
2103 004700 012767 000015 174272
2104
2105 004706
2106 004706 032767 100000 174304
2107 004714 001416
2108 004716 126727 004344 000000
2109 004724 001012
2110 004726
2111 004726 032767 040000 174264
2112 004734 001401
2113
2114 004736
2115 004736 000404
2116 004740 $100:
2117 004740
2118 004740 012767 000001 174212
2119 004746 000567
2120 004750
2121 004750 $101:
2122 004750
2123 004750 000404
2124 004752 $77:
2125 004752
2126 004752 012767 000001 174200
2127 004760 000562
2128 004762
2129 004762 $102:
2130
2131 004762
2132 004762 052777 000004 174274
2133
2134
2135
2136 004770
2137 004770 012767 004776 174112
2138
2139
2140
2141
2142
2143 004776
2144 004776 105077 174266
2145
2146 005002
2147 005002 010546
2148 005004 012745 000050
2149 005010 004767 003724
2150 005014 012605
2151

```

```

*****
*****
*TEST 15 TEST THE OVERRUN BIT - RBUF 14
*****
TST15: SCOPE
MOV #10,STIMES ;;DO 10 ITERATIONS
MOV #15,STESTN ;;SET TEST NUMBER IN APT MAIL BOX

IF #ERRBITS SETIN SUSWR ANDB APTCON EQ #FALSE TH
BIT #ERRBITS,SUSWR
BEQ $77
CMPB APTCON,#FALSE
BNE $77
IF #MAINTJUMP SETIN SUSWR THEN
BIT #MAINTJUMP,SUSWR
BEQ $100
ELSE ;; NULL ---EXECUTE TEST
BR $101
$100: BR $101
EXIT TEST
MOV #1,STIMES
BR TST16 ;;EXIT THIS TEST
ENDIF
ELSE
BR $102
$77: BR $102
EXIT TEST
MOV #1,STIMES
BR TST16 ;;EXIT THIS TEST
ENDIF
$102:
LET @TCSR := @TCSR SET.BY #MAINT
BIS #MAINT,@TCSR
BGNSUB
MOV #64$,$LPERR
;OUTPUT 2 CHARACTERS WITH
;AMPLE DELAYS BETWEEN FOR RECEPTION.
;THIS SHOULD AN CAUSE OVERRUN ERROR.
;OUTPUT 1 CHARACTER
LET @TBUF :B= #0
;GO AWAY FOR 500 M SEC
WAITUS 50
MOV R5,-(SP)
MOV #50,-(R5)
JSR PC,WAIT
MOV (SP)+,R5

```



M04

```

2152                                     ;OUTPUT 2ND CHARACTER
2153 005016                               LET @TBUF :B= #0
2154 005016 105077 174246                 CLRB @TBUF
2155                                     ;LET OVERRUN HAPPEN
2156 005022                               WAITUS 50
2157 005022 010546                         MOV R5, -(SP)
2158 005024 012745 000050                 MOV #50, -(R5)
2159 005030 004767 003704                 JSR PC, WAIT
2160 005034 012605                         MOV (SP)+, R5
2161
2162                                     ;READ BUFFER AND ERROR BITS
2163 005036                               LET R4 := @RBUF
2164 005036 017704 174220                 MOV @RBUF, R4
2165
2166                                     ;IT DIDN'T SET
2167 005042                               IF #ORERR NOTSETIN R4 THEN
2168 005042 032704 040000                 BIT #ORERR, R4
2169 005046 001010                         BNE $103
2170
2171                                     ;ORERR DID NOT SET IN RBUF
2172                                     ; CAN NOT LEAVE WITH MAINT SET
2173 005050                               LET @TCSR := @TCSR CLR.BY #MAINT
2174 005056 042777 000004 174206         BIC #MAINT, @TCSR
2175 005056 104101                         ERRHRD 101, ,DIDNOT
2176
2177                                     ;NO USE COMPOUNDING ERRORS
2178 005060                               EXIT TST
2179 005060 012767 000001 174072         MOV #1, $TIMES
2180 005066 000517                         BR TST16          ;;EXIT THIS TEST
2181 005070                               ENDIF
2182 005070                               $103:
2183 005070                               ENDSUB
2184
2185                                     ;NOW SEE IF ERROR BIT SET WITH OVERRUN ERROR:
2186 005070                               BGNSUB
2187 005070 012767 005076 174012         MOV #64$, $LPERR
2188 005076                               IF #ERROR NOTSETIN R4 THEN
2189 005076 032704 100000                 BIT #ERROR, R4
2190 005102 001010                         BNE $104
2191
2192                                     ;ERROR DID NOT SET IN RBUF
2193                                     ; CAN NOT LEAVE WITH MAINT SET
2194 005104                               LET @TCSR := @TCSR CLR.BY #MAINT
2195 005104 042777 000004 174152         BIC #MAINT, @TCSR
2196 005112                               ERRHRD 102, ,DIDNOT
2197 005112 104102                         ERROR 102
2198
2199                                     ;-WHEN ORERR SET.
2200                                     ;GET OUT NOW.
2201 005114                               EXIT TST
2202 005114 012767 000001 174036         MOV #1, $TIMES
2203 005122 000501                         BR TST16          ;;EXIT THIS TEST
2204 005124                               ENDIF
2205 005124                               $104:
2206 005124                               ENDSUB
2207
  
```

```

2208 005124                                BGNSUB
2209 005124 012767 005132 173756        MOV    #64S,SLPERR
2210                                ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
2211
2212 005132                                IF #ORERR NOTSETIN @RBUF THEN
2213 005132 032777 040000 174122        BIT    #ORERR,@RBUF
2214 005140 001010                        BNE    $105
2215
2216                                ;READING RBUF CLEARED ORERR.
2217                                ; CAN NOT LEAVE WITH MAINT SET
2218                                LET    @TCSR := @TCSR CLR.BY #MAINT
2219 005142 042777 000004 174114        BIC    #MAINT,@TCSR
2220 005150                                ERRHRD 103,ITCLRED
2221 005150 104103                        ERROR  103
2222                                ;SKIP REST OF TEST
2223                                EXIT TEST
2224 005152 012767 000001 174000        MOV    #1,STIMES
2225 005160 000462                        BR     TST16                ;;;EXIT THIS TEST
2226                                ENDF
2227                                $105:
2228                                ENDSUB
2229                                BGNSUB
2230 005162                                MOV    #64S,SLPERR
2231 005162 012767 005170 173720        ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
2232
2233                                ;SEND A CHARACTER AROUND.
2234                                LET    @TBUF :B= #0
2235 005170 105077 174074                CLR    @TBUF
2236                                ;LET IT CIRCULATE
2237                                WAITUS 50
2238 005174 010546                        MOV    R5, -(SP)
2239 005176 012745 000050                MOV    #50, -(R5)
2240 005202 004767 003532                JSR    PC, WAIT
2241 005206 012605                        MOV    (SP)+, R5
2242
2243                                IF #ORERR SETIN @RBUF THEN
2244 005210 032777 040000 174044        BIT    #ORERR,@RBUF
2245 005216 001410                        BEQ    $106
2246
2247                                ;ORERR DID NOT CLEAR IN RBUF
2248                                ; CAN NOT LEAVE WITH MAINT SET
2249                                LET    @TCSR := @TCSR CLR.BY #MAINT
2250 005220 042777 000004 174036        BIC    #MAINT,@TCSR
2251 005226 104104                        ERRHRD 104,,DIDNOT
2252
2253                                ;-AFTER RECEIVING ANOTHER CHAR
2254                                ;SKIP AROUND REST
2255                                EXIT TST
2256 005230 012767 000001 173722        MOV    #1,STIMES
2257 005236 000433                        BR     TST16                ;;;EXIT THIS TEST
2258                                ENDF
2259                                $106:
2260                                IF #ERROR SETIN @RBUF THEN
2261 005240 032777 100000 174014        BIT    #ERROR,@RBUF
    
```

2264 005246 001404  
 2265  
 2266  
 2267 005250  
 2268 005250 042777 000004 174006  
 2269 005256  
 2270 005256 104105  
 2271  
 2272 005260  
 2273 005260  
 2274 005260  
 2275 005260  
 2276 005260 000422  
 2277 005262 042522 042101 047111  
 2278 005270 020107 041122 043125  
 2279 005276 041440 042514 051101  
 2280 005304 042105 047440 042526  
 2281 005312 051122 047125 042440  
 2282 005320 051122 051117 000056  
 2283  
 2284 005326  
 2285

BEQ \$107  
 BIC #MAINT,@TCSR  
 ERROR 105  
 \$107:  
 BR TST16

;ERROR DID NOT CLEAR IN RBUF  
 ; CAN NOT LEAVE WITH MAINT SET  
 LET @TCSR := @TCSR CLR.BY #MAINT  
 ERRHRD 105,,DIDNOT

ENDIF

ENDSUB  
 EXIT  
 ;;EXIT THIS TEST  
 ITCLRED: .ASCIZ /READING RBUF CLEARED OV

.EVEN  
 ENDTST



2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294 005326 000004  
2295 005330 012767 000010 173622  
2296 005336 012767 000016 173634  
2297  
2298  
2299  
2300  
2301  
2302  
2303 005344  
2304 005344 032767 000200 173646  
2305 005352 001416  
2306 005354 032767 040000 173636  
2307 005362 001412  
2308 005364  
2309 005364 126727 003676 000000  
2310 005372 001001  
2311  
2312 005374  
2313 005374 000404  
2314 005376 \$111:  
2315 005376  
2316 005376 012767 000001 173554  
2317 005404 000565  
2318 005406  
2319 005406 \$112:  
2320 005406  
2321 005406 000404  
2322 005410 \$110:  
2323 005410  
2324 005410 012767 000001 173542  
2325 005416 000560  
2326 005420  
2327 005420 \$113:  
2328  
2329 005420  
2330 005420 005067 002710  
2331 005424  
2332 005424 012767 177777 000322  
2333 005432  
2334 005432 012767 177777 000316  
2335 005440  
2336 005440 052777 000004 173616  
2337  
2338 005446  
2339 005446 005003  
2340 005450 000401  
2341 005452 \$115:

```

*****
*****
*TEST 16      PROGRAMMABLE BAUD RATE TEST
*             TEST AT ALL SPEEDS AVAILABLE
*             A COMPARISON WILL BE MADE TO SEE
*             IF NEW TIME IS LESS THAN PREVIOUS.
*****

```

```

TST16: SCOPE
MOV      #10,$TIMES      ;;DO 10 ITERATIONS
MOV      #16,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

;;NOTE: IF PROGRAMMABLE BAUDE RATE TEST IS TO BE RUN ON THE APT CONSOLE
;;       THE OPERATOR MUST MAKE CERTAIN THAT LOCATION SUSWR IS
;;       SET WITH #16 IN BITS <11:8>(9600 BAUD).

```

IF #PBR SETIN SUSWR AND #MAINTJUMP SETIN SUSWR T

```

BIT      #PBR,SUSWR
BEQ      $110
BIT      #MAINTJUMP,SUSWR
BEQ      $110
IFB APTCON EQ #FALSE THEN
CMPB    APTCON,#FALSE
BNE     $111
ELSE;;; NULL --- EXECUTE TEST
$111: BR      $112
EXIT TST
MOV      #1,$TIMES
BR      TST17      ;;EXIT THIS TEST
ENDIF
$112: ELSE ; CATCH IT HERE
BR      $113
$110: EXIT TEST ; IF EITHER OF CASES FAIL
MOV      #1,$TIMES
BR      TST17      ;;EXIT THIS TEST
ENDIF
$113: LET ERRCHK := #0 ; CLEAR ERROR WORD
CLR      ERRCHK
LET OLD := #-1
MOV      #-1,OLD
LET OLD+2 := #-1
MOV      #-1,OLD+2
LET @TCSR := @TCSR SET.BY #MAINT
;EACH BAUD RATE
INCR R3 FROM #0 TO #15. BY #1
CLR      R3
BR      $114
$115:

```

|      |        |        |        |        |        |                   |                                       |
|------|--------|--------|--------|--------|--------|-------------------|---------------------------------------|
| 2342 | 005452 | 005203 |        |        | INC    | R3                |                                       |
| 2343 | 005454 |        |        |        | \$114: |                   |                                       |
| 2344 | 005454 | 020327 | 000017 |        | CMP    | R3,#15.           |                                       |
| 2345 | 005460 | 003062 |        |        | BGT    | \$116             |                                       |
| 2346 | 005462 |        |        |        |        |                   | LET R0 := @RBUF                       |
| 2347 | 005462 | 017700 | 173574 |        | MOV    | @RBUF,R0          |                                       |
| 2348 |        |        |        |        |        |                   | ;CHANGE BAUDE RATE                    |
| 2349 | 005466 |        |        |        | MOVB   | RATES(R3),@TCSRHI | LET @TCSRHI := RATES(R3)              |
| 2350 | 005466 | 116377 | 005676 | 173572 |        |                   | ;FLAG                                 |
| 2351 |        |        |        |        |        |                   | LET BIT := #0                         |
| 2352 | 005474 |        |        |        | CLR    | BIT               | ;OUTPUT THE CHARACTER                 |
| 2353 | 005474 | 005002 |        |        |        |                   | LET @TBUF := #0                       |
| 2354 |        |        |        |        |        |                   | ;INITIALIZE COUNTER                   |
| 2355 | 005476 |        |        |        | CLR    | @TBUF             | LET NEW := #0                         |
| 2356 | 005476 | 005077 | 173566 |        |        |                   | LET NEW+2 := #0                       |
| 2357 |        |        |        |        |        |                   | WHILE BIT EQ #0 DO                    |
| 2358 | 005502 |        |        |        | CLR    | NEW               |                                       |
| 2359 | 005502 | 005067 | 000242 |        |        |                   |                                       |
| 2360 | 005506 |        |        |        | CLR    | NEW+2             |                                       |
| 2361 | 005506 | 005067 | 000240 |        |        |                   |                                       |
| 2362 | 005512 |        |        |        |        |                   |                                       |
| 2363 | 005512 |        |        |        | \$117: |                   |                                       |
| 2364 | 005512 | 005702 |        |        | TST    | BIT               |                                       |
| 2365 | 005514 | 001014 |        |        | BNE    | \$120             |                                       |
| 2366 | 005516 |        |        |        |        |                   | IF #RCVRDONE SETIN @RCSR THEN         |
| 2367 | 005516 | 032777 | 000200 | 173534 | BIT    | #RCVRDONE,@RCSR   |                                       |
| 2368 | 005524 | 001403 |        |        | BEQ    | \$121             |                                       |
| 2369 |        |        |        |        |        |                   | ;DONE - ITS READY                     |
| 2370 | 005526 |        |        |        |        |                   | LET BIT := #1                         |
| 2371 | 005526 | 012702 | 000001 |        | MOV    | #1,BIT            |                                       |
| 2372 | 005532 |        |        |        |        |                   | ELSE                                  |
| 2373 | 005532 | 000404 |        |        | BR     | \$122             |                                       |
| 2374 | 005534 |        |        |        | \$121: |                   |                                       |
| 2375 |        |        |        |        |        |                   | ;OTHERWISE-INCREMENT TIME             |
| 2376 | 005534 |        |        |        |        |                   | LET NEW := NEW + #1                   |
| 2377 | 005534 | 005267 | 000210 |        | INC    | NEW               |                                       |
| 2378 | 005540 |        |        |        |        |                   | LET NEW+2 := NEW+2 + CARRY            |
| 2379 | 005540 | 005567 | 000206 |        | ADC    | NEW+2             |                                       |
| 2380 | 005544 |        |        |        |        |                   | ENDIF                                 |
| 2381 | 005544 |        |        |        | \$122: |                   |                                       |
| 2382 |        |        |        |        |        |                   | ;SIGNALS DONE                         |
| 2383 | 005544 |        |        |        |        |                   | ENDDO                                 |
| 2384 | 005544 | 000762 |        |        | BR     | \$117             |                                       |
| 2385 | 005546 |        |        |        | \$120: |                   |                                       |
| 2386 |        |        |        |        |        |                   | IF NEW+2 LO OLD+2 THEN                |
| 2387 | 005546 |        |        |        |        |                   |                                       |
| 2388 | 005546 | 026767 | 000200 | 000202 | CMP    | NEW+2,OLD+2       |                                       |
| 2389 | 005554 | 103001 |        |        | BHIS   | \$123             |                                       |
| 2390 |        |        |        |        |        |                   | ; OK                                  |
| 2391 | 005556 |        |        |        |        |                   | ELSE                                  |
| 2392 | 005556 | 000414 |        |        | BR     | \$124             |                                       |
| 2393 | 005560 |        |        |        | \$123: |                   |                                       |
| 2394 |        |        |        |        |        |                   | ; NEW+2 >= OLD+2                      |
| 2395 | 005560 |        |        |        |        |                   | IF NEW+2 EQ OLD+2 AND NEW LO OLD THEN |
| 2396 | 005560 | 026767 | 000166 | 000170 | CMP    | NEW+2,OLD+2       |                                       |
| 2397 | 005566 | 001005 |        |        | BNE    | \$125             |                                       |

```

2398 005570 026767 000154 000156      CMP      NEW,OLD
2399 005576 103001                      BHIS     $125
2400                                     ;OK
2401 005600                                ELSE
2402 005600 000403                      BR       $126
2403 005602                                $125:
2404                                     ;NEW+2 > OLD+2 OR
2405                                     ;(NEW+2 = OLD+2 AND
2406                                     ;NEW >= OLD)
2407                                     ;BAUD RATE DIDN'T CHANGE
2408 005602                                LET ERRCHK := #BIT2 ; SET ERROR INDICATOR
2409 005602 012767 000004 002524      MOV      #BIT2,ERRCHK
2410 005610                                ENDIF
2411 005610                                $126:
2412 005610                                ENDIF
2413 005610                                $124:
2414                                     ;UPDATE OLD TIME
2415 005610                                LET OLD := NEW
2416 005610 016767 000134 000136      MOV      NEW,OLD
2417 005616 016767 000130 000132      MOV      NEW+2,OLD+2
2418 005616                                LET OLD+2 := NEW+2
2419
2420                                ENDINC ;BAUD RATE
2421 005624 000712                      BR       $115
2422 005626                                $116:
2423 005626                                LET R3 :B= $USWR+1 AND #17 ; PUT BAUD BACK
2424 005626 116703 173367      MOV      $USWR+1,R3
2425 005632 110346      MOV      R3,-(SP)
2426 005634 142716 000017      BICB    #17,(SP)
2427 005640 142603      BICB    (SP)+,R3
2428 005642                                LET R3 := R3 CLR.BY #177400
2429 005642 042703 177400      BIC     #177400,R3
2430 005646                                LET @TCSRHI :B= RATES(R3) ; LIKE HE WANTED IT
2431 005646 116377 005676 173412      MOV      RATES(R3),@TCSRHI
2432
2433                                     ; CAN NOT LEAVE WITH MAINT SET
2434 005654                                LET @TCSR := @TCSR CLR.BY #MAINT
2435 005654 042777 000004 173402      BIC     #MAINT,@TCSR
2436 005662                                IF #BIT2 SETIN ERRCHK THEN
2437 005662 032767 000004 002444      BIT     #BIT2,ERRCHK
2438 005670 001401      BEQ     $127
2439
2440                                     ; REPORT DEFERED ERROR
2441 005672                                ERRHRD 126
2442 005672 104126      ERROR 126
2443 005674                                ENDIF
2444 005674                                $127:
2445 005674                                EXIT ;SKIP TABLE
2446 005674 000431      BR       TST17 ;;;EXIT THIS TEST
2447 005676

```

```

RATES: ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
;UPPER BYTE OF XCSR FOR EACH BAUD RATE
; ** NOTE: THE VALUE INDICATED IN THE COLUMN 'OFFSET
; ** INTO TABLE' CAN BE PLACED INTO BITS<11:8>
; ** OF LOCATION 'SUSWR' TO CAUSE THE CORRESPONDING
; ** BAUD TO BE SELECTED IN THE DLV11-F UPON
; ** COMPLETION OF THIS TEST.

```

```

2448
2449
2450
2451
2452
2453

```



|        |        |        |        |           |       | BAUD   | OFFSET INTO TABLE          |
|--------|--------|--------|--------|-----------|-------|--------|----------------------------|
| 005676 | 010    |        |        | R0050:    | .BYTE | 010    | 0                          |
| 005677 | 030    |        |        | R0070:    | .BYTE | 030    | 1                          |
| 005700 | 050    |        |        | R0110:    | .BYTE | 050    | 2                          |
| 005701 | 070    |        |        | R0135:    | .BYTE | 070    | 3                          |
| 005702 | 110    |        |        | R0150:    | .BYTE | 110    | 4                          |
| 005703 | 130    |        |        | R0300:    | .BYTE | 130    | 5                          |
| 005704 | 150    |        |        | R0600:    | .BYTE | 150    | 6                          |
| 005705 | 170    |        |        | R0200:    | .BYTE | 170    | 7                          |
| 005706 | 210    |        |        | R1800:    | .BYTE | 1800   | 10                         |
| 005707 | 230    |        |        | R2000:    | .BYTE | 2000   | 11                         |
| 005710 | 250    |        |        | R2400:    | .BYTE | 2400   | 12                         |
| 005711 | 270    |        |        | R3600:    | .BYTE | 3600   | 13                         |
| 005712 | 310    |        |        | R4800:    | .BYTE | 4800   | 14                         |
| 005713 | 330    |        |        | R7200:    | .BYTE | 7200   | 15                         |
| 005714 | 350    |        |        | R9600:    | .BYTE | 9600   | 16                         |
| 005715 | 370    |        |        | R10000:   | .BYTE | 19200  | 17                         |
|        |        |        |        |           |       |        |                            |
| 005716 | 040502 | 042125 | 051040 | BAUDRATE: |       | .ASCIZ | /BAUD RATE DIDN'T CHANGE./ |
| 005724 | 052101 | 020105 | 044504 |           |       |        |                            |
| 005732 | 047104 | 052047 | 041440 |           |       |        |                            |
| 005740 | 040510 | 043516 | 027105 |           |       |        |                            |
| 005746 | 000    |        |        |           |       |        |                            |
|        | 005750 |        |        | .EVEN     |       |        |                            |
| 005750 | 000000 | 000000 |        | NEW:      | 0,0   |        |                            |
| 005754 | 000000 | 000000 |        | OLD:      | 0,0   |        |                            |
| 005760 |        |        |        |           |       |        |                            |
|        |        |        |        |           |       | ENDTST |                            |

05  
06  
07  
08  
09  
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  
37  
38  
39  
40

```
*****
*****
*TEST 17 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
*****
```

```
TST17: SCOPE
MOV #10,$TIMES ;:DO 10 ITERATIONS
MOV #17,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
;CLEAR 'INTERRUPT OCCURED' FLAG
LET INTFLAG := #0
;GET VECTOR ADDRESS
LET R3 := DLVEC
;FOR THE TRANSMITTER
LET R3 := R3 + #4
;SET VECTOR TO POINT TO TRANS.SRV AT PRI
SETVEC R3, #INTSRV, #PR7
MOV R1, -(SP)
MOV R3, R1
MOV #INTSRV (R1)+
MOV #PR7, (R1)
MOV (SP)+, R1
BGNSUB
;: MAKE SURE THAT TRANSMITTER READY IS SET
CALL TIMER IN <#50, #XMITRDY, TCSR, #SET>
;CLEAR INTERRUPT ENABLE
LET @TCSR := @TCSR CLR.BY #XMITIE
;SET IT TO 0
;:PUT NEW PS ON STACK
;:PUT NEW PC ON STACK
;:POP NEW PC AND PS
b55:
;NOW SET I.E. BIT
LET @TCSR := @TCSR SET.BY #XMITIE
;LET INTERRUPT HAVE TIME TO OCCUR
```

|        |        |        |        |
|--------|--------|--------|--------|
| 005760 | 000004 |        |        |
| 005762 | 012767 | 000010 | 173170 |
| 005770 | 012767 | 000017 | 173202 |
| 005776 |        |        |        |
| 005776 | 005067 | 003024 |        |
| 006002 |        |        |        |
| 006002 | 016703 | 173250 |        |
| 006006 |        |        |        |
| 006006 | 062703 | 000004 |        |
| 006012 |        |        |        |
| 006012 | 010146 |        |        |
| 006014 | 010301 |        |        |
| 006016 | 012721 | 011020 |        |
| 006022 | 012711 | 000340 |        |
| 006026 | 012601 |        |        |
| 006030 |        |        |        |
| 006030 | 012767 | 006036 | 173052 |
| 006036 |        |        |        |
| 006036 | 010546 |        |        |
| 006040 | 012745 | 177777 |        |
| 006044 | 016745 | 173214 |        |
| 006050 | 012745 | 000200 |        |
| 006054 | 012745 | 000050 |        |
| 006060 | 004767 | 002376 |        |
| 006064 | 012605 |        |        |
| 006066 |        |        |        |
| 006066 | 042777 | 000100 | 173170 |
| 006074 | 012746 | 000000 |        |
| 006100 | 012746 | 006106 |        |
| 006104 | 000002 |        |        |
| 006106 |        |        |        |
| 006106 |        |        |        |
| 006106 | 052777 | 000100 | 173150 |
| 006106 |        |        |        |

# H05

MAINDEC-11-DVDVC-A    MACY11 27(1006)    08-AUG-77 09:20 PAGE 60  
 DVDVCA.P11    08-AUG-77 09:16    T17    TRANSMITTER INTERRUPT LOGIC TEST

SEQ 0059

```

2541 006114                               WAITUS 10
2542 006114 010546                       MOV    R5,-(SP)
2543 006116 012745 000010                 MOV    #10,-(R5)
2544 006122 004767 002612                 JSR    PC,WAIT
2545 006126 012605                       MOV    (SP)+,R5

2546
2547
2548                               ;DID EXACTLY 1 INTERRUPT OCCUR
2549 006130                               IF INTFLAG NE #1 THEN
2550 006130 026727 002672 000001           CMP    INTFLAG,#1
2551 006136 001406                       BEQ    $13C
2552
2553                               ;NO - WAS IT 0 OR MORE THAN ONCE
2554 006140                               IF INTFLAG EQ #0 THEN
2555 006140 005767 002662                 TST    INTFLAG
2556 006144 001002                       BNE    $131
2557
2558                               ;TRANSMITTER DID NOT INTERRUPT IN TIME
2559 006146                               ERRHRD 106,,DIDNOT
2560 006146 104106                       ERROR  106
2561 006150                               ELSE
2562 006150 000401                       BR     $132
2563 006152                               ;TWICE
2564 006152 104107                       ERROR  107
2565 006154                               ;TRANSMITTER INTERRUPTED TWICE
2566 006154                               ERRHRD 107,,TWICE
2567 006154                               ENDIF
2568 006154                               ENDIF
2569 006154                               ENDIF
2570                               ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
2571 006154                               BGNSUB
2572 006154 012767 006162 172726           MOV    #64$,$LPERR
2573                               ;CLEAR 'INTERRUPT OCCURED' FLAG
2574 006162                               LET INTFLAG := #0
2575 006162 005067 002640                 CLR    INTFLAG
2576                               ;CLEAR INTERRUPT ENABLE
2577 006166                               LET @TCSR := @TCSR CLR.BY #XMITIE
2578 006166 042777 000100 173070         BIC    #XMITIE,@TCSR
2579                               ;NO INTERRUPTS SHOULD OCCUR.
2580 006174 012746 000000                 MOV    #PRO,-(SP)
2581 006200 012746 006206                 MOV    #65$,-(SP)
2582 006204 000002                       RTI
2583 006206                               ;:PUT NEW PS ON STACK
2584                               ;:PUT NEW PC ON STACK
2585                               ;:POP NEW PC AND PS
2586                               ;DARE IT TO HAPPEN
2587 006206                               WAITUS 10
2588 006206 010546                       MOV    R5,-(SP)
2589 006210 012745 000010                 MOV    #10,-(R5)
2590 006214 004767 002520                 JSR    PC,WAIT
2591 006220 012605                       MOV    (SP)+,R5
2592 006222                               IF INTFLAG NE #0 THEN
2593 006222 005767 002600                 TST    INTFLAG
2594 006226 001401                       BEQ    $133
2595
2596                               ;INTERRUPT OCCURED WITH I E CLEARED
2597 006230                               ERRHRD 110,NOTENAB
2598 006230 104110                       ERROR  110
2599 006232                               ENDIF
  
```



```

2597 006232          $133:
2598 006232
2599 006232 000005          RESET
2600 006234
2601
2602 006234
2603 006234 010146          MOV     R1,-(SP)
2604 006236 010246          MOV     R2,-(SP)
2605 006240 012701 000003          MOV     #R3,R1
2606 006244 010102          MOV     R1,R2
2607 006246 062702 000002          ADD     #2,R2
2608 006252 010221          MOV     R2,(R1)+
2609 006254 005011          CLR     (R1)
2610 006256 012602          MOV     (SP)+,R2
2611 006260 012601          MOV     (SP)+,R1
2612
2613 006262
2614
2615
2616
2617
2618
2619

```

BRESET  
 ENDSUB  
 CLRVEC R3 :RESTORE VECTOR AREA  
 ;;PUSH R1 ON STACK  
 ;;PUSH R2 ON STACK  
 ;;POP STACK INTO R2  
 ;;POP STACK INTO R1  
 ENDTST

```

2620 ;*****
2621 ;*****
2622 ;*****
2623 ;TEST 20 RECEIVER INTERRUPT LOGIC TEST
2624 ;* THIS TEST COVERS ALL OF THE RECEIVER
2625 ;* SIDE OF THE INTERRUPT LOGIC IN
2626 ;* CHARACTER MODE.
2627 ;*****
2627 006262 000004 TST20: SCOPE
2628 006264 012767 000010 172666 MOV #10,$TIMES ;DO 10 ITERATIONS
2629 006272 012767 000020 172700 MOV #20,$TESTN ;SET TEST NUMBER IN APT MAIL BOX
2630 006300 ; IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
2631 006300 032767 040000 172712 BIT #MAINTJUMP,$USWR
2632 006306 001404 BEQ $134
2633 006310 126727 002752 000001 CMPB APTCON,#TRUE
2634 006316 001002 BNE $135
2635 006320 $134:
2636 006320 000167 000422 JMP TST21 ; EXIT TEST
2637 006324 ; ENDIF
2638 006324 $135:
2639 ;
2640 ; CLEAR INTERRUPT OCCURED FLAG
2641 ; SET UP RECEIVER INTER.VECTOR
2642 006324 SETVEC DLVEC,#INTSRV,#PR7
2643 006324 010146 MOV R1,-(SP)
2644 006326 016701 172724 MOV DLVEC,R1
2645 006332 012721 011020 MOV #INTSRV,(R1)+
2646 006336 012711 000340 MOV #PR7,(R1)
2647 006342 012601 MOV (SP)+,R1
2648 ; PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
2649 006344 ; BGNSUB
2650 006344 012767 006352 172536 MOV #64,$LPERR
2651 006352 LET INTFLAG := #0
2652 006352 005067 002450 CLR INTFLAG
2653 ; SET MAINT. BIT
2654 006356 LET @TCSR := @TCSR SET.BY #MAINT
2655 006356 052777 000004 172700 BIS #MAINT,@TCSR
2656 ; CLEAR INTERRUPTS
2657 006364 LET @RCR := @RCR CLR.BY #RCVRIE
2658 006364 042777 000100 172666 BIC #RCVRIE,@RCR
2659 ; CHANGE PRIORITY
2660 ; TO 0
2661 006372 012746 000000 MOV #PRO,-(SP) ; PUT NEW PS ON STACK
2662 006376 012746 006404 MOV #65,-(SP) ; PUT NEW PC ON STACK
2663 006402 000002 RTI ; POP NEW PC AND PS
2664 006404 65$:
2665 ; SEND A CHARACTER
2666 LET @TBUF := #0
2667 006404 105077 172660 CLRB @TBUF
2668 ; WAIT A MAXIMUM
2669 ; OF 50 MSEC FOR
2670 ; XMIT RDY TO SET IN TCSR
2671 CALL TIMER IN (<#50,#XMITRDY,TCSR,#SET)
2672 006410
2673 006410 010546 MOV R5,-(SP)
2674 006412 012745 177777 MOV #SET,-(R5)
2675 006416 016745 172642 MOV TCSR,-(R5)
    
```

KOL

```

2676 006422 012745 000200      MOV      #XMITRDY, -(RS)
2677 006426 012745 000050      MOV      #50, -(RS)
2678 006432 004767 002024      JSR      PC, TIMER
2679 006436 012605      MOV      (SP)+, RS
2680
2681 006440      ;SET INTERRUPT ENABLE
2682 006440 052777 000100 172612      BIS      #RCVRIE, @RCSR      LET @RCSR := @RCSR SET.BY #RCVRIE
2683
2684 006446      ;LET IT COME IN.
2685 006446 010546      MOV      RS, -(SP)      WAITUS 10
2686 006450 012745 000010      MOV      #10, -(RS)
2687 006454 004767 002260      JSR      PC, WAIT
2688 006460 012605      MOV      (SP)+, RS
2689
2690      ;DID HE DO IT RIGHT?
2691 006462      IF INTFLAG NE #1 THEN
2692 006462 026727 002340 000001      CMP      INTFLAG, #1
2693 006470 001411      BEQ      $136
2694
2695      ;NONE OCCURED
2696 006472      ; CAN NOT LEAVE WITH MAINT SET
2697 006472 042777 000004 172564      BIC      #MAINT, @TCSR      LET @TCSR := @TCSR CLR.BY #MAINT
2698 006500      IF INTFLAG EQ #0 THEN
2699 006500 005767 002322      TST      INTFLAG
2700 006504 001002      BNE      $137
2701
2702 006506      ;RECEIVER DID NOT INTERRUPT IN TIME
2703 006506 104111      ERROR    111      ERRHRD 111,,DIDNOT
2704
2705 006510      ;TWICE OR MORE
2706 006510 000401      BR       $140      ELSE
2707 006512      ;RECEIVER INTERRUPTED TWICE
2708      ERRHRD 112,,TWICE
2709 006512      ENDIF
2710 006512 104112      ERROR    112
2711 006514      ENDIF
2712 006514      $140:
2713 006514      $136:
2714 006514
2715      ;RESET MAINT. BIT.
2716 006514 042777 000004 172542      BIC      #MAINT, @TCSR      LET @TCSR := @TCSR CLR.BY #MAINT
2717 006514
2718 006522      ENDSUB
2719
2720
2721
2722
2723
2724
2725      ;INTERRUPT WITHOUT I E SET.
2726 006522      BGNSUB
2727 006522 012767 006530 172360      MOV      #64$, $LPERR
2728 006530      LET @TCSR := @TCSR SET.BY #MAINT
2729 006530 052777 000004 172526      BIS      #MAINT, @TCSR
2730
2731      ;CLEAR INTERRUPT FLAG

```





M05

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 65  
 DVDVCA.P11 08-AUG-77 09:16 T20 RECEIVER INTERRUPT LOGIC TEST

SEQ 0064

```

2788 006702 104115          ERROR 115
2789 006704
2790 006704          $142:
2791
2792 006704
2793 006704 000005          RESET
2794
2795
2796 006706
2797 006706 042777 000004 172350      BIC    #MAINT,@TCSR
2798 006714
2799
2800 006714
2801 006714 017704 172336          MOV    @DLVEC,R4
2802 006720
2803 006720 010146          MOV    R1,-(SP)      ;;PUSH R1 ON STACK
2804 006722 010246          MOV    R2,-(SP)      ;;PUSH R2 ON STACK
2805 006724 012701 000004          MOV    #R4,R1
2806 006730 010102          MOV    R1,R2
2807 006732 062702 000002          ADD    #2,R2
2808 006736 010221          MOV    R2,(R1)+
2809 006740 005011          CLR    (R1)
2810 006742 012602          MOV    (SP)+,R2
2811 006744 012601          MOV    (SP)+,R1
2812 006746
  
```

ENDIF  
 :CLEAR THE WORLD  
 BRESBT  
 :RESET MAINT. BIT.  
 LET @TCSR := @TCSR CLR.BY #MAINT  
 ENDSUB  
 LET R4 := @DLVEC  
 CLRVEC R4  
 ;;POP STACK INTO R2  
 ;;POP STACK INTO R1  
 ENDTST

```

2813 ;*****
2814 ;*****
2815 ;TEST 21 TEST ACTUAL DATA TRANSFERRED
2816 ;NON-INTERRUPT MAINTENANCE BIT SET
2817 ;*****
2818 TST21: SCOPE
2819 MOV #1,$TIMES ;DO 1 ITERATION
2820 MOV #21,$TESTN ;SET TEST NUMBER IN APT MAIL BOX
2821 ; IF #MAINTJUMP NOTSET IN SUSWR ORB APTCON EQ #TRUE
2822 BIT #MAINTJUMP,SUSWR
2823 BEQ $143
2824 CMPB APTCON,#TRUE
2825 BNE $144
2826 $143:
2827 ;EXIT TEST
2828 MOV #1,$TIMES
2829 BR TST22 ;;;EXIT THIS TEST
2830 ;ENDIF
2831 $144:
2832
2833 LET ERRCHK := #0
2834 CLR ERRCHK
2835 ;SET MAINT. BIT
2836 LET @TCSR := @TCSR SET.BY #MAINT
2837 BIS #MAINT,@TCSR
2838
2839 ;CHANGE PRIORITY
2840 ; TO 0
2841 MOV #PRO,-(SP) ;PUT NEW PS ON STACK
2842 MOV #64$,-(SP) ;PUT NEW PC ON STACK
2843 RTI ;POP NEW PC AND PS
2844 64$:
2845 ;GET DATA MASK.
2846 CALL DATLNG OUT <R1>
2847 SUB #1*2,R5
2848 JSR PC,DATLNG
2849 MOV (R5)+,R1
2850
2851 ;ALL BINARY CHAR.
2852 INCR R2 FROM #0 TO #377 BY #1
2853
2854 CLR R2
2855 BR $145
2856 $146:
2857 INC R2
2858 $145:
2859 CMP R2,#377
2860 BGT $147
2861
2862 ;TRANSMIT CHAR IN R2
2863 CALL TIMER IN <#50,#XMITRDY,TCSR,#SET>
2864 MOV R5,-(SP)
2865 MOV #SET,-(R5)
2866 MOV TCSR,-(R5)
2867 MOV #XMITRDY,-(R5)
2868

```



|      |        |        |        |        |       |                 |  |
|------|--------|--------|--------|--------|-------|-----------------|--|
| 2869 | 007104 | 012745 | 000050 |        | MOV   | #50,-(R5)       |  |
| 2870 | 007110 | 004767 | 001346 |        | JSR   | PC,TIMER        |  |
| 2871 | 007114 | 012605 |        |        | MOV   | (SP)+,R5        |  |
| 2872 | 007116 |        |        |        |       |                 | IF.ERROR THEN                            |
| 2873 | 007116 | 103003 |        |        | BCC   | \$150           | LET ERRCHK := ERRCHK SET.BY #BIT3        |
| 2874 | 007120 |        |        |        |       |                 | ENDIF                                    |
| 2875 | 007120 | 052767 | 000010 | 001206 | BIS   | #BIT3,ERRCHK    |  |
| 2876 | 007126 |        |        |        |       |                 |  |
| 2877 | 007126 |        |        |        |       |                 |  |
| 2878 |        |        |        |        |       |                 |  |
| 2879 |        |        |        |        |       |                 |  |
| 2880 | 007126 |        |        |        |       |                 | ;TRANSMIT IT                             |
| 2881 | 007126 | 110277 | 172136 |        | MOV   | R2,@TBUF        | LET @TBUF :B= R2                         |
| 2882 |        |        |        |        |       |                 |  |
| 2883 | 007132 |        |        |        |       |                 | CALL TIMER IN (<#50,#RCVRDONE,RCSR,#SET) |
| 2884 | 007132 | 010546 |        |        | MOV   | R5,-(SP)        |  |
| 2885 | 007134 | 012745 | 177777 |        | MOV   | #SET,-(R5)      |  |
| 2886 | 007140 | 016745 | 172114 |        | MOV   | RCSR,-(R5)      |  |
| 2887 | 007144 | 012745 | 000200 |        | MOV   | #RCVRDONE,-(R5) |  |
| 2888 | 007150 | 012745 | 000050 |        | MOV   | #50,-(R5)       |  |
| 2889 | 007154 | 004767 | 001302 |        | JSR   | PC,TIMER        |  |
| 2890 | 007160 | 012605 |        |        | MOV   | (SP)+,R5        |  |
| 2891 | 007162 |        |        |        |       |                 | IF.ERROR THEN                            |
| 2892 | 007162 | 103003 |        |        | BCC   | \$151           | LET ERRCHK := ERRCHK SET.BY #BIT4        |
| 2893 | 007164 |        |        |        |       |                 | ENDIF                                    |
| 2894 | 007164 | 052767 | 000020 | 001142 | BIS   | #BIT4,ERRCHK    |  |
| 2895 | 007172 |        |        |        |       |                 |  |
| 2896 | 007172 |        |        |        |       |                 |  |
| 2897 |        |        |        |        |       |                 |  |
| 2898 | 007172 |        |        |        |       |                 | ;AND SAVE IT                             |
| 2899 | 007172 | 017703 | 172064 |        | MOV   | @RBUF,R3        | LET R3 := @RBUF                          |
| 2900 |        |        |        |        |       |                 |  |
| 2901 |        |        |        |        |       |                 |  |
| 2902 |        |        |        |        |       |                 | ;COMPARE TO SEE IF WE RECEIVED IT ALL    |
| 2903 |        |        |        |        |       |                 |  |
| 2904 |        |        |        |        |       |                 | ;CLEAN OFF NON-DATA BITS                 |
| 2905 |        |        |        |        |       |                 | ;ON BOTH TRANSMITTED AND                 |
| 2906 | 007176 |        |        |        |       |                 | LET R4 := R2 CLR.BY R1                   |
| 2907 | 007176 | 010204 |        |        | MOV   | R2,R4           |  |
| 2908 | 007200 | 040104 |        |        | BIC   | R1,R4           |  |
| 2909 | 007202 |        |        |        |       |                 | LET R3 := R3 CLR.BY R1                   |
| 2910 | 007202 | 040103 |        |        | BIC   | R1,R3           |  |
| 2911 |        |        |        |        |       |                 |  |
| 2912 |        |        |        |        |       |                 | ;RECEIVED DATA                           |
| 2913 | 007204 |        |        |        |       |                 | IF R4 NE R3 THEN                         |
| 2914 | 007204 | 020403 |        |        | CMP   | R4,R3           |  |
| 2915 | 007206 | 001404 |        |        | BEQ   | \$152           |  |
| 2916 |        |        |        |        |       |                 |  |
| 2917 |        |        |        |        |       |                 | ;DATA COMPARE ERROR                      |
| 2918 | 007210 |        |        |        |       |                 | ;CAN NOT LEAVE WITH MAINT SET            |
| 2919 | 007210 | 042777 | 000004 | 172046 | BIC   | #MAINT,@TCSR    | LET @TCSR := @TCSR CLR.BY #MAINT         |
| 2920 | 007216 |        |        |        |       |                 | ERRHRD 116,COMP,SBWAS                    |
| 2921 | 007216 | 104116 |        |        | ERROR | 116             |  |
| 2922 |        |        |        |        |       |                 | ; <TRANSMITTED> <RECEIVED>               |
| 2923 | 007220 |        |        |        |       |                 | ENDIF                                    |
| 2924 | 007220 |        |        |        |       |                 |  |







```

3006 007406 005002          CLR    R2
3007 007410 000401          BR     $160
3008 007412                $161:    INC    R2
3009 007412 005202          $160:    CMP    R2,#377
3010 007414 020227 000377    BGT    $162
3011 007414 003057
3012 007420
3013
3014
3015                                ;TRANSMIT THE CHAR. IN R2.
3016                                ; MAKE SURE IT'S READY
3017                                CALL TIMER IN <#50,#XMITRDY,TCSR,#SET>
3018 007422
3019 007422 010546          MOV    R5,-(SP)
3020 007424 012745 177777    MOV    #SET,-(R5)
3021 007430 016745 171630    MOV    TCSR,-(R5)
3022 007434 012745 000200    MOV    #XMITRDY,-(R5)
3023 007440 012745 000050    MOV    #50,-(R5)
3024 007444 004767 001012    JSR   PC,TIMER
3025 007450 012605          MOV    (SP)+,R5
3026 007452
3027 007452 103005          BCC   $163
3028 007454 104123          ERROR 123
3029 007456
3030 007456 012767 000001 171474    MOV    #1,$TIMES
3031 007464 000440          BR     TST23
3032 007466
3033 007466                $163:    ;;EXIT THIS TEST
3034
3035                                ENDIF
3036 007466
3037 007466 110277 171576          MOVB  R2,@TBUF
3038
3039                                ;START IT ON ITS WAY
3040                                LET @TBUF:=R2
3041 007472 010546          MOV    R5,-(SP)
3042 007474 012745 177777    MOV    #SET,-(R5)
3043 007500 016745 171554    MOV    RCSR,-(R5)
3044 007504 012745 000200    MOV    #RCVRDONE,-(R5)
3045 007510 012745 000050    MOV    #50,-(R5)
3046 007514 004767 000742    JSR   PC,TIMER
3047 007520 012605          MOV    (SP)+,R5
3048 007522
3049 007522 103005          BCC   $164
3050 007524 104124          ERROR 124
3051
3052                                ;RECEIVER NEVER BECAME READY
3053                                EXIT TEST
3054 007526 012767 000001 171424    MOV    #1,$TIMES
3055 007534 000414          BR     TST23
3056 007536                $164:    ;;EXIT THIS TEST
3057
3058                                ENDIF
3059 007536
3060 007536 017703 171520          MOV   @RBUF,R3
3061                                ;RETRIEVE
3062                                LET R3:=@RBUF

```

```

3062
3063 007542
3064 007542 010204      MOV    R2,R4
3065 007544 040104      BIC    R1,R4
3066 007546
3067 007546 040103      BIC    R1,R3
3068
3069
3070 007550
3071 007550 020403      CMP    R4,R3
3072 007552 001401      BEQ    $165
3073
3074 007554
3075 007554 104117      ERROR  117
3076
3077 007556
3078 007556          $165:
3079
3080 007556
3081 007556 000715      BR     $161
3082 007560          $162:
3083
3084
3085
3086 007560
3087 007560 052777 000011 171472  BIS    #11,ARCSR
3088
3089
3090
3091 007566
3092
3093
3094
3095

```

```

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

LET R3 := R3 CLR.BY R1

;WE HAVE TROUBLE
IF R4 NE R3 THEN

;DATA COMPARE ERROR
ERRHRD 117,COMP,SBWAS

;(R2) (R3)
ENDIF

ENDINC ; R2

; NOW THAT THE TEST IS DONE
; WE WILL TOGGLE READER RUN
; TO TURN OFF THE SPECIAL MODULE.
LET ARCSR := ARCSR SET.BY #11

```

ENDTST

```

3096
3097
3098
3099
3100
3101 007566 000004
3102 007570 012767 000001 171362
3103 007576 012767 000023 171374
3104
3105 007604
3106 007604 032767 040000 171406
3107 007612 001404
3108 007614 126727 001446 000001
3109 007622 001004
3110 007624
3111 007624
3112 007624 012767 000001 171326
3113 007632 000540
3114 007634
3115 007634
3116
3117
3118 007634
3119 007634 162705 000002
3120 007640 004767 000774
3121 007644 012503
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133 007646 012746 000000
3134 007652 012746 007660
3135 007656 000002
3136 007660
3137
3138 007660
3139 007660 016701 171372
3140
3141 007664
3142 007664 012721 010040
3143 007670
3144 007670 012721 000340
3145
3146
3147 007674
3148 007674 012721 007776
3149 007700
3150 007700 012711 000340
3151

;*****
;*****
;TEST 23 FULL DATA TRANSFER WITH INTERRUPTS
; AND MAINTENANCE MODE.
;*****
†ST23: SCOPE
MOV #1,$TIMES ;;DO 1 ITERATION
MOV #23,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
BIT #MAINTJUMP,$USWR
BEQ $166
CMPB APTCON,#TRUE
BNE $167
$166:
EXIT TEST
MOV #1,$TIMES
BR †ST24 ;;EXIT THIS TEST
ENDIF
$167:
;;GET DATA MASK
CALL DATLNG OUT <R3>
SUB #1*2,$R5
JSR PC,DATLNG
MOV ($R5)+,$R3

;SINCE THIS WILL BE A FULL SPEED INTERRUPT TEST
;IF A DATA COMPARE ERROR OCCURS THE REPORT
;WILL BE DEFERRED UNTIL ALL DATA PATTERNS HAVE
;BEEN TRANSFERED. IF THE NUMBER OF CHAR. TRANSMITTED
;DOES NOT EQUAL THE NUMBER RECEIVED THIS ALSO WILL
;BE REPORTED.

;CHANGE PRIORITY
;...TO 0
MOV #PRO,-($SP) ;;PUT NEW PS ON STACK
MOV #64$,-($SP) ;;PUT NEW PC ON STACK
RTI ;;POP NEW PC AND PS

$E4$:
;GET VECTOR ADDRESS
LET R1 := DLVEC
;RCVR VECTOR
LET (R1)+ := #REC
LET (R1)+ := #PR7
;POINT TO TRANSMITTER VECTOR
;AND SET IT UP ALSO
LET (R1)+ := #TRAN
LET (R1) := #PR7
    
```



H06

```

3152                                     ; CLEAR ERROR COUNTER
3153 007704                               LET ERRCNT := #0
3154 007704 005067 000060                 CLR   ERRCNT
3155                                     ; START COUNT AT 0
3156 007710                               LET R1 := #0
3157 007710 005001                         CLR   R1
3158                                     ; RECEIVER STORAGE
3159 007712                               LET R2 := #0
3160 007712 005002                         CLR   R2
3161                                     ; # OF RECEIVED CHAR. COUNT.
3162 007714                               LET R4 := #0
3163 007714 005004                         CLR   R4
3164
3165 007716                               BRESET ;SET UP ALL REGISTERS
3166 007716 000005                         RESET
3167                                     ;SET UP MAINTENANCE
3168 007720                               LET @TCSR := @TCSR SET.BY #MAINT
3169 007720 052777 000004 171336          BIS   #MAINT,@TCSR
3170
3171                                     ;SET I.E. IN TRANSMITTER
3172 007726                               LET @TCSR := @TCSR SET.BY #XMITIE
3173 007726 052777 000100 171330          BIS   #XMITIE,@TCSR
3174                                     ;AND RECEIVER
3175 007734                               LET @RCR := @RCR SET.BY #RCVRIE
3176 007734 052777 000100 171316          BIS   #RCVRIE,@RCR
3177
3178                                     ; CLEAR OUR DONE AND ERROR FLAG WORD
3179
3180                                     ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3181 007742                               REPEAT
3182 007742                               UNTIL R4 HIS NUMBER
3183 007742                               $170:
3184 007742 020467 000024                 CMP   R4,NUMBER
3185 007746 103775                         BLO   $170
3186
3187 007750                               LET @TCSR := @TCSR CLR.BY #MAINT
3188 007750 042777 000004 171306          BIC   #MAINT,@TCSR
3189
3190                                     ; CHECK FOR DATA COMPARE ERRORS.
3191 007756                               IF ERRCNT NE #0 THEN
3192 007762 001401                         TST   ERRCNT
3193                                     ; DATA COMPARE ERROR
3194 007764                               ERRHRD 120,COMP,FIRST
3195 007764 104120                         ERROR 120
3196
3197                                     ; <R3> OCCURED, FIRST: SB <SB>, WAS <WAS>
3198 007766                               ENDIF
3199
3200                                     ;SKIP OVER SUPPORT ROUTINES & STORAGE
3201 007766 000462                         BR    TST24          ;;EXIT THIS TEST
3202
3203 007770 000000                         ERRCNT: 0
3204 007772 001000                         NUMBER: 1000
3205 007774 000000                         SB: .BYTE 0
3206 007775 000000                         WAS: .BYTE 0
3207

```

```

3208
3209
3210
3211 007776          TRAN:          BGNSRV  :TRANSMIT INTERRUPT HANDLER
3212 007776          :          TRAN
3213
3214          :          :SET UP FOR TRANSFER
3215 007776          :          LET HOLD := R1      CLR.BY R3
3216 007776 010167 000032      MOV    R1,HOLD
3217 010002 040367 000026      BIC    R3,HOLD
3218
3219 010006          :          :AND SEND.
3220 010006 016777 000022 171254      MOV    HOLD,@TBUF
3221          :          :INCREMENT CHAR COUNT,
3222 010014          :          LET R1 := R1 + #1
3223 010014 005201          :          :
3224          :          :ALL DONE
3225 010016          :          IF R1 EQ NUMBER THEN
3226 010016 020167 177750      CMP    R1,NUMBER
3227 010022 001003          :          BNE    $172
3228
3229 010024          :          :STOP INTERRUPT PROCESSING
3230 010024 042777 000100 171232      BIC    #XMITIE,@TCSR
3231 010032          :          LET @TCSR := @TCSR CLR.BY #XMITIE
3232 010032          :          ENDIF
3233          $172:
3234 010032 000401          :          BR    ZZZ
3235          :          :
3236 010034 000000          :          HOLD:0
3237          :          :
3238 010036          :          ZZZ:          ENDSRV
3239 010036 000002          :          RTI
3240
3241
3242
3243
3244 010040          :          :RECEIVER INTERRUPT HANDLER
3245 010040          :          REC
3246          :          :
3247          :          :GET CHAR IN + MASK IT
3248 010040          :          LET R2 := @RBUF CLR.BY R3
3249 010040 017702 171216      MOV    @RBUF,R2
3250 010044 040302          :          BIC    R3,R2
3251          :          :RHLD WILL CONTAIN EXPECTED INPUT
3252 010046          :          LET RHL := R4 CLR.BY R3
3253 010046 010467 000056      MOV    R4,RHL
3254 010052 040367 000052      BIC    R3,RHL
3255
3256          :          :DO THEY COMPARE
3257 010056          :          IF R2 NE RHL THEN
3258 010056 020267 000046      CMP    R2,RHL
3259 010062 001412          :          BEQ    $173
3260          :          :FIRST ERROR
3261 010064          :          IF ERRCNT EQ #0 THEN
3262 010064 005767 177700      TST    ERRCNT
3263 010070 001005          :          BNE    $174

```

J06

```

3264                                     ;SAVE RECORD OF FIRST MISS
3265 010072                               LET SB :B= RHLD
3266 010072 116767 000032 177674        MOVB  RHLD,SB
3267 010100                               LET WAS :B= R2
3268 010100 110267 177671              MOVB  R2,WAS
3269 010104                               ENDIF
3270 010104                               $174:
3271                                     ;COUNT IT.
3272 010104                               LET ERRCNT := ERRCNT + #1
3273 010104 005267 177660              INC   ERRCNT
3274 010110                               ENDIF
3275 010110                               $173:
3276
3277                                     ;COUNT THIS CHAR.
3278 010110                               LET R4 := R4 + #1
3279 010110 005204                      INC   R4
3280
3281 010112                               ;ALL DONE?
3282 010112 020467 177654              CMP   R4,NUMBER
3283 010116 001003                      BNE   $175
3284
3285 010120                               ;STOP RECEIVER INTERRUPTS
3286 010120 042777 000100 171132      BIC   #RCVRIE,#RCSR
3287
3288                                     ;INDICATE ALL DONE TO TIMER
3289 010126                               ;MAIN REPEAT LOOP IS CHECKING
3290 010126                               ENDIF
3291
3292                                     ;FOR 'R4 = NUMBER' ALSO
3293 010126 000401                      BR   ZZZZ
3294
3295 010130 000000                      RHLD:0
3296 010132                               ZZZZ:
3297 010132                               ENDSRV
3298 010132 000002                      RTI
3299
3300 010134                               ENDTST
3301
3302
3303

```



3304  
3305  
3306  
3307  
3308  
3309  
3310  
3311  
3312  
3313  
3314  
3315  
3316  
3317  
3318  
3319  
3320  
3321  
3322  
3323  
3324  
3325  
3326  
3327  
3328  
3329  
3330  
3331  
3332  
3333  
3334  
3335  
3336  
3337  
3338  
3339  
3340  
3341  
3342  
3343  
3344  
3345  
3346  
3347  
3348  
3349  
3350  
3351  
3352  
3353  
3354  
3355  
3356  
3357  
3358  
3359

010134 000004  
010136 012767 000010 171014  
010144 012767 000024 171026  
010152  
010152 032767 040000 171040  
010160 001404  
010162 032767 010000 171030  
010170 001004  
010172  
010172  
010172 012767 000001 170760  
010200 000456  
010202  
010202  
010202 005067 000126  
010206  
010206 052777 000004 171050  
010214  
010214 052777 000001 171042  
010222  
010222 012777 000252 171040  
010230  
010230  
010230  
010230 032777 000200 171022  
010236 001774  
010240  
010240 105777 171016  
010244 001404  
010246  
010246 052767 000001 000060  
010254  
010254 000413  
010256  
010256  
010256 032767 100000 170734  
010264 001407  
010266  
010266 032777 020000 170766

\*\*\*\*\*  
\*\*\*\*\*  
\*TEST 24 TEST BREAK GENERATION LOGIC  
\* TRANSMIT KNOWN CHAR WITH BREAK SET  
\* AND COMPARE RECEIVED WITH 0.  
\* FRAMING ERROR WILL ALSO BE CHECKED  
\* IF ERROR BITS ARE ENABLED.  
\*\*\*\*\*

TST24: SCOPE  
MOV #10,STIMES ;;DO 10 ITERATIONS  
MOV #24,STESTN ;;SET TEST NUMBER IN APT MAIL BOX  
IF #MAINTJUMP NOTSETIN SUSWR OR #BRK NOTSETIN SU  
BIT #MAINTJUMP,SUSWR  
BEQ \$176  
BIT #BRK,SUSWR  
BNE \$177  
\$176: EXIT TEST  
MOV #1,STIMES  
BR TST25 ;;;EXIT THIS TEST  
\$177: ENDF  
LET ERRCHK := #0 ; CLEAR ERROR WORD  
;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  
\$200: UNTIL #RCVRDONE SETIN @RCSR  
BIT #RCVRDONE,@RCSR  
BEQ \$200  
IFB @RBUF NE #0 THEN  
; BREAK DID NOT EQUAL 0  
LET ERRCHK := ERRCHK SET.BY #BIT0  
ELSE  
\$201: IF #ERRBITS SETIN SUSWR THEN  
BIT #ERRBITS,SUSWR  
BEQ \$203  
IF #FRERR NOTSETIN @RBUF THEN  
BIT #FRERR,@RBUF

```

3360 010274 001003          BNE      $204
3361 010276
3362 010276 052767 000002 000030  BIS      #BIT1,ERRCHK          LET ERRCHK := ERRCHK SET.BY #BIT1
3363 010304
3364 010304          $204:          ENDIF
3365 010304          ENDIF
3366 010304          $203:          ENDIF
3367 010304          ENDIF
3368 010304          $202:
3369
3370 010304          BRESET ;CLEAN UP
3371 010304 000005          RESET
3372
3373 010306
3374 010306 032767 000001 000020  BIT      #BIT0,ERRCHK          IF #BIT0 SETIN ERRCHK THEN
3375 010314 001401          BEQ      $205          ERRHRD 121 ;BREAK ERROR
3376 010316
3377 010316 104121          ERROR   121          ENDIF
3378 010320
3379 010320          $205:          IF #BIT1 SETIN ERRCHK THEN
3380 010320
3381 010320 032767 000002 000006  BIT      #BIT1,ERRCHK          ERRHRD 122 ; FRAMING ERROR
3382 010326 001401          BEQ      $206          ENDIF
3383 010330
3384 010330 104122          ERROR   122          EXIT
3385 010332          $206:          ;;EXIT THIS TEST
3386 010332
3387 010332
3388 010332 000401          BR      TST25          ENDTST
3389 010334 000000          ERRCHK: .WORD 0
3390 010336
3391
    
```

M06

```

3392
3393
3394
3395
3396 010336 000004
3397 010340 012767 000001 170612
3398 010346 104401 010354
3399 010352 000404
3400
3401 010364
3402 010364 016746 170664
3403 010370 104402
3404 010372 104401 010400
3405 010376 000405
3406
3407 010412
3408 010412 016746 170640
3409 010416 104402
3410 010420 104401 010426
3411 010424 000405
3412
3413 010440
3414 010440 016746 170446
3415 010444 104405
3416 010446 005067 170440
3417 010452 104401 001171
3418 010456 000167 171454

```

```

*****
*TEST 25      NOT A TEST - SEND BACK TO LOOP
*****
TST25: 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      SERTTL,-(SP) ;;SAVE SERTTL FOR TYPEOUT
        TYPDS     ;;GO TYPE--DECIMAL ASCII WITH SIGN
        CLR      SERTTL ;;RESET FOR NEXT DEVICE/PASS
        TYPE     ,SCLF
        JMP      LOOP          ; BACK UP TO THE BEGINNING

```



3419  
3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466  
3467  
3468  
3469  
3470  
3471  
3472  
3473  
3474

010462  
010462

000001  
000000

010462  
010462  
010470  
010470  
010476  
010476

016567 000004 000136  
016567 000000 000132  
112767 000000 000126

010504  
010504

\$211:

010504  
010504  
010512  
010514  
010514  
010522  
010522  
010524  
010524  
010532  
010532

036577 000002 000114  
001004  
112767 000000 000111  
000403  
112767 177777 000101

\$213:

\$214:

```
;;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
```

010532

```

3475 010532 126765 000075 000006      CMPB  HOLDSC,SETCLR(R5)
3476 010540 001003                    BNE   $215
3477                                     ; JUST THE THING WE NEEDED
3478 010542                    LET   FLAG :B= #TRUE
3479 010542 112767 000001 000062      MOVB  #TRUE,FLAG
3480 010550                    ENDIF
3481 010550      $215:
3482
3483 010550                    EXIFB  FLAG EQ #TRUE OR TIMSAV LE #0
3484 010550 126727 000056 000001      CMPB  FLAG,#TRUE
3485 010556 001414                    BEQ   $212
3486 010560 005767 000044            TST   TIMSAV
3487 010564 003411                    BLE   $212
3488                                     ; ONE WAY OR THE OTHER, WE ARE DONE
3489                                     ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
3490
3491 010566                    WAITUS 10          ;WAIT FOR 10 MILLI-SECONDS
3492 010566 010546                    MOV   R5, -(SP)
3493 010570 012745 000010            MOV   #10, -(R5)
3494 010574 004767 000140            JSR   PC, WAIT
3495 010600 012605                    MOV   (SP)+, R5
3496 010602                    LET   TIMSAV := TIMSAV - #1 ; COUNTING DOWN
3497 010602 005367 000022            DEC   TIMSAV
3498 010606                    ENDLOOP          ; CONTINUED AT THE TOP
3499 010606 000736                    BR    $211
3500 010610      $212:
3501
3502                                     ; ONLY 2 WAYS TO GET HERE
3503                                     ; 1). WE RAN OUT OF TIME---ERROR !!
3504                                     ; 2). THE BIT IS IN THE CORRECT CONDITION--GOOD !!
3505
3506 010610                    IFB   FLAG EQ #TRUE THEN
3507 010610 126727 000016 000001      CMPB  FLAG,#TRUE
3508 010616 001001                    BNE   $216
3509 010620                    RETURN NO.ERROR    ; GOOD
3510 010620 000405                    BR    $207
3511 010622                    ENDIF
3512 010622      $216:
3513 010622                    RETURN ERROR      ; BAD
3514 010622 000261                    SEC
3515 010624 000404                    BR    $210
3516
3517 010626 000000                    REGSAV: .WORD 0
3518 010630 000000                    TIMSAV: .WORD 0
3519 010632 000                    FLAG: .BYTE 0
3520 010633 000                    HOLDSC: .BYTE 0
3521                                     ; WE ARE DONE GO BACK HOME
3522 010634                    ENDRTN
3523 010634      $207:
3524 010634 000241                    CLC
3525 010636      $210:
3526 010636 000207                    RTS   PC

```

3527  
3528  
3529  
3530  
3531  
3532  
3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544  
3545  
3546  
3547  
3548  
3549  
3550  
3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568  
3569  
3570  
3571  
3572  
3573  
3574  
3575

010640  
010640

010640  
010640 005065 000000  
010644  
010644 016767 170350 000062  
010652 016746 000056  
010656 042716 000017  
010662 042667 000046  
  
010666  
010666 012767 000001 170376  
010674 000402  
010676  
010676 005267 170370  
010702  
010702 026767 170364 000024  
010710 003006  
010712  
010712 006365 000000  
010716  
010716 052765 000001 000000  
010724  
010724 000764  
010726  
010726  
010726 005165 000000  
010732  
010732 000401  
010734 000000  
010736  
010736  
010736 000207

```
*****  
ROUTINE DATLNG <MASK>  
DATLNG:  
* ROUTINE:DATLNG  
* THIS ROUTINE SETS UP A MASK FOR DATA, WITH  
* INPUT - NOTHING IS PASSED TO THIS ROUTINE  
* BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:  
* SUSWR-- THE WORD FOR SOFTWARE PARAMETERS  
* DATA-- A MASK FOR THE LOCATION OF THE OCTAL  
* NUMBER OF DATA BITS  
* OUTPUT----  
* MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED  
* THE NUMBER OF WHICH IS DEFINED IN SUSWR WORD.  
*****
```

```
;;*****  
LET MASK(R5) := #0 ; START  
CLR MASK(R5)  
LET NUMBR := SUSWR AND #DATA  
MOV SUSWR, NUMBR  
MOV NUMBR, -(SP)  
BIC #DATA, (SP)  
BIC (SP)+, NUMBR  
  
INCR I FROM #1 TO NUMBR BY #1  
MOV #1, I  
BR $221  
$222: INC I  
$221: CMP I, NUMBR  
BGT $223  
LET MASK(R5) := MASK(R5) SHIFT 1  
LET MASK(R5) := MASK(R5) SET.BY #1  
ASL MASK(R5)  
BIS #1, MASK(R5)  
ENDINC  
BR $222  
$223: LET MASK(R5) := COMP MASK(R5)  
COM MASK(R5)  
RETURN  
BR $217  
NUMBR:0  
ENDRTN  
$217:  
$220:  
RTS PC
```



```

3576
3577
3578 010740
3579 010740
3580
3581
3582
3583
3584
3585
3586
3587 010740 010146
3588 010742 010246
3589 010744 010346
3590 010746
3591 010746 016501 000000
3592 010752
3593 010752 012702 000001
3594 010756 000402
3595 010760
3596 010760 062702 000001
3597 010764
3598 010764 020201
3599 010766 101010
3600 010770
3601 010770 005003
3602 010772 000401
3603 010774
3604 010774 005203
3605 010776
3606 010776 020327 000700
3607 011002 003001
3608 011004
3609 011004 000773
3610 011006
3611 011006
3612 011006 000764
3613 011010
3614 011010 012603
3615 011012 012602
3616 011014 012601
3617 011016
3618 011016
3619 011016
3620 011016 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
MOV #1,R2
BR $226
$227: ADD #01,R2
$226: CMP R2,R1
BHI $230
INCR R3 FROM #0 TO #700 BY #1
CLR R3
BR $231
$232: INC R3
$231: CMP R3,#700
BGT $233
ENDINC
BR $232
$233: ENDINC
BR $227
$230: MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
ENDRTN
$224:
$225: RTS PC

```

3621  
3622  
3623  
3624  
3625  
3626  
3627  
3628  
3629  
3630  
3631  
3632  
3633  
3634  
3635  
3636  
3637

011020

011020

005267 000002

011024

000002

011024

000000

011026

```

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

```

3638 011030  
3639 011030  
3640  
3641  
3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650 011030  
3651 011030 112767 000000 000230  
3652 011036  
3653 011036 112767 000000 000223  
3654 011044  
3655 011044  
3656 011044  
3657 011044 005767 000200  
3658 011050 001027  
3659 011050  
3660 011050 026727 000174 000001  
3661 011060 001003  
3662 011062  
3663 011062 005067 000164  
3664 011066  
3665 011066 000403  
3666 011070  
3667 011070  
3668 011070 004767 000370  
3669  
3670 011074  
3671 011074  
3672 011074 012600  
3673 011076  
3674 011076  
3675 011076  
3676 011076 012767 000001 000144  
3677 011104  
3678 011104 012767 000001 170072  
3679 011112  
3680 011112 016767 170132 000134  
3681 011120  
3682 011120 016767 170120 000130  
3683 011126  
3684 011126 000410  
3685 011130  
3686 011130  
3687 011130 012704 000010  
3688 011134  
3689 011134 006167 000110  
3690 011140  
3691 011140 060467 000110  
3692 011144  
3693 011144 060467 000106

ROUTINE CYCLE

CYCLE:

\*\*\*\*\*  
\* ROUTINE: CYCLE  
\* THIS ROUTINE CAUSES ADRS TO POINT TO THE  
\* ADDRESS OF DLV11-F UNDER TEST, ADRS +2 TO  
\* POINT TO THE VECTOR OF THE DLV11-F UNDER TEST.  
\* IT KEEPS TRACK OF THE CURRENT DEVICE AND BIT  
\* MASKS. THE CONSOLE IS TREATED SPECIAL BY THIS ROUTINE.  
\* IT IS ONLY TESTED ONCE IF UNDER APT, IF NOT UNDER APT  
\* ALL TESTS THAT REQUIRE THE MAINT BIT ARE NOT RUN.  
\*\*\*\*\*

```
LET APTCON :B= #FALSE ; SET DEFAULT VALUE
LET CONSOLE :B= #FALSE
REPEAT ; UNTIL BITMASK SETIN SDEVN
IF BITMASK EQ #0 THEN
  TST BITMASK
  BNE $237
  IF INITFLAG EQ #1 THEN
    CMP INITFLAG,#1
    BNE $240
    LET INITFLAG := #0
  ELSE
    BR $241
$240: CALL SEOP ; AS A SUBROUTINE
JSR PC,SEOP
SPECIALADDRESS: ; BECAUSE SEOP RETURNS AS A JUMP
LET RO := POP
MOV (SP)+,RO
ENDIF
$241: LET BITMASK := #1
LET SDEVCT := #1
LET ADDRESS := $BASE
LET VECTOR := $VECT1
ELSE
  BR $242
$237: 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
```



```

3694 011150                               ENDIF
3695 011150                               $242:
3696 011150                               UNTIL BITMASK SETIN SDEVN
3697 011150 036767 000074 170074          BIT   BITMASK,SDEVN
3698 011156 001732                          BEQ   $236
3699 011160                               IF BITMASK EQ #BIT15 THEN
3700 011160 026727 000064 100000          CMP   BITMASK,#BIT15
3701 011166 001023                          BNE   $243
3702 011170                               LET CONSOLE :B= #TRUE
3703 011170 112767 000001 000071          MOVB  #TRUE,CONSOLE
3704 011176                               LET ADDRESS := CONADR
3705 011176 016767 000060 000050          MOV   CONADR,ADDRESS
3706 011204                               LET VECTOR := CONVECT
3707 011204 016767 000054 000044          MOV   CONVECT,VECTOR
3708                               IF #CONMAINT NOTSETIN SUSWR THEN
3709                               LET NOCONMAINT :B= #TRUE
3710                               ENDIF
3711 011212                               IF #APTENV SETIN SENV THEN ; APT MODE
3712 011212 032767 000001 167774          BIT   #APTENV,SENV
3713 011220 001406                          BEQ   $244
3714 011222                               IF $PASS NE #0 THEN ; NOT FIRST PASS
3715 011222 005767 167754          TST  $PASS
3716 011226 001403                          BEQ   $245
3717                               ; DEFINE DEVICE AS APT CONSOLE
3718 011230                               LET APTCON :B= #TRUE
3719 011230 112767 000001 000030          MOVB  #TRUE,APTCON
3720 011236                               ENDIF ; FIRST PASS
3721 011236                               ENDIF ; APT
3722 011236                               ENDIF ; BITMASK
3723 011236                               $245:
3724 011236                               $244:
3725 011236                               $243:
3726 011236                               LET ADRS := #ADDRESS
3727 011236 012701 011254          MOV   #ADDRESS,ADRS
3728 011242                               LET $DEVCT := $DEVCT + #1
3729 011242 005267 167736          INC  $DEVCT
3730 011246                               RETURN
3731 011246 000411          BR   $234
3732 011250 100000          BITMASK: 100000 ; CONSOLE FIRST
3733 011252 000001          INITFLAG: 1
3734 011254 000000          ADDRESS: 0
3735 011256 000000          VECTOR: 0
3736 011260 000000          OK: 0
3737 011262 177560          CONADR: 177560 ; CONSOLE ADDRESS
3738 011264 000060          CONVECT: 60 ; CONSOLE VECTOR
3739 011266 000          APTCON: .BYTE 0
3740 011267 000          CONSOLE: .BYTE 0
3741 011270 000          NOCONMANT: .BYTE 0
3742 011272                               .EVEN
3743 011272                               ENDRTN
3744 011272                               $234:
3745 011272                               $235:
3746 011272                               RTS   PC
3747 011272 000207
3748 011272
3749

```

MAINDEC-11-DVDVC-A      MACY11 27(1006)    08-AUG-77 09:20    PAGE 86  
DVDVCA.P11      08-AUG-77 09:16      ROUTINE - CYCLE

H07

SEQ 0085

3750

```

3751
3752 011274 ROUTINE MYTYPE
3753 011274 MYTYPE:
3754 ;:*****
3755 011274 104401 011302 TYPE 65$ ;:TYPE ASCIZ STRING
3756 011300 000405 BR 64$ ;:GET OVER THE ASCIZ
3757 ;:65$: .ASCIZ <CRLF>*TEST # *
3758 011314 64$: MOV $TESTN,-(SP) ;:SAVE $TESTN FOR TYPEOUT
3759 011314 016746 167660 TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
3760 011320 104402 TYPE 67$ ;:TYPE ASCIZ STRING
3761 011322 104401 011330 BR 66$ ;:GET OVER THE ASCIZ
3762 011326 000405 ;:67$: .ASCIZ *,ERROR # *
3763 66$: ;:67$: .ASCIZ *,ERROR # *
3764 011342 66$: MOV $ITEMB,$FATAL ;:APT FATAL ERROR NUMBER
3765 011342 116767 167546 167626 MOV $FATAL,-(SP) ;:SAVE $FATAL FOR TYPEOUT
3766 011350 016746 167622 TYPOS ;:GO TYPE--OCTAL ASCII
3767 011354 104403 .BYTE 6 ;:TYPE 6 DIGITS
3768 011356 006 .BYTE 0 ;:SUPPRESS LEADING ZEROS
3769 011357 000 TYPE 69$ ;:TYPE ASCIZ STRING
3770 011360 104401 011366 BR 68$ ;:GET OVER THE ASCIZ
3771 011364 000404 ;:69$: .ASCIZ *,PC = *
3772 68$: ;:69$: .ASCIZ *,PC = *
3773 011376 68$: MOV $ERRPC,-(SP) ;:SAVE $ERRPC FOR TYPEOUT
3774 011376 016746 167514 TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
3775 011402 104402 TYPE 71$ ;:TYPE ASCIZ STRING
3776 011404 104401 011412 BR 70$ ;:GET OVER THE ASCIZ
3777 011410 000404 ;:71$: .ASCIZ *,CSR: *
3778 70$: ;:71$: .ASCIZ *,CSR: *
3779 011422 70$: MOV $DLADD,-(SP) ;:SAVE $DLADD FOR TYPEOUT
3780 011422 016746 167626 TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
3781 011426 104402 TYPE 73$ ;:TYPE ASCIZ STRING
3782 011430 104401 011436 BR 72$ ;:GET OVER THE ASCIZ
3783 011434 000405 ;:73$: .ASCIZ *,VECTOR: *
3784 ;:73$: .ASCIZ *,VECTOR: *
3785 011450 72$: MOV $DLVEC,-(SP) ;:SAVE $DLVEC FOR TYPEOUT
3786 011450 016746 167602 TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
3787 011454 104402 TYPE ;:TYPE ASCIZ STRING
3788 011456 104401 001171 ;:GET OVER THE ASCIZ
3789 011462 ENDRTN
3790 011462 $246:
3791 011462 $247:
3792 011462 000207 RTS PC

```



```

3793
3794
3795
3796
3797
3798
3799
3800
3801
3802 011464
3803 011464 000004
3804 011466 005067 167410
3805 011472 005067 167462
3806 011476 005267 167500
3807 011502 042767 100000 167472
3808 011510 005327
3809 011512 000001
3810 011514 003022
3811 011516 012737
3812 011520 000001
3813 011522 011512
3814 011524 104401 011571
3815 011530 016746 167446
3816 011534 104405
3817 011536 104401 011566
3818 011542 013700 000042
3819 011546 001405
3820 011550 000005
3821 011552 004710
3822 011554 000240
3823 011556 000240
3824 011560 000240
3825 011562
3826 011562 000137
3827 011564 011074
3828 011566 377 377 000
3829 011571 015 042412 042116
3830 011576 050040 051501 02C123
3831 011604 000043

```

.SBTTL END OF PASS ROUTINE

```

;*****
;INCREMENT THE PASS NUMBER ($PASS)
;INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
;TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
;IF THERES A MONITOR GO TO IT
;IF THERE ISN'T JUMP TO SPECIALADDRESS

```

SEOP:

```

SCOPE
CLR $STNM ;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)+,a(PC)+ ;RESTORE COUNTER
SENDCT: .WORD 1
TYPE $SENDMG ;TYPE "END PASS #"
MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
TYPDS ;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE $SENULL ;TYPE A NULL CHARACTER
SGET42: MOV a#42,R0 ;GET MONITOR ADDRESS
BEQ $DOAGN ;BRANCH IF NO MONITOR
RESET ;CLEAR THE WORLD
SENDAD: JSR PC,(R0) ;GO TO MONITOR
NOP ;SAVE ROOM
NOP ;FOR
NOP ;ACT11
$DOAGN: JMP a(PC)+ ;RETURN
SRTNAD: .WORD SPECIALADDRESS
SENULL: .BYTE -1,-1,0 ;NULL CHARACTER STRING
SENDMG: .ASCIZ <15><12>/END PASS #/

```

.SBTTL POWER DOWN AND UP ROUTINES

```

3832
3833
3834 ;:*****
3835 :POWER DOWN ROUTINE
3836 011606 012737 011752 000024 $PWRDN: MOV $SILLUP, @#PWRVEC ;; SET FOR FAST UP
3837 011614 012737 000340 000026 MOV #340, @#PWRVEC+2 ;; PRIO:7
3838 011622 010046 MOV R0, -(SP) ;; PUSH R0 ON STACK
3839 011624 010146 MOV R1, -(SP) ;; PUSH R1 ON STACK
3840 011626 010246 MOV R2, -(SP) ;; PUSH R2 ON STACK
3841 011630 010346 MOV R3, -(SP) ;; PUSH R3 ON STACK
3842 011632 010446 MOV R4, -(SP) ;; PUSH R4 ON STACK
3843 011634 010546 MOV R5, -(SP) ;; PUSH R5 ON STACK
3844 011636 017746 167276 MOV @SMR, -(SP) ;; PUSH @SMR ON STACK
3845 011642 010667 000110 MOV SP, $SAVR6 ;; SAVE SP
3846 011646 012737 011660 000024 MOV $PWRUP, @#PWRVEC ;; SET UP VECTOR
3847 011654 000000 HALT
3848 011656 000776 BR .-2 ;; HANG UP
3849
3850 ;:*****
3851 :POWER UP ROUTINE
3852 011660 012737 011752 000024 $PWRUP: MOV $SILLUP, @#PWRVEC ;; SET FOR FAST DOWN
3853 011666 016706 000064 MOV $SAVR6, SP ;; GET SP
3854 011672 005067 000060 CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
3855 011676 005267 000054 1$: INC $SAVR6 ;; WAIT FOR THE INC
3856 011702 001375 BNE 1$ ;; OF WORD
3857 011704 012677 167230 MOV (SP)+, @SMR ;; POP STACK INTO @SMR
3858 011710 012605 MOV (SP)+, R5 ;; POP STACK INTO R5
3859 011712 012604 MOV (SP)+, R4 ;; POP STACK INTO R4
3860 011714 012603 MOV (SP)+, R3 ;; POP STACK INTO R3
3861 011716 012602 MOV (SP)+, R2 ;; POP STACK INTO R2
3862 011720 012601 MOV (SP)+, R1 ;; POP STACK INTO R1
3863 011722 012600 MOV (SP)+, R0 ;; POP STACK INTO R0
3864 011724 012737 011606 000024 MOV $PWRDN, @#PWRVEC ;; SET UP THE POWER DOWN VECTOR
3865 011732 012737 000340 000026 MOV #340, @#PWRVEC+2 ;; PRIO:7
3866 011740 104401 TYPE ;; REPORT THE POWER FAILURE
3867 011742 011760 SPWRMG: .WORD $POWER ;; POWER FAIL MESSAGE POINTER
3868 011744 012716 MOV (PC)+, (SP) ;; RESTART AT LOOP
3869 011746 002136 SPWRAD: .WORD LOOP ;; RESTART ADDRESS
3870 011750 000002 RTI
3871 011752 000000 $SILLUP: HALT ;; THE POWER UP SEQUENCE WAS STARTED
3872 011754 000776 BR .-2 ;; BEFORE THE POWER DOWN WAS COMPLETE
3873 011756 000000 $SAVR6: 0 ;; PUT THE SP HERE
3874 011760 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
3875 011766 000122
3876 .EVEN

```

.SBTTL TYPE ROUTINE

```

3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894 011770 105767 167163 STYPE: TSTB STPFLG ;; IS THERE A TERMINAL?
3895 011774 100002 BPL 1$ ;; BR IF YES
3896 011776 000000 HALT ;; HALT HERE IF NO TERMINAL
3897 012000 000430 BR 3$ ;; LEAVE
3898 012002 010046 1$: MOV RO,-(SP) ;; SAVE RO
3899 012004 017600 000002 MOV 32(SP),RO ;; GET ADDRESS OF ASCIZ STRING
3900 012010 122767 000001 167176 CMPB #APTENV,SENV ;; RUNNING IN APT MODE
3901 012016 001011 BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
3902 012020 132767 000100 167167 BITB #APTPOOL,SENV ;; SPOOL MESSAGE TO APT
3903 012026 001405 BEQ 62$ ;; NO GO CHECK FOR CONSOLE
3904 012030 010067 000004 MOV RO,61$ ;; SETUP MESSAGE ADDRESS FOR APT
3905 012034 004767 000774 JSR PC,SATY3 ;; SPOOL MESSAGE TO APT
3906 012040 000000 61$: .WORD 0 ;; MESSAGE ADDRESS
3907 012042 132767 000040 167145 62$: BITB #APTCSUP,SENV ;; APT CONSOLE SUPPRESSED
3908 012050 001003 BNE 60$ ;; YES, SKIP TYPE OUT
3909 012052 112046 2$: MOVB (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
3910 012054 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
3911 012056 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
3912 012060 012600 60$: MOV (SP)+,RO ;; RESTORE RO
3913 012062 062716 000002 3$: ADD #2,(SP) ;; ADJUST RETURN PC
3914 012066 000002 RTI ;; RETURN
3915 012070 122716 000011 4$: CMPB #HT,(SP) ;; BRANCH IF <HT>
3916 012074 001430 BEQ 8$
3917 012076 122716 000200 CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
3918 012102 001006 BNE 5$
3919 012104 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
3920 012106 104401 TYPE ;; TYPE A CR AND LF
3921 012110 001171 SCRLF
3922 012112 105067 000130 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
3923 012116 000755 BR 2$ ;; GET NEXT CHARACTER
3924 012120 004767 000056 5$: JSR PC,STYPEC ;; GO TYPE THIS CHARACTER
3925 012124 126726 167026 6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
3926 012130 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
3927 012132 016746 167016 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
3928 AND THE NULL CHAR.
3929 012136 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
3930 012142 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
3931 012144 004767 000032 JSR PC,STYPEC ;; GO TYPE A NULL
3932 012150 105367 000072 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT

```



```

3933 012154 000770          BR      7$          ;;LOOP
3934
3935          ;HORIZONTAL TAB PROCESSOR
3936
3937 012156 112716 000040      8$:   MOVB   #' (SP)          ;;REPLACE TAB WITH SPACE
3938 012162 004767 000014      9$:   JSR    PC,$TYPEC          ;;TYPE A SPACE
3939 012166 132767 000007 000052  BITB   #',$SCHARCNT          ;;BRANCH IF NOT AT
3940 012174 001372          BNE    9$          ;;TAB STOP
3941 012176 005726          TST    (SP)+          ;;POP SPACE OFF STACK
3942 012200 000724          BR     2$          ;;GET NEXT CHARACTER
3943 012202 105777 166742      $TYPEC: TSTB  2$STPS          ;;WAIT UNTIL PRINTER IS READY
3944 012206 100375          BPL   $TYPEC
3945 012210 116677 000002 166734  MOVB   2(SP),2$STPB          ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3946 012216 122766 000015 000002  CMPB   #CR,2(SP)          ;;IS CHARACTER A CARRIAGE RETURN?
3947 012224 001003          BNE    1$          ;;BRANCH IF NO
3948 012226 105067 000014          CLRB  $SCHARCNT          ;;YES--CLEAR CHARACTER COUNT
3949 012232 000406          BR     $TYPEX          ;;EXIT
3950 012234 122766 000012 000002  1$:   CMPB   #LF,2(SP)          ;;IS CHARACTER A LINE FEED?
3951 012242 001402          BEQ   $TYPEX          ;;BRANCH IF YES
3952 012244 105227          INCB  (PC)+          ;;COUNT THE CHARACTER
3953 012246 000000          $SCHARCNT: .WORD 0          ;;CHARACTER COUNT STORAGE
3954 012250 000207          $TYPEX: RTS   PC
3955

```

```

3956 .SBTTL TTY INPUT ROUTINE
3957
3958 ;:*****
3959 .ENABL LSB
3960
3961 ;:*****
3962 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
3963 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
3964 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
3965 ;*WHEN OPERATING IN TTY FLAG MODE.
3966 012252 022767 000176 166660 $CKSWR: CMP      $SWREG,SWR      ;: IS THE SOFT-SWR SELECTED?
3967 012260 001074          BNE      15$          ;: BRANCH IF NO
3968 012262 105777 166656      TSTB    $STKS        ;: CHAR THERE?
3969 012266 100071          BPL      15$          ;: IF NO, DON'T WAIT AROUND
3970 012270 117746 166652      MOVB    $STKB,-(SP)    ;: SAVE THE CHAR
3971 012274 042716 177600      BIC     $C177,(SP)    ;: STRIP-OFF THE ASCII
3972 012300 022726 000007      CMP     $7,(SP)+      ;: IS IT A CONTROL G?
3973 012304 001062          BNE     15$          ;: NO, RETURN TO USER
3974 012306 126727 166622 000001  CMPB   $AUTOB,#1     ;: ARE WE RUNNING IN AUTO-MODE?
3975 012314 001456          BEQ     15$          ;: BRANCH IF YES
3976
3977 012316 104401 012777          TYPE   ,SCNTLG       ;: ECHO THE CONTROL-G (↑G)
3978 012322 104401 013004      SGTSWR: TYPE   ,SMSWR       ;: TYPE CURRENT CONTENTS
3979 012326 016746 165644      MOV    $SWREG,-(SP)  ;: SAVE SWREG FOR TYPEOUT
3980 012332 104402          TYPOC          ;: GO TYPE--OCTAL ASCII(ALL DIGITS)
3981 012334 104401 013015      TYPE   ,SMNEW       ;: PROMPT FOR NEW SWR
3982 012340 005046          19$: CLR    -(SP)      ;: CLEAR COUNTER
3983 012342 005046          CLR    -(SP)      ;: THE NEW SWR
3984 012344 105777 166574      7$: TSTB  $STKS      ;: CHAR THERE?
3985 012350 100375          BPL    7$          ;: IF NOT TRY AGAIN
3986
3987 012352 117746 166570      MOVB   $STKB,-(SP)  ;: PICK UP CHAR
3988 012356 042716 177600      BIC    $C177,(SP)  ;: MAKE IT 7-BIT ASCII
3989
3990
3991
3992 012362 021627 000025          9$: CMP    (SP),#25    ;: IS IT A CONTROL-U?
3993 012366 001005          BNE    10$          ;: BRANCH IF NOT
3994 012370 104401 012772          TYPE   ,SCNTLU      ;: YES, ECHO CONTROL-U (↑U)
3995 012374 062706 000006      20$: ADD   $6,SP     ;: IGNORE PREVIOUS INPUT
3996 012400 000757          BR     19$          ;: LET'S TRY IT AGAIN
3997
3998
3999 012402 021627 000015          10$: CMP   (SP),#15    ;: IS IT A <CR>?
4000 012406 001022          BNE   16$          ;: BRANCH IF NO
4001 012410 005766 000004          TST   4(SP)        ;: YES, IS IT THE FIRST CHAR?
4002 012414 001403          BEQ   11$          ;: BRANCH IF YES
4003 012416 016677 000002 166514      MOV   2(SP),$SWR    ;: SAVE NEW SWR
4004 012424 062706 000006      11$: ADD   $6,SP     ;: CLEAR UP STACK
4005 012430 104401 001171          14$: TYPE  ,SCRLF     ;: ECHO <CR> AND <LF>
4006 012434 126727 166475 000001  CMPB  $INTAG,#1     ;: RE-ENABLE TTY KBD INTERRUPTS?
4007 012442 001003          BNE   15$          ;: BRANCH IF NOT
4008 012444 012777 000100 166472      MOV   $100,$STKS   ;: RE-ENABLE TTY KBD INTERRUPTS
4009 012452 000002          15$: RTI                    ;: RETURN
4010 012454 004767 177522          16$: JSR   PC,$TYPEC   ;: ECHO CHAR
4011 012460 021627 000060      CMP   (SP),#60     ;: CHAR < 0?

```



```

4012 012464 002420          BLT      18$          ;; BRANCH IF YES
4013 012466 021627 000067    CMP      (SP),#67    ;; CHAR > 7?
4014 012472 003015          BGT      18$          ;; BRANCH IF YES
4015 012474 042726 000060    BIC      #60,(SP)+   ;; STRIP-OFF ASCII
4016 012500 005766 000002    TST      2(SP)      ;; IS THIS THE FIRST CHAR
4017 012504 001403          BEQ      17$          ;; BRANCH IF YES
4018 012506 006316          ASL      (SP)       ;; NO, SHIFT PRESENT
4019 012510 006316          ASL      (SP)       ;; CHAR OVER TO MAKE
4020 012512 006316          ASL      (SP)       ;; ROOM FOR NEW ONE.
4021 012514 005266 000002    17$: INC      2(SP)    ;; KEEP COUNT OF CHAR
4022 012520 056616 177776    BIS      -2(SP),(SP) ;; SET IN NEW CHAR
4023 012524 000707          BR       7$          ;; GET THE NEXT ONE
4024 012526 104401 001170    16$: TYPE $QUES    ;; TYPE ?<CR><LF>
4025 012532 000720          BR       20$         ;; SIMULATE CONTROL-U
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037 012534 011646          SRDCHR: MOV      (SP),-(SP) ;; PUSH DOWN THE PC
4038 012536 016666 000004 000002  MOV      4(SP),2(SP) ;; SAVE THE PS
4039 012544 105777 166374 1$: TSTB    @STKS    ;; WAIT FOR
4040 012550 100375          BPL      1$          ;; A CHARACTER
4041 012552 117766 166370 000004  MOVB    @STKB,4(SP) ;; READ THE TTY
4042 012560 042766 177600 000004  BIC      #C(177),4(SP) ;; GET RID OF JUNK IF ANY
4043 012566 026627 000004 000023  CMP      4(SP),#23  ;; IS IT A CONTROL-5?
4044 012574 001013          BNE      3$          ;; BRANCH IF NO
4045 012576 105777 146342 2$: TSTB    @STKS    ;; WAIT FOR A CHARACTER
4046 012602 100375          BPL      2$          ;; LOOP UNTIL ITS THERE
4047 012604 117746 166336  MOVB    @STKB,-(SP) ;; GET CHARACTER
4048 012610 042716 177600  BIC      #C(177),(SP) ;; MAKE IT 7-BIT ASCII
4049 012614 022627 000021  CMP      (SP)+,#21  ;; IS IT A CONTROL-Q?
4050 012620 001366          BNE      2$          ;; IF NOT DISCARD IT
4051 012622 000750          BR       1$          ;; YES, RESUME
4052 012624 026627 000004 000140 3$: CMP      4(SP),#140 ;; IS IT UPPER CASE?
4053 012632 002407          BLT      4$          ;; BRANCH IF YES
4054 012634 026627 000004 000175  CMP      4(SP),#175 ;; IS IT A SPECIAL CHAR?
4055 012642 003003          BGT      4$          ;; BRANCH IF YES
4056 012644 042766 000040 000004  BIC      #40,4(SP)  ;; MAKE IT UPPER CASE
4057 012652 000002          4$: RTI          ;; GL BACK TO USER
4058
4059
4060
4061
4062
4063
4064
4065 012654 010346          SRDLIN: MOV      R3,-(SP) ;; SAVE R3
4066 012656 012703 012762 1$: MOV      #TTYIN,R3 ;; GET ADDRESS
4067 012662 022703 012772 2$: CMP      #TTYIN+8.,R3 ;; BUFFER FULL?

```

\*\*\*\*\*  
THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY

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

\*\*\*\*\*  
THIS ROUTINE WILL INPUT A STRING FROM THE TTY

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



|      |        |        |        |        |          |        |                  |  |  |
|------|--------|--------|--------|--------|----------|--------|------------------|--|--|
| 4068 | 012666 | 101405 |        |        |          | BLOS   | 4S               |  | :: BR IF YES                               |
| 4069 | 012670 | 104410 |        |        |          | RDCHR  |                  |  | :: GO READ ONE CHARACTER FROM THE TTY      |
| 4070 | 012672 | 112613 |        |        |          | MOVB   | (SP)+, (R3)      |  | :: GET CHARACTER                           |
| 4071 | 012674 | 122713 | 000177 |        | 10S:     | CMPB   | #177, (R3)       |  | :: IS IT A RUBOUT                          |
| 4072 | 012700 | 001003 |        |        |          | BNE    | 3S               |  | :: SKIP IF NOT                             |
| 4073 | 012702 | 104401 | 001170 |        | 4S:      | TYPE   | \$QUES           |  | :: TYPE A '?'                              |
| 4074 | 012706 | 000763 |        |        |          | BR     | 1S               |  | :: CLEAR THE BUFFER AND LOOP               |
| 4075 | 012710 | 111367 | 000044 |        | 3S:      | MOVB   | (R3), 9S         |  | :: ECHO THE CHARACTER                      |
| 4076 | 012714 | 104401 | 012760 |        |          | TYPE   | 9S               |  |  |
| 4077 | 012720 | 122723 | 000015 |        |          | CMPB   | #15, (R3)+       |  | :: CHECK FOR RETURN                        |
| 4078 | 012724 | 001356 |        |        |          | BNE    | 2S               |  | :: LOOP IF NOT RETURN                      |
| 4079 | 012726 | 105063 | 177777 |        |          | CLRB   | -1(R3)           |  | :: CLEAR RETURN (THE 15)                   |
| 4080 | 012732 | 104401 | 001172 |        |          | TYPE   | \$LF             |  | :: TYPE A LINE FEED                        |
| 4081 | 012736 | 012603 |        |        |          | MOV    | (SP)+, R3        |  | :: RESTORE R3                              |
| 4082 | 012740 | 011646 |        |        |          | MOV    | (SP)-, (SP)      |  | :: ADJUST THE STACK AND PUT ADDRESS OF THE |
| 4083 | 012742 | 016666 | 000004 | 000002 |          | MOV    | 4(SP), 2(SP)     |  | :: FIRST ASCII CHARACTER ON IT             |
| 4084 | 012750 | 012766 | 012762 | 000004 |          | MOV    | #STTYIN, 4(SP)   |  |  |
| 4085 | 012756 | 000002 |        |        |          | RTI    |                  |  | :: RETURN                                  |
| 4086 | 012760 | 000    |        |        | 9S:      | .BYTE  | 0                |  | :: STORAGE FOR ASCII CHAR. TO TYPE         |
| 4087 | 012761 | 000    |        |        |          | .BYTE  | 0                |  | :: TERMINATOR                              |
| 4088 | 012762 | 000010 |        |        |          | .BLKB  | 8.               |  | :: RESERVE 8 BYTES FOR TTY INPUT           |
| 4089 | 012772 | 052536 | 005015 | 000    | \$TTYIN: | .ASCIZ | /↑U/<15><12>     |  | :: CONTROL "U"                             |
| 4090 | 012777 | 136    | 006507 | 000012 | \$CNTLG: | .ASCIZ | /↑G/<15><12>     |  | :: CONTROL "G"                             |
| 4091 | 013004 | 005015 | 053523 | 020122 | \$MSWR:  | .ASCIZ | <15><12>/SWR = / |  |  |
| 4092 | 013012 | 020075 | 000    |        |          |        |                  |  |  |
| 4093 | 013015 | 040    | 047040 | 053505 | \$MNEW:  | .ASCIZ | / NEW = /        |  |  |
| 4094 | 013022 | 036440 | 000040 |        |          |        |                  |  |  |

```

.SBTTL APT COMMUNICATIONS ROUTINE
*****
4095
4096
4097
4098 013026 112767 000001 000236 $ATY1: MOVB #1,$FFLG ;; TO REPORT FATAL ERROR
4099 013034 112767 000001 000226 $ATY3: MOVB #1,$MFLG ;; TO TYPE A MESSAGE
4100 013042 000403 BR SATYC
4101 013044 112767 000001 000220 $ATY4: MOVB #1,$FFLG ;; TO ONLY REPORT FATAL ERROR
4102 013052 SATYC:
4103 013052 010046 MOV RO,-(SP) ;; PUSH RO ON STACK
4104 013054 010146 MOV R1,-(SP) ;; PUSH R1 ON STACK
4105 013056 105767 000206 TSTB SMFLG ;; SHOULD TYPE A MESSAGE?
4106 013062 001450 BEQ 5$ ;; IF NOT: BR
4107 013064 122767 000001 166122 CMPB #APTENV,$ENV ;; OPERATING UNDER APT?
4108 013072 001031 BNE 3$ ;; IF NOT: BR
4109 013074 132767 000100 166113 BITB #APTPOOL,$ENVM ;; SHOULD SPOOL MESSAGES?
4110 013102 001425 BEQ 3$ ;; IF NOT: BR
4111 013104 017600 000004 MOV #4(SP),RO ;; GET MESSAGE ADDR.
4112 013110 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDR.
4113 013116 005767 166052 1$: TST $MSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
4114 013122 001375 BNE 1$ ;; IF NOT: WAIT
4115 013124 010067 166060 MOV RO,$MSGAD ;; PUT ADDR IN MAILBOX
4116 013130 105720 2$: TSTB (RO)+ ;; FIND END OF MESSAGE
4117 013132 001376 BNE 2$
4118 013134 166700 166050 SJB $MSGAD,RO ;; SUB START OF MESSAGE
4119 013140 006200 ASR RO ;; GET MESSAGE LGTH IN WORDS
4120 013142 010067 166044 MOV RO,$MSGLGTH ;; PUT LENGTH IN MAILBOX
4121 013146 012767 000004 166020 MOV #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
4122 013154 000413 BR 5$
4123 013156 017667 000004 000016 3$: MOV #4(SP),4$ ;; PUT MSG ADDR IN JSR LINKAGE
4124 013164 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDRESS
4125 013172 016746 164600 MOV 177776,-(SP) ;; PUSH 177776 ON STACK
4126 013176 004767 176566 JSR PC,$TYPE ;; CALL TYPE MACRO
4127 013202 000000 4$: .WORD 0
4128 013204 5$:
4129 013204 105767 000062 10$: TSTB $FFLG ;; SHOULD REPORT FATAL ERROR?
4130 013210 001416 BEQ 12$ ;; IF NOT: BR
4131 013212 005767 165776 TST $ENV ;; RUNNING UNDER APT?
4132 013216 001413 BEQ 12$ ;; IF NOT: BR
4133 013220 005767 165750 11$: TST $MSGTYPE ;; FINISHED LAST MESSAGE?
4134 013224 001375 BNE 11$ ;; IF NOT: WAIT
4135 013226 017667 000004 165742 MOV #4(SP),$FATAL ;; GET ERROR #
4136 013234 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDR.
4137 013242 005267 165726 INC $MSGTYPE ;; TELL APT TO TAKE ERROR
4138 013246 105067 000020 12$: CLRB $FFLG ;; CLEAR FATAL FLAG
4139 013252 105067 000013 CLRB $LFLG ;; CLEAR LOG FLAG
4140 013256 105067 000006 CLRB $MFLG ;; CLEAR MESSAGE FLAG
4141 013262 012601 MOV (SP)+,R1 ;; POP STACK INTO R1
4142 013264 012600 MOV (SP)+,RO ;; POP STACK INTO RO
4143 013266 000207 RTS PC ;; RETURN
4144 013270 000 $MFLG: .BYTE 0 ;; MESSG. FLAG
4145 013271 000 $LFLG: .BYTE 0 ;; LOG FLAG
4146 013272 000 $FFLG: .BYTE 0 ;; FATAL FLAG
4147 013274 .EVEN
4148 000200 APTSIZE=200
4149 000001 APTENV=001
4150 000100 APTPOOL=100

```

E08

MAINDEC-11-DVDVC-A    MACY11 27(1006)    08-AUG-77 09:20    PAGE 96  
DVDVCA.P11    08-AUG-77 09:16    APT COMMUNICATIONS ROUTINE  
4151    000040    APTCSUP=040

SEQ 0095



```

4152 .SBTTL ERROR HANDLER ROUTINE
4153
4154 ;:*****
4155 ;:THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
4156 ;:SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4157 ;:AND GO TO MYTYPE ON ERROR
4158 ;:THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4159 ;:#SW15=1 HALT ON ERROR
4160 ;:#SW13=1 INHIBIT ERROR TYPEOUTS
4161 ;:#SW10=1 BELL ON ERROR
4162 ;:#SW09=1 LOOP ON ERROR
4163 ;:#CALL
4164 ;:# ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
4165
4166 013274 SERROR:
4167 013274 104407 7S: CKSWR ;; TEST FOR CHANGE IN SOFT-SWR
4168 013276 105267 165601 INCB SERFLG ;; SET THE ERROR FLAG
4169 013302 001775 BEQ 7S ;; DON'T LET THE FLAG GO TO ZERO
4170 013304 016777 165572 165630 MOV $STNM, @DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
4171 013312 032777 002000 165620 BIT #BIT10, @SWR ;; BELL ON ERROR?
4172 013320 001402 BEQ 1S ;; NO - SKIP
4173 013322 104401 001164 TYPE $BELL ;; RING BELL
4174 013326 005267 165560 1S: INC $ERTTL ;; COUNT THE NUMBER OF ERRORS
4175 013332 011667 165560 MOV (SP), $ERRPC ;; GET ADDRESS OF ERROR INSTRUCTION
4176 013336 162767 000002 165552 SUB #2, $ERRPC
4177 013344 117767 165546 165542 MOVB @ERRPC, $ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
4178 013352 032777 020000 165560 BIT #BIT13, @SWR ;; SKIP TYPEOUT IF SET
4179 013360 001004 BNE 20S ;; SKIP TYPEOUTS
4180 013362 004767 175706 JSR PC, MYTYPE ;; GO TO USER ERROR ROUTINE
4181 013366 104401 001171 TYPE , $CRLF
4182 013372
4183 013372 122767 000001 165614 20S: CMPB #APTENV, $ENV ;; RUNNING IN APT MODE
4184 013400 001007 BNE 2S ;; NO SKIP APT ERROR REPORT
4185 013402 116767 165506 000004 MOVB $ITEMB, 21S ;; SET ITEM NUMBER AS ERROR NUMBER
4186 013410 004767 177430 JSR PC, $ATY4 ;; REPORT FATAL ERROR TO APT
4187 013414 000 21S: .BYTE 0
4188 013415 000 .BYTE 0
4189 013416 000777 22S: BR 22S ;; APT ERROR LOOP
4190 013420 005777 165514 2S: TST @SWR ;; HALT ON ERROR
4191 013424 100002 BPL 3S ;; SKIP IF CONTINUE
4192 013426 000000 HALT ;; HALT ON ERROR!
4193 013430 104407 CKSWR ;; TEST FOR CHANGE IN SOFT-SWR
4194 013432 032777 001000 165500 3S: BIT #BIT09, @SWR ;; LOOP ON ERROR SWITCH SET?
4195 013440 001402 BEQ 4S ;; BR IF NO
4196 013442 016716 165442 MOV $LPERR, (SP) ;; FUDGE RETURN FOR LOOPING
4197 013446 005767 165510 4S: TST $ESCAPE ;; CHECK FOR AN ESCAPE ADDRESS
4198 013452 001402 BEQ 5S ;; BR IF NONE
4199 013454 016716 165502 MOV $ESCAPE, (SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE
4200 013460
4201 013460 022737 011552 000042 5S: CMP #SENDAD, @42 ;; ACT-11 AUTO-ACCEPT?
4202 013466 001001 BNE 6S ;; BRANCH IF NO
4203 013470 000000 HALT ;; YES
4204 013472
4205 013472 000002 6S: RTI ;; RETURN
    
```

```

4206 .SBTTL SCOPE HANDLER ROUTINE
4207
4208 ::*****
4209 #THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4210 #AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4211 #AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
4212 #THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4213 #SW14=1 LOOP ON TEST
4214 #SW11=1 INHIBIT ITERATIONS
4215 #SM09=1 LOOP ON ERROR
4216 #SM08=1 LOOP ON TEST IN SWR<7:0>
4217 #CALL
4218 # SCOPE ;;SCOPE=IOT
4219
4220 $SCOPE:
4221 013474 104407 CKSWR ;; TEST FOR CHANGE IN SOFT-SWR
4222 013476 032777 040000 165434 1$: BIT #BIT14,$SWR ;; LOOP ON PRESENT TEST?
4223 013504 001114 BNE $OVER ;; YES IF SW14=1
4224 :*****START OF CODE FOR THE XOR TESTER*****
4225 013506 000416 $XTSTR: BR 6$ ;; IF RUNNING ON THE "XOR" TESTER CHANGE
4226 ;; THIS INSTRUCTION TO A "NOP" (NOP=240)
4227 013510 013746 000004 MOV $ERRVEC,-($P) ;; SAVE THE CONTENTS OF THE ERROR VECTOR
4228 013514 012737 013534 000004 MOV $S,$ERRVEC ;; SET FOR TIMEOUT
4229 013522 005737 177060 TST $177060 ;; TIME OUT ON XOR?
4230 013526 012637 000004 MOV ($P)+,$ERRVEC ;; RESTORE THE ERROR VECTOR
4231 013532 000463 BR $SVLAD ;; GO TO THE NEXT TEST
4232 013534 022626 5$: CMP ($P)+,($P)+ ;; CLEAR THE STACK AFTER A TIME OUT
4233 013536 012637 000004 MOV ($P)+,$ERRVEC ;; RESTORE THE ERROR VECTOR
4234 013542 000423 BR 7$ ;; LOOP ON THE PRESENT TEST
4235 013544 6$:;*****END OF CODE FOR THE XOR TESTER*****
4236 013544 032777 000400 165366 BIT #BIT08,$SWR ;; LOOP ON SPEC. TEST?
4237 013552 001404 BEQ 2$ ;; BR IF NO
4238 013554 127767 165360 165320 CMPB $SWR,$TSTNM ;; ON THE RIGHT TEST? SWR<7:0>
4239 013562 001465 BEQ $OVER ;; BR IF YES
4240 013564 105767 165313 2$: TSTB SERFLG ;; HAS AN ERROR OCCURRED?
4241 013570 001421 BEQ 3$ ;; BR IF NO
4242 013572 126767 165317 165303 CMPB $ERMAX,SERFLG ;; MAX. ERRORS FOR THIS TEST OCCURRED?
4243 013600 101015 BHI 3$ ;; BR IF NO
4244 013602 032777 001000 165330 BIT #BIT09,$SWR ;; LOOP ON ERROR?
4245 013610 001404 BEQ 4$ ;; BR IF NO
4246 013612 016767 165272 165266 7$: MOV $LPERR,$LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
4247 013620 000446 BR $OVER
4248 013622 105067 165255 4$: CLRB SERFLG ;; ZERO THE ERROR FLAG
4249 013626 005067 165326 CLR $TIMES ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
4250 013632 000415 BR 1$ ;; ESCAPE TO THE NEXT TEST
4251 013634 032777 004000 165276 3$: BIT #BIT11,$SWR ;; INHIBIT ITERATIONS?
4252 013642 001011 BNE 1$ ;; BR IF YES
4253 013644 005767 165332 TST $PASS ;; IF FIRST PASS OF PROGRAM
4254 013650 001406 BEQ 1$ ;; INHIBIT ITERATIONS
4255 013652 005267 165226 INC $ICNT ;; INCREMENT ITERATION COUNT
4256 013656 026767 165276 165220 CMP $TIMES,$ICNT ;; CHECK THE NUMBER OF ITERATIONS MADE
4257 013664 002024 BGE $OVER ;; BR IF MORE ITERATION REQUIRED
4258 013666 012767 000001 165210 1$: MOV #1,$ICNT ;; REINITIALIZE THE ITERATION COUNTER
4259 013674 016767 000052 165256 MOV $MXCNT,$TIMES ;; SET NUMBER OF ITERATIONS TO DO
4260 013702 105267 165174 $SVLAD: INCB $TSTNM ;; COUNT TEST NUMBERS
4261 013706 116767 165170 165264 MOVB $TSTNM,$TESTN ;; SET TEST NUMBER IN APT MAILBOX

```

H08

MAINDEC-11-DVDVC-A MACY11 27(1006) 08-AUG-77 09:20 PAGE 99  
DVDVCA.P11 08-AUG-77 09:16 SCOPE HANDLER ROUTINE

SEQ 0098

|      |        |        |        |        |               |                 |   |
|------|--------|--------|--------|--------|---------------|-----------------|---|
| 4262 | 013714 | 011667 | 165166 |        | MOV           | (SP),SLPADR     | :::SAVE SCOPE LOOP ADDRESS              |
| 4263 | 013720 | 011667 | 165164 |        | MOV           | (SP),SLPERR     | :::SAVE ERROR LOOP ADDRESS              |
| 4264 | 013724 | 005077 | 165232 |        | CLR           | SESCAPE         | :::CLEAR THE ESCAPE FROM ERROR ADDRESS  |
| 4265 | 013730 | 112767 | 000001 | 165157 | MOVB          | #1,SERMAX       | :::ONLY ALLOW ONE(1) ERROR ON NEXT TEST |
| 4266 | 013736 | 016777 | 165140 | 165176 | SOVER: MOV    | STSTNM,SDISPLAY | :::DISPLAY TEST NUMBER                  |
| 4267 | 013744 | 016716 | 165136 |        | MOV           | SLPADR,(SP)     | :::FUDGE RETURN ADDRESS                 |
| 4268 | 013750 | 000002 |        |        | RTI           |                 | :::FIXES PS                             |
| 4269 | 013752 | 003720 |        |        | SMXCNT: 2000. |                 | :::MAX. NUMBER OF ITERATIONS            |



4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281  
4282  
4283  
4284  
4285  
4286  
4287  
4288  
4289  
4290  
4291  
4292  
4293  
4294  
4295  
4296  
4297  
4298  
4299  
4300  
4301  
4302  
4303  
4304  
4305  
4306  
4307  
4308  
4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325

013754  
013754 010046  
013756 010146  
013760 010246  
013762 010346  
013764 010546  
013766 012746 020200  
013772 016605 000020  
013776 100004  
014000 005405  
014002 112766 000055 000001  
014010 005000  
014012 012703 014170  
014016 112723 000040  
014022 005002  
014024 016001 014160  
014030 160105  
014032 002402  
014034 005202  
014036 000774  
014040 060105  
014042 005702  
014044 001002  
014046 105716  
014050 100407  
014052 106316  
014054 103003  
014056 116663 000001 177777  
014064 052702 000060  
014070 052702 000040  
014074 110223  
014076 005720  
014100 020027 000010  
014104 002746  
014106 003002  
014110 010502  
014112 000764  
014114 105726  
014116 100003  
014120 116663 177777 177776  
014126 105013  
014130 012605  
014132 012603  
014134 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

STYPDS:  
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 #SDBLK,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  
MOVB -1(SP),-2(R3) ;:YES--SET THE SIGN FOR TYPING  
9\$: CLRB (R3) ;:SET THE TERMINATOR  
MOV (SP)+,R5 ;:POP STACK INTO R5  
MOV (SP)+,R3 ;:POP STACK INTO R3  
MOV (SP)+,R2 ;:POP STACK INTO R2

J08

MAINDEC-11-DVDVC-A  
DVDVCA.P11

MACY11 27(1006)  
08-AUG-77 09:16

08-AUG-77 09:20 PAGE 101  
CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0100

|      |        |        |        |        |         |             |                       |
|------|--------|--------|--------|--------|---------|-------------|-----------------------|
| 4326 | 014136 | 012601 |        |        | MOV     | (SP)+,R1    | ::POP STACK INTO R1   |
| 4327 | 014140 | 012600 |        |        | MOV     | (SP)+,R0    | ::POP STACK INTO R0   |
| 4328 | 014142 | 104401 | 014170 |        | TYPE    | \$DBLK      | ::NOW TYPE THE NUMBER |
| 4329 | 014146 | 016666 | 000002 | 000004 | MOV     | 2(SP),4(SP) | ::ADJUST THE STACK    |
| 4330 | 014154 | 012616 |        |        | MOV     | (SP)+,(SP)  |                       |
| 4331 | 014156 | 000002 |        |        | RTI     |             | ::RETURN TO USER      |
| 4332 | 014160 | 023420 |        |        | \$DTBL: | 10000.      |                       |
| 4333 | 014162 | 001750 |        |        |         | 1000.       |                       |
| 4334 | 014164 | 000144 |        |        |         | 100.        |                       |
| 4335 | 014166 | 000012 |        |        |         | 10.         |                       |
| 4336 | 014170 | 000004 |        |        | \$DBLK: | .BLKW 4     |                       |

4337  
4338  
4339  
4340  
4341  
4342  
4343  
4344  
4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352  
4353  
4354  
4355  
4356  
4357  
4358  
4359  
4360  
4361  
4362  
4363  
4364  
4365  
4366  
4367  
4368  
4369  
4370  
4371  
4372  
4373  
4374  
4375  
4376  
4377  
4378  
4379  
4380  
4381  
4382  
4383  
4384  
4385  
4386  
4387  
4388  
4389  
4390  
4391  
4392

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

;*****
;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;OCTAL (ASCII) NUMBER AND TYPE IT.
;STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
;CALL:
;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;      TYPOS    ;;CALL FOR TYPEOUT
;      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;      .BYTE   M              ;;M=1 OR 0
;                               ;;1=TYPE LEADING ZEROS
;                               ;;0=SUPPRESS LEADING ZEROS
;
;STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
;STYPOS OR STYPOC
;CALL:
;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;      TYPON    ;;CALL FOR TYPEOUT
;
;STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
;CALL:
;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;      TYPOC    ;;CALL FOR TYPEOUT
;
;STYPOS: MOV      2(SP),-(SP)    ;;PICKUP THE MODE
;        MOV     1(SP),SOFILL   ;;LOAD ZERO FILL SWITCH
;        MOV     (SP)+,SOMODE+1 ;;NUMBER OF DIGITS TO TYPE
;        ADD    #2,(SP)        ;;ADJUST RETURN ADDRESS
;        BR     STYPON
;
;STYPOC: MOV     #1,SOFILL      ;;SET THE ZERO FILL SWITCH
;        MOV     #6,SOMODE+1    ;;SET FOR SIX(6) DIGITS
;
;STYPON: MOV     #5,SOCNT       ;;SET THE ITERATION COUNT
;        MOV     R3,-(SP)       ;;SAVE R3
;        MOV     R4,-(SP)       ;;SAVE R4
;        MOV     R5,-(SP)       ;;SAVE R5
;        MOV     SOMODE+1,R4    ;;GET THE NUMBER OF DIGITS TO TYPE
;        NEG     R4
;        ADD     #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
;        MOV     R4,SOMODE      ;;SAVE IT FOR USE
;        MOV     SOFILL,R4     ;;GET THE ZERO FILL SWITCH
;        MOV     12(SP),R5     ;;PICKUP THE INPUT NUMBER
;        CLR    R3             ;;CLEAR THE OUTPUT WORD
;        ROL    R5             ;;ROTATE MSB INTO "C"
;        BR     3$            ;;GO DO MSB
;        ROL    R5             ;;FORM THIS DIGIT
;
;        ROL    R5
;        ROL    R5
;        ROL    R5
;        MOV    R5,R3
;
;3$:    ROL    R3              ;;GET LSB OF THIS DIGIT
;        DECB  SOMODE         ;;TYPE THIS DIGIT?
;        BPL   7$            ;;BR IF NO
;        BIC   #177770,R3    ;;GET RID OF JUNK
;        BNE   4$            ;;TEST FOR 0
;        TST   R4            ;;SUPPRESS THIS 0?
;        BEQ   5$            ;;BR IF YES
;
;1$:    ROL    R5
;        BR     3$
;
;2$:    ROL    R5
;        ROL    R5
;        ROL    R5
;
;5$:    MOV    R5,R3
;        ROL    R3
;        DECB  SOMODE
;        BPL   7$
;        BIC   #177770,R3
;        BNE   4$
;        TST   R4
;        BEQ   5$
    
```

014200 017646 000000  
 014204 116667 000001 000211  
 014212 112667 000207  
 014216 062716 000002  
 014222 000406  
 014224 112767 000001 000171  
 014232 112767 000006 000165  
 014240 112767 000005 000154  
 014246 010346  
 014250 010446  
 014252 010546  
 014254 116704 000145  
 014260 005404  
 014262 062704 000006  
 014266 110467 000132  
 014272 116704 000125  
 014276 016605 000012  
 014302 005003  
 014304 006105  
 014306 000404  
 014310 006105  
 014312 006105  
 014314 006105  
 014316 010503  
 014320 006103  
 014322 105367 000076  
 014326 100016  
 014330 042703 177770  
 014334 001002  
 014336 005704  
 014340 001403



|      |        |        |               |          |       |             |                                    |
|------|--------|--------|---------------|----------|-------|-------------|------------------------------------|
| 4393 | 014342 | 005204 |               | 4\$:     | INC   | R4          | :: DON'T SUPPRESS ANYMORE 0'S      |
| 4394 | 014344 | 052703 | 000060        |          | BIS   | #'0,R3      | :: MAKE THIS DIGIT ASCII           |
| 4395 | 014350 | 052703 | 000040        | 5\$:     | BIS   | #' ,R3      | :: MAKE ASCII IF NOT ALREADY       |
| 4396 | 014354 | 110367 | 000040        |          | MOVB  | R3,8\$      | :: SAVE FOR TYPING                 |
| 4397 | 014360 | 104401 | 014420        |          | TYPE  | 8\$         | :: GO TYPE THIS DIGIT              |
| 4398 | 014364 | 105367 | 000032        | 7\$:     | DECB  | \$OCNT      | :: COUNT BY 1                      |
| 4399 | 014370 | 003347 |               |          | BGT   | 2\$         | :: BR IF MORE TO DO                |
| 4400 | 014372 | 002402 |               |          | BLT   | 6\$         | :: BR IF DONE                      |
| 4401 | 014374 | 005204 |               |          | INC   | R4          | :: INSURE LAST DIGIT ISN'T A BLANK |
| 4402 | 014376 | 000744 |               |          | BR    | 2\$         | :: GO DO THE LAST DIGIT            |
| 4403 | 014400 | 012605 |               | 6\$:     | MOV   | (SP)+,R5    | :: RESTORE R5                      |
| 4404 | 014402 | 012604 |               |          | MOV   | (SP)+,R4    | :: RESTORE R4                      |
| 4405 | 014404 | 012603 |               |          | MOV   | (SP)+,R3    | :: RESTORE R3                      |
| 4406 | 014406 | 016666 | 000002 000004 |          | MOV   | 2(SP),4(SP) | :: SET THE STACK FOR RETURNING     |
| 4407 | 014414 | 012616 |               |          | MOV   | (SP)+,(SP)  |                                    |
| 4408 | 014416 | 000002 |               |          | RTI   |             | :: RETURN                          |
| 4409 | 014420 | 000    |               | 8\$:     | .BYTE | 0           | :: STORAGE FOR ASCII DIGIT         |
| 4410 | 014421 | 000    |               |          | .BYTE | 0           | :: TERMINATOR FOR TYPE ROUTINE     |
| 4411 | 014422 | 000    |               | \$OCNT:  | .BYTE | 0           | :: OCTAL DIGIT COUNTER             |
| 4412 | 014423 | 000    |               | \$OFILL: | .BYTE | 0           | :: ZERO FILL SWITCH                |
| 4413 | 014424 | 000000 |               | \$OMODE: | .WORD | 0           | :: NUMBER OF DIGITS TO TYPE        |

```

4414
4415
4416
4417
4418
4419
4420
4421
4422 014426 010046
4423 014430 016600 000002
4424 014434 005740
4425 014436 111000
4426 014440 006300
4427 014442 016000 014462
4428 014446 000200
4429
4430
4431
4432
4433 014450 011646
4434 014452 016666 000004 000002
4435 014460 000002
4436
4437
4438
4439
4440
4441
4442
4443
4444 014462 014450
4445 014464 011770
4446 014466 014224
4447 014470 014200
4448 014472 014240
4449 014474 013754
4450
4451 014476 012322
4452
4453 014500 012252
4454 014502 012534
4455 014504 012654
4456 000001

```

.SBTTL TRAP DECODER

```

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

```

```

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

```

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

```

STRAP2: 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  $STRAP2      TRAP+1(104401)  TTY TYPEOUT ROUTINE
        $TYPE   ;;CALL=TYPE  TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOC  ;;CALL=TYPOC TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPOS  ;;CALL=TYPOS  TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPON  ;;CALL=TYPON  TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
        $TYPDS  ;;CALL=TYPDS
        $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









|                |       |       |       |       |       |       |       |       |       |       |       |       |       |  |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| PR7 = 000340   | 667#  | 1143  | 1369  | 2512  | 2646  | 2769  | 3144  | 3150  |       |       |       |       |       |  |
| PS = 177776    | 640#  | 641   |       |       |       |       |       |       |       |       |       |       |       |  |
| PSM = 177776   | 641#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PMRVEC= 000024 | 732#  | 1031# | 1032* | 3836* | 3837* | 3846* | 3852* | 3864* | 3865* |       |       |       |       |  |
| RATES 005676   | 2350  | 2431  | 2447# |       |       |       |       |       |       |       |       |       |       |  |
| RBUF 001262    | 1009# | 1112# | 1113# | 1931  | 2075  | 2164  | 2213  | 2245  | 2263  | 2347  | 2899  | 3003  | 3060  |  |
|                | 3249  | 3347  | 3359  |       |       |       |       |       |       |       |       |       |       |  |
| RCSR 001260    | 1008# | 1110# | 1457  | 1470* | 1473  | 1487* | 1490  | 1504* | 1509  | 1540  | 1605  | 1831  | 1859  |  |
|                | 1905  | 1934  | 1986  | 2026  | 2030  | 2033  | 2056  | 2078  | 2367  | 2658* | 2682* | 2736* | 2773* |  |
|                | 2886  | 2990# | 3042  | 3087* | 3176* | 3286* | 3343  |       |       |       |       |       |       |  |
| RCVRAC= 004000 | 761#  | 1605  | 1986  | 2026  | 2033  |       |       |       |       |       |       |       |       |  |
| RCVRD0= 000200 | 765#  | 1540  | 1832  | 1859  | 1906  | 1934  | 2030  | 2056  | 2078  | 2367  | 2887  | 3043  | 3343  |  |
| RCVRIE= 000100 | 766#  | 1457  | 1470  | 1473  | 1487  | 1490  | 1504  | 1509  | 2658  | 2682  | 2736  | 2773  | 3176  |  |
|                | 3286  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA0= 000001 | 792#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA1= 000002 | 791#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA2= 000004 | 790#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA3= 000010 | 789#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA4= 000020 | 788#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA5= 000040 | 787#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA6= 000100 | 786#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RDATA7= 000200 | 785#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RCHR = 104410  | 4069  | 4454# |       |       |       |       |       |       |       |       |       |       |       |  |
| ROLIN = 104411 | 4455# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RORRUM= 000001 | 772#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| REC 010040     | 3142  | 3245# |       |       |       |       |       |       |       |       |       |       |       |  |
| REG = 000004   | 3424# | 3448  |       |       |       |       |       |       |       |       |       |       |       |  |
| REGSAV 010626  | 3448* | 3461  | 3517# |       |       |       |       |       |       |       |       |       |       |  |
| RESVEC= 000010 | 727#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RHLD 010130    | 3253* | 3254* | 3258  | 3266  | 3295# |       |       |       |       |       |       |       |       |  |
| RO050 005676   | 2456# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO070 005677   | 2457# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO110 005700   | 2458# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO135 005701   | 2459# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO150 005702   | 2460# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO200 005705   | 2463# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO300 005703   | 2461# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RO600 005704   | 2462# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R10000 005715  | 2471# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R1800 005706   | 2464# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R2000 005707   | 2465# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R2400 005710   | 2466# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R3600 005711   | 2467# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R4800 005712   | 2468# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| RSSTAC 001334  | 1015# | 1124  |       |       |       |       |       |       |       |       |       |       |       |  |
| R7200 005713   | 2469# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| R9600 005714   | 2470# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| S8 007774      | 3205# | 3266* |       |       |       |       |       |       |       |       |       |       |       |  |
| SET = 177777   | 750#  | 1703  | 1734  | 1745  | 1761  | 1770  | 1777  | 1781  | 1830  | 1904  | 1989  | 1998  | 2519  |  |
|                | 2674  | 2866  | 2885  | 3020  | 3041  | 3469  |       |       |       |       |       |       |       |  |
| SETCLR= 000006 | 3424# | 3475  |       |       |       |       |       |       |       |       |       |       |       |  |
| SPECIA 011074  | 3670# | 3827  |       |       |       |       |       |       |       |       |       |       |       |  |
| STACK = 001100 | 631#  | 1023  |       |       |       |       |       |       |       |       |       |       |       |  |
| START 001336   | 864   | 1016# |       |       |       |       |       |       |       |       |       |       |       |  |
| STKLMT= 177774 | 642#  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| SWR 001140     | 925#  | 1021  | 1043* | 1045  | 1051* | 1058* | 1072  | 3844  | 3857* | 3966  | 4003* | 4171  | 4178  |  |













|         |        |       |       |       |       |       |       |       |       |       |       |       |       |      |
|---------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
|         | 3510#  | 3511  | 3514# | 3516  | 3545# | 3546  | 3547# | 3551  | 3553# | 3554  | 3556  | 3560  | 3561# |      |
|         | 3562   | 3563# | 3564  | 3568# | 3569  | 3591# | 3592  | 3593# | 3594  | 3596  | 3600  | 3601# | 3602  |      |
|         | 3604   | 3608  | 3634# | 3635  | 3651# | 3652  | 3653# | 3654  | 3663# | 3664  | 3668# | 3672# | 3673  |      |
|         | 3676#  | 3677  | 3678# | 3679  | 3680# | 3681  | 3682# | 3683  | 3687# | 3688  | 3689# | 3690  | 3691# |      |
|         | 3692   | 3693# | 3694  | 3703# | 3704  | 3705# | 3706  | 3707# | 3708  | 3719# | 3720  | 3728# | 3729  |      |
|         | 3730#  | 3731  |       |       |       |       |       |       |       |       |       |       |       |      |
| SERROR  | 013274 | 1027  | 4175# | 4176# | 4177  | 4206  |       |       |       |       |       |       |       |      |
| SERRPC  | 001116 | 915#  | 3774  |       |       |       |       |       |       |       |       |       |       |      |
| SERRTB  | 001254 | 1004# |       |       |       |       |       |       |       |       |       |       |       |      |
| SERTTL  | 001112 | 912#  | 3414  | 3416# | 4174# | 4206  |       |       |       |       |       |       |       |      |
| SESCAP  | 001162 | 936#  | 1035# | 4197  | 4199  | 4206  | 4264# |       |       |       |       |       |       |      |
| SETABL  | 001214 | 955#  |       |       |       |       |       |       |       |       |       |       |       |      |
| SETEND  | 001254 | 897   | 988#  |       |       |       |       |       |       |       |       |       |       |      |
| SFATAL  | 001176 | 948#  | 3765# | 3766  | 4135# |       |       |       |       |       |       |       |       |      |
| SFFLG   | 013272 | 4098# | 4101# | 4129  | 4138# | 4146# |       |       |       |       |       |       |       |      |
| SFILLC  | 001156 | 933#  | 3925  | 3956  |       |       |       |       |       |       |       |       |       |      |
| SFILLS  | 001155 | 932#  | 3956  |       |       |       |       |       |       |       |       |       |       |      |
| SFSAND= | 000310 | 1#    | 1085  | 1160  | 1201  | 1214  | 1230  | 1247  | 1266  | 1286  | 1299  | 1315  | 1332  | 1351 |
|         |        | 1381  | 1397  | 1414  | 1433  | 1459  | 1475  | 1492  | 1511  | 1542  | 1573  | 1580  | 1587  | 1607 |
|         |        | 1645  | 1675  | 1732  | 1763  | 1766  | 1805  | 1861  | 1879  | 1936  | 1960  | 1988  | 2006  | 2028 |
|         |        | 2032  | 2035  | 2058  | 2080  | 2108  | 2113  | 2170  | 2191  | 2215  | 2247  | 2265  | 2306  | 2311 |
|         |        | 2366  | 2369  | 2390  | 2398  | 2439  | 2551  | 2555  | 2593  | 2633  | 2694  | 2701  | 2749  | 2786 |
|         |        | 2824  | 2916  | 2935  | 2942  | 2960  | 2970  | 3073  | 3108  | 3193  | 3228  | 3260  | 3264  | 3284 |
|         |        | 3318  | 3349  | 3358  | 3361  | 3376  | 3383  | 3463  | 3477  | 3509  | 3659  | 3662  | 3702  | 3714 |
|         |        | 3717  |       |       |       |       |       |       |       |       |       |       |       |      |
| SFSBAD= | 000401 | 1#    | 1085  | 1160  | 1201  | 1214  | 1230  | 1247  | 1266  | 1286  | 1299  | 1315  | 1332  | 1351 |
|         |        | 1381  | 1397  | 1414  | 1433  | 1459  | 1475  | 1492  | 1511  | 1542  | 1573  | 1580  | 1587  | 1607 |
|         |        | 1645  | 1675  | 1732  | 1763  | 1766  | 1808  | 1861  | 1882  | 1936  | 1962  | 1988  | 2006  | 2028 |
|         |        | 2032  | 2035  | 2058  | 2080  | 2110  | 2113  | 2170  | 2191  | 2215  | 2247  | 2265  | 2308  | 2311 |
|         |        | 2366  | 2369  | 2390  | 2400  | 2439  | 2551  | 2555  | 2593  | 2636  | 2694  | 2701  | 2749  | 2786 |
|         |        | 2827  | 2916  | 2935  | 2942  | 2963  | 2970  | 3073  | 3111  | 3193  | 3228  | 3260  | 3264  | 3284 |
|         |        | 3321  | 3349  | 3358  | 3361  | 3376  | 3383  | 3463  | 3477  | 3509  | 3659  | 3662  | 3702  | 3714 |
|         |        | 3717  |       |       |       |       |       |       |       |       |       |       |       |      |
| SFSBLA= | 000170 | 1#    |       |       |       |       |       |       |       |       |       |       |       |      |
| SFSCAS= | 000150 | 1#    |       |       |       |       |       |       |       |       |       |       |       |      |
| SFSDEC= | 000220 | 1#    |       |       |       |       |       |       |       |       |       |       |       |      |
| SFSG00= | 000400 | 1#    | 1082  | 1083  | 1158  | 1199  | 1212  | 1228  | 1245  | 1264  | 1284  | 1297  | 1313  | 1330 |
|         |        | 1349  | 1379  | 1395  | 1412  | 1431  | 1457  | 1473  | 1490  | 1509  | 1540  | 1571  | 1578  | 1585 |
|         |        | 1605  | 1643  | 1673  | 1712  | 1730  | 1752  | 1761  | 1764  | 1803  | 1839  | 1859  | 1877  | 1912 |
|         |        | 1934  | 1958  | 1986  | 2004  | 2025  | 2026  | 2030  | 2033  | 2056  | 2078  | 2106  | 2111  | 2168 |
|         |        | 2189  | 2213  | 2245  | 2263  | 2304  | 2309  | 2363  | 2364  | 2367  | 2388  | 2396  | 2437  | 2549 |
|         |        | 2553  | 2591  | 2631  | 2692  | 2699  | 2747  | 2784  | 2822  | 2873  | 2892  | 2914  | 2933  | 2940 |
|         |        | 2958  | 2968  | 3027  | 3048  | 3071  | 3106  | 3191  | 3226  | 3258  | 3262  | 3282  | 3316  | 3347 |
|         |        | 3356  | 3359  | 3374  | 3381  | 3461  | 3475  | 3507  | 3657  | 3660  | 3700  | 3712  | 3715  |      |
| SFSIF = | 000110 | 1#    | 1158  | 1164  | 1199  | 1205  | 1212  | 1218  | 1228  | 1234  | 1245  | 1251  | 1264  | 1270 |
|         |        | 1284  | 1290  | 1297  | 1303  | 1313  | 1319  | 1330  | 1336  | 1349  | 1355  | 1379  | 1385  | 1395 |
|         |        | 1401  | 1412  | 1418  | 1431  | 1437  | 1457  | 1463  | 1473  | 1479  | 1490  | 1496  | 1509  | 1515 |
|         |        | 1540  | 1551  | 1571  | 1575  | 1578  | 1582  | 1585  | 1589  | 1595  | 1597  | 1599  | 1605  | 1622 |
|         |        | 1643  | 1654  | 1673  | 1679  | 1712  | 1717  | 1730  | 1738  | 1752  | 1757  | 1759  | 1761  | 1764 |
|         |        | 1772  | 1774  | 1779  | 1803  | 1805  | 1812  | 1839  | 1847  | 1859  | 1865  | 1877  | 1879  | 1886 |
|         |        | 1912  | 1923  | 1934  | 1946  | 1958  | 1960  | 1964  | 1970  | 1986  | 1991  | 1996  | 2004  | 2017 |
|         |        | 2030  | 2033  | 2047  | 2049  | 2056  | 2068  | 2078  | 2087  | 2106  | 2108  | 2111  | 2115  | 2121 |
|         |        | 2123  | 2129  | 2168  | 2182  | 2189  | 2205  | 2213  | 2227  | 2245  | 2260  | 2263  | 2273  | 2304 |
|         |        | 2306  | 2309  | 2313  | 2319  | 2321  | 2327  | 2367  | 2373  | 2381  | 2388  | 2392  | 2396  | 2398 |
|         |        | 2402  | 2411  | 2413  | 2437  | 2443  | 2549  | 2553  | 2559  | 2566  | 2568  | 2591  | 2597  | 2631 |
|         |        | 2633  | 2638  | 2692  | 2699  | 2706  | 2712  | 2714  | 2747  | 2754  | 2784  | 2790  | 2822  | 2824 |

|                |       |       |       |       |       |       |       |       |       |       |       |       |       |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | 2831  | 2873  | 2877  | 2892  | 2896  | 2914  | 2924  | 2933  | 2938  | 2940  | 2945  | 2958  | 2960  |
|                | 2968  | 2974  | 2976  | 3027  | 3033  | 3048  | 3056  | 3071  | 3078  | 3106  | 3108  | 3115  | 3191  |
|                | 3198  | 3226  | 3232  | 3258  | 3262  | 3270  | 3275  | 3282  | 3290  | 3316  | 3318  | 3325  | 3347  |
|                | 3353  | 3356  | 3359  | 3364  | 3366  | 3368  | 3374  | 3379  | 3381  | 3386  | 3461  | 3466  | 3471  |
|                | 3475  | 3481  | 3507  | 3512  | 3657  | 3660  | 3665  | 3674  | 3684  | 3695  | 3700  | 3712  | 3715  |
|                | 3721  | 3723  | 3725  |       |       |       |       |       |       |       |       |       |       |
| SFSINC= 000210 | 1#    | 2339  | 2421  | 2853  | 2926  | 3006  | 3081  | 3553  | 3565  | 3593  | 3601  | 3609  | 3612  |
| SFSL00= 000200 | 1#    | 1684  | 1781  | 1784  | 3458  | 3484  | 3499  |       |       |       |       |       |       |
| SFSNAM= 000160 | 1#    |       |       |       |       |       |       |       |       |       |       |       |       |
| SFSNO = 000403 | 1#    | 1083  | 1158  | 1199  | 1212  | 1228  | 1245  | 1264  | 1284  | 1297  | 1313  | 1330  | 1349  |
|                | 1379  | 1395  | 1412  | 1431  | 1457  | 1473  | 1490  | 1509  | 1540  | 1571  | 1578  | 1585  | 1605  |
|                | 1643  | 1730  | 1761  | 1764  | 1803  | 1859  | 1877  | 1934  | 1960  | 1986  | 2004  | 2026  | 2030  |
|                | 2033  | 2056  | 2078  | 2106  | 2111  | 2168  | 2189  | 2213  | 2245  | 2263  | 2304  | 2306  | 2364  |
|                | 2367  | 2388  | 2396  | 2398  | 2437  | 2549  | 2553  | 2591  | 2631  | 2692  | 2699  | 2747  | 2784  |
|                | 2822  | 2914  | 2933  | 2940  | 2958  | 2960  | 3071  | 3106  | 3191  | 3226  | 3258  | 3262  | 3282  |
| SFSOR = 000320 | 1#    | 3316  | 3318  | 3356  | 3359  | 3374  | 3381  | 3461  | 3657  | 3660  | 3700  | 3712  | 3715  |
|                | 1381  | 1397  | 1414  | 1433  | 1459  | 1475  | 1492  | 1511  | 1542  | 1573  | 1580  | 1587  | 1607  |
|                | 1645  | 1675  | 1732  | 1763  | 1766  | 1805  | 1861  | 1879  | 1936  | 1960  | 1962  | 1988  | 2006  |
|                | 2028  | 2032  | 2035  | 2058  | 2080  | 2108  | 2110  | 2113  | 2170  | 2191  | 2215  | 2247  | 2265  |
|                | 2306  | 2308  | 2311  | 2366  | 2369  | 2390  | 2398  | 2400  | 2439  | 2551  | 2555  | 2593  | 2633  |
|                | 2694  | 2701  | 2749  | 2786  | 2824  | 2916  | 2935  | 2942  | 2960  | 2970  | 3073  | 3108  | 3193  |
|                | 3228  | 3260  | 3264  | 3284  | 3318  | 3349  | 3358  | 3361  | 3376  | 3383  | 3463  | 3477  | 3509  |
|                | 3659  | 3662  | 3702  | 3714  | 3717  |       |       |       |       |       |       |       |       |
| SFSRTN= 000300 | 1#    | 3424  | 3523  | 3531  | 3573  | 3580  | 3618  | 3640  | 3746  | 3754  | 3790  |       |       |
| SFSSEL= 000140 | 1#    |       |       |       |       |       |       |       |       |       |       |       |       |
| SFSUNT= 000130 | 1#    | 1148  | 1172  | 1984  | 1998  | 3182  | 3184  | 3340  | 3343  | 3655  | 3697  |       |       |
| SFSWHI= 000120 | 1#    | 1082  | 1083  | 1090  | 1805  | 1879  | 1960  | 2025  | 2026  | 2051  | 2108  | 2306  | 2363  |
|                | 2364  | 2384  | 2398  | 2633  | 2824  | 2960  | 3108  | 3318  |       |       |       |       |       |
| SFSYES= 000402 | 1#    | 1083  | 1158  | 1199  | 1212  | 1228  | 1245  | 1264  | 1284  | 1297  | 1313  | 1330  | 1349  |
|                | 1379  | 1395  | 1412  | 1431  | 1457  | 1473  | 1490  | 1509  | 1540  | 1571  | 1578  | 1585  | 1605  |
|                | 1643  | 1673  | 1730  | 1761  | 1764  | 1803  | 1805  | 1859  | 1877  | 1879  | 1934  | 1958  | 1960  |
|                | 1986  | 2004  | 2026  | 2030  | 2033  | 2056  | 2078  | 2106  | 2108  | 2111  | 2168  | 2189  | 2213  |
|                | 2245  | 2263  | 2304  | 2306  | 2309  | 2364  | 2367  | 2388  | 2396  | 2398  | 2437  | 2549  | 2553  |
|                | 2591  | 2631  | 2633  | 2692  | 2699  | 2747  | 2784  | 2822  | 2824  | 2914  | 2933  | 2940  | 2958  |
|                | 2960  | 2968  | 3071  | 3106  | 3108  | 3191  | 3226  | 3258  | 3262  | 3282  | 3316  | 3318  | 3347  |
|                | 3356  | 3359  | 3374  | 3381  | 3461  | 3475  | 3507  | 3657  | 3660  | 3700  | 3712  | 3715  |       |
| SGDAOR 001120  | 916#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SGDOAT 001124  | 918#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SGET42 011542  | 3818# |       |       |       |       |       |       |       |       |       |       |       |       |
| SGTSMR 012322  | 3978# | 4451  |       |       |       |       |       |       |       |       |       |       |       |
| SHD = 000000   | 616   |       |       |       |       |       |       |       |       |       |       |       |       |
| SHIBTS 001000  | 892#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SICNT 001104   | 909#  | 4255# | 4256  | 4258# | 4269  |       |       |       |       |       |       |       |       |
| SIFLEV= 177777 | 1#    | 1158# | 1164# | 1199# | 1205# | 1212# | 1218# | 1228# | 1234# | 1245# | 1251# | 1264# | 1270# |
|                | 1284# | 1290# | 1297# | 1303# | 1313# | 1319# | 1330# | 1336# | 1349# | 1355# | 1379# | 1385# | 1395# |
|                | 1401# | 1412# | 1418# | 1431# | 1437# | 1457# | 1463# | 1473# | 1479# | 1490# | 1496# | 1509# | 1515# |
|                | 1540# | 1551# | 1571# | 1578# | 1585# | 1595# | 1597# | 1599# | 1605# | 1622# | 1643# | 1654# | 1673# |
|                | 1679# | 1712# | 1717# | 1730# | 1752# | 1757# | 1759# | 1761# | 1764# | 1772# | 1779# | 1803# | 1812# |
|                | 1839# | 1847# | 1859# | 1865# | 1877# | 1886# | 1912# | 1923# | 1934# | 1946# | 1958# | 1970# | 1986# |
|                | 1996# | 2004# | 2017# | 2030# | 2033# | 2047# | 2049# | 2056# | 2068# | 2078# | 2087# | 2106# | 2111# |
|                | 2121# | 2129# | 2168# | 2182# | 2189# | 2205# | 2213# | 2227# | 2245# | 2260# | 2263# | 2273# | 2304# |
|                | 2309# | 2319# | 2327# | 2367# | 2381# | 2388# | 2396# | 2411# | 2413# | 2437# | 2443# | 2549# | 2553# |
|                | 2566# | 2568# | 2591# | 2597# | 2631# | 2638# | 2692# | 2699# | 2712# | 2714# | 2747# | 2754# | 2784# |
|                | 2790# | 2822# | 2831# | 2873# | 2877# | 2892# | 2896# | 2914# | 2924# | 2933# | 2938# | 2940# | 2945# |
|                | 2958# | 2968# | 2974# | 2976# | 3027# | 3033# | 3048# | 3056# | 3071# | 3078# | 3106# | 3115# | 3191# |



SILLUP 011752  
SINTAG 001135  
SISKO = 000001

SISK1 = 000001

SISK2 = 000001  
SITEMB 001114  
SLF 001172  
SLFLG 013271  
SLOCTA= 177777

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 3198# | 3226# | 3232# | 3258# | 3262# | 3270# | 3275# | 3282# | 3290# | 3316# | 3325# | 3347# | 3356# |
| 3359# | 3364# | 3366# | 3368# | 3374# | 3379# | 3381# | 3386# | 3461# | 3471# | 3475# | 3481# | 3507# |
| 3512# | 3557# | 3660# | 3674# | 3695# | 3700# | 3712# | 3715# | 3721# | 3723# | 3725# |       |       |
| 3836# | 3852# | 3871# |       |       |       |       |       |       |       |       |       |       |
| 923#  | 4006# | 4095# |       |       |       |       |       |       |       |       |       |       |
| 1158# | 1164# | 1199# | 1205  | 1212# | 1218  | 1228# | 1234  | 1245# | 1251  | 1264# | 1270  | 1284# |
| 1290  | 1297# | 1303  | 1313# | 1319  | 1330# | 1336  | 1349# | 1355  | 1379# | 1385  | 1395# | 1401  |
| 1412# | 1418  | 1431# | 1437  | 1457# | 1463  | 1473# | 1479  | 1490# | 1496  | 1509# | 1515  | 1540# |
| 1551  | 1571# | 1599  | 1605# | 1622  | 1643# | 1654  | 1673# | 1679  | 1712# | 1717  | 1730# | 1759  |
| 1761# | 1779  | 1803# | 1812  | 1839# | 1847  | 1859# | 1865  | 1877# | 1886  | 1912# | 1923  | 1934# |
| 1946  | 1958# | 1970  | 1986# | 1996  | 2004# | 2017  | 2030# | 2049  | 2056# | 2068  | 2078# | 2087  |
| 2106# | 2129  | 2168# | 2182  | 2189# | 2205  | 2213# | 2227  | 2245# | 2260  | 2263# | 2273  | 2304# |
| 2327  | 2367# | 2381  | 2388# | 2413  | 2437# | 2443  | 2549# | 2568  | 2591# | 2597  | 2631# | 2638  |
| 2692# | 2714  | 2747# | 2754  | 2784# | 2790  | 2822# | 2831  | 2873# | 2877  | 2892# | 2896  | 2914# |
| 2924  | 2933# | 2938  | 2940# | 2945  | 2958# | 2976  | 3027# | 3033  | 3048# | 3056  | 3071# | 3078  |
| 3106# | 3115  | 3191# | 3198  | 3226# | 3232  | 3258# | 3275  | 3282# | 3290  | 3316# | 3325  | 3347# |
| 3368  | 3374# | 3379  | 3381# | 3386  | 3461# | 3471  | 3475# | 3481  | 3507# | 3512  | 3657# | 3695  |
| 3700# | 3725  |       |       |       |       |       |       |       |       |       |       |       |
| 1578# | 1597  | 1752# | 1757  | 1764# | 1772  | 2033# | 2047  | 2111# | 2121  | 2309# | 2319  | 2396# |
| 2411  | 2553# | 2566  | 2699# | 2712  | 2968# | 2974  | 3262# | 3270  | 3356# | 3366  | 3660# | 3674  |
| 3712# | 3723  |       |       |       |       |       |       |       |       |       |       |       |
| 1585# | 1595  | 3359# | 3364  | 3715# | 3721  |       |       |       |       |       |       |       |
| 913#  | 3765  | 4177# | 4185  | 4206  |       |       |       |       |       |       |       |       |
| 940#  | 3956  | 4080  | 4089  | 4206  |       |       |       |       |       |       |       |       |
| 4139# | 4145# |       |       |       |       |       |       |       |       |       |       |       |
| 1#    | 1082  | 1083  | 1084  | 1085  | 1090  | 1091  | 1092  | 1148  | 1149  | 1159  | 1160  | 1164  |
| 1165  | 1173  | 1174  | 1200  | 1201  | 1205  | 1206  | 1213  | 1214  | 1218  | 1219  | 1229  | 1230  |
| 1234  | 1235  | 1246  | 1247  | 1251  | 1252  | 1265  | 1266  | 1270  | 1271  | 1285  | 1286  | 1290  |
| 1291  | 1298  | 1299  | 1303  | 1304  | 1314  | 1315  | 1319  | 1320  | 1331  | 1332  | 1336  | 1337  |
| 1350  | 1351  | 1355  | 1356  | 1380  | 1381  | 1385  | 1386  | 1396  | 1397  | 1401  | 1402  | 1413  |
| 1414  | 1418  | 1419  | 1432  | 1433  | 1437  | 1438  | 1458  | 1459  | 1463  | 1464  | 1474  | 1475  |
| 1479  | 1480  | 1491  | 1492  | 1496  | 1497  | 1510  | 1511  | 1515  | 1516  | 1541  | 1542  | 1551  |
| 1552  | 1572  | 1573  | 1575  | 1576  | 1577  | 1579  | 1580  | 1582  | 1583  | 1584  | 1586  | 1587  |
| 1589  | 1590  | 1591  | 1595  | 1596  | 1597  | 1598  | 1599  | 1600  | 1606  | 1607  | 1622  | 1623  |
| 1644  | 1645  | 1654  | 1655  | 1674  | 1675  | 1679  | 1680  | 1684  | 1685  | 1712  | 1713  | 1717  |
| 1718  | 1731  | 1732  | 1738  | 1739  | 1740  | 1752  | 1753  | 1757  | 1758  | 1759  | 1760  | 1762  |
| 1763  | 1765  | 1766  | 1772  | 1773  | 1774  | 1775  | 1776  | 1779  | 1780  | 1782  | 1783  | 1784  |
| 1785  | 1786  | 1804  | 1805  | 1806  | 1807  | 1808  | 1812  | 1813  | 1839  | 1840  | 1847  | 1848  |
| 1860  | 1861  | 1865  | 1866  | 1878  | 1879  | 1880  | 1881  | 1882  | 1886  | 1887  | 1912  | 1913  |
| 1923  | 1924  | 1935  | 1936  | 1946  | 1947  | 1959  | 1960  | 1961  | 1962  | 1964  | 1965  | 1966  |
| 1970  | 1971  | 1984  | 1985  | 1987  | 1988  | 1991  | 1992  | 1993  | 1996  | 1997  | 1999  | 2000  |
| 2001  | 2002  | 2003  | 2005  | 2006  | 2017  | 2018  | 2025  | 2026  | 2027  | 2028  | 2031  | 2032  |
| 2034  | 2035  | 2047  | 2048  | 2049  | 2050  | 2051  | 2052  | 2053  | 2057  | 2058  | 2068  | 2069  |
| 2079  | 2080  | 2087  | 2088  | 2107  | 2108  | 2109  | 2110  | 2112  | 2113  | 2115  | 2116  | 2117  |
| 2121  | 2122  | 2123  | 2124  | 2125  | 2129  | 2130  | 2169  | 2170  | 2182  | 2183  | 2190  | 2191  |
| 2205  | 2206  | 2214  | 2215  | 2227  | 2228  | 2246  | 2247  | 2260  | 2261  | 2264  | 2265  | 2273  |
| 2274  | 2305  | 2306  | 2307  | 2308  | 2310  | 2311  | 2313  | 2314  | 2315  | 2319  | 2320  | 2321  |
| 2322  | 2323  | 2327  | 2328  | 2340  | 2341  | 2342  | 2343  | 2344  | 2345  | 2346  | 2363  | 2364  |
| 2365  | 2366  | 2368  | 2369  | 2373  | 2374  | 2375  | 2381  | 2382  | 2384  | 2385  | 2386  | 2389  |
| 2390  | 2392  | 2393  | 2394  | 2397  | 2398  | 2399  | 2400  | 2402  | 2403  | 2404  | 2411  | 2412  |
| 2413  | 2414  | 2421  | 2422  | 2423  | 2438  | 2439  | 2443  | 2444  | 2550  | 2551  | 2554  | 2555  |
| 2559  | 2560  | 2561  | 2566  | 2567  | 2568  | 2569  | 2592  | 2593  | 2597  | 2598  | 2632  | 2633  |
| 2634  | 2635  | 2636  | 2638  | 2639  | 2693  | 2694  | 2700  | 2701  | 2706  | 2707  | 2708  | 2712  |
| 2713  | 2714  | 2715  | 2748  | 2749  | 2754  | 2755  | 2785  | 2786  | 2790  | 2791  | 2823  | 2824  |
| 2825  | 2826  | 2827  | 2831  | 2832  | 2854  | 2855  | 2856  | 2857  | 2858  | 2859  | 2860  | 2873  |
| 2874  | 2877  | 2878  | 2892  | 2893  | 2896  | 2897  | 2915  | 2916  | 2924  | 2925  | 2926  | 2927  |



K09

MAINDEC-11-DVDVC-A  
DVDVCA.P11

MACY11 27(1006)  
08-AUG-77 09:16

08-AUG-77 09:20 PAGE 116  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0114

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 2928  | 2934  | 2935  | 2938  | 2939  | 2941  | 2942  | 2945  | 2946  | 2959  | 2960  | 2961  | 2962  |
| 2963  | 2969  | 2970  | 2974  | 2975  | 2976  | 2977  | 3007  | 3008  | 3009  | 3010  | 3011  | 3012  |
| 3013  | 3027  | 3028  | 3033  | 3034  | 3048  | 3049  | 3056  | 3057  | 3072  | 3073  | 3078  | 3079  |
| 3081  | 3082  | 3083  | 3107  | 3108  | 3109  | 3110  | 3111  | 3115  | 3116  | 3182  | 3183  | 3185  |
| 3186  | 3192  | 3193  | 3198  | 3199  | 3227  | 3228  | 3232  | 3233  | 3259  | 3260  | 3263  | 3264  |
| 3270  | 3271  | 3275  | 3276  | 3283  | 3284  | 3290  | 3291  | 3317  | 3318  | 3319  | 3320  | 3321  |
| 3325  | 3326  | 3340  | 3341  | 3344  | 3345  | 3348  | 3349  | 3353  | 3354  | 3355  | 3357  | 3358  |
| 3360  | 3361  | 3364  | 3365  | 3366  | 3367  | 3368  | 3369  | 3375  | 3376  | 3379  | 3380  | 3382  |
| 3383  | 3386  | 3387  | 3423  | 3458  | 3459  | 3462  | 3463  | 3466  | 3467  | 3468  | 3471  | 3472  |
| 3476  | 3477  | 3481  | 3482  | 3485  | 3486  | 3487  | 3488  | 3499  | 3500  | 3501  | 3508  | 3509  |
| 3510  | 3511  | 3512  | 3513  | 3515  | 3516  | 3523  | 3524  | 3525  | 3526  | 3530  | 3554  | 3555  |
| 3556  | 3557  | 3558  | 3559  | 3560  | 3565  | 3566  | 3567  | 3570  | 3571  | 3573  | 3574  | 3575  |
| 3579  | 3594  | 3595  | 3596  | 3597  | 3598  | 3599  | 3600  | 3602  | 3603  | 3604  | 3605  | 3606  |
| 3607  | 3608  | 3609  | 3610  | 3611  | 3612  | 3613  | 3614  | 3618  | 3619  | 3620  | 3639  | 3655  |
| 3656  | 3658  | 3659  | 3661  | 3662  | 3665  | 3666  | 3667  | 3674  | 3675  | 3684  | 3685  | 3686  |
| 3695  | 3696  | 3698  | 3699  | 3701  | 3702  | 3713  | 3714  | 3716  | 3717  | 3721  | 3722  | 3723  |
| 3724  | 3725  | 3726  | 3732  | 3733  | 3746  | 3747  | 3748  | 3753  | 3790  | 3791  | 3792  |       |
| 910#  | 1037* | 4246* | 4262* | 4267  | 4269  |       |       |       |       |       |       |       |
| 911#  | 1038* | 1150* | 1209* | 1223* | 1239* | 1256* | 1294* | 1308* | 1324* | 1341* | 1376* | 1390* |
| 1406* | 1423* | 1454* | 1468* | 1484* | 1501* | 1538* | 1602* | 1640* | 1819* | 1852* | 1892* | 2137* |
| 2187* | 2209* | 2231* | 2515* | 2572* | 2650* | 2727* | 2762* | 4196  | 4246  | 4263* | 4269  |       |
| 1#    | 1082  | 1083  | 1085  | 1090  | 1091  | 1148  | 1149  | 1158  | 1160  | 1164  | 1172  | 1199  |
| 1201  | 1205  | 1212  | 1214  | 1218  | 1228  | 1230  | 1234  | 1245  | 1247  | 1251  | 1264  | 1266  |
| 1270  | 1284  | 1286  | 1290  | 1297  | 1299  | 1303  | 1313  | 1315  | 1319  | 1330  | 1332  | 1336  |
| 1349  | 1351  | 1355  | 1379  | 1381  | 1385  | 1395  | 1397  | 1401  | 1412  | 1414  | 1418  | 1431  |
| 1433  | 1437  | 1457  | 1459  | 1463  | 1473  | 1475  | 1479  | 1490  | 1492  | 1496  | 1509  | 1511  |
| 1515  | 1540  | 1542  | 1551  | 1571  | 1573  | 1576  | 1577  | 1578  | 1580  | 1583  | 1584  | 1585  |
| 1587  | 1590  | 1591  | 1595  | 1597  | 1599  | 1605  | 1607  | 1622  | 1643  | 1645  | 1654  | 1673  |
| 1675  | 1679  | 1684  | 1685  | 1712  | 1713  | 1717  | 1730  | 1732  | 1739  | 1740  | 1752  | 1753  |
| 1757  | 1759  | 1761  | 1763  | 1764  | 1766  | 1772  | 1775  | 1776  | 1779  | 1784  | 1785  | 1803  |
| 1808  | 1812  | 1839  | 1840  | 1847  | 1859  | 1861  | 1865  | 1877  | 1882  | 1886  | 1912  | 1913  |
| 1923  | 1934  | 1936  | 1946  | 1958  | 1962  | 1965  | 1966  | 1970  | 1984  | 1985  | 1986  | 1988  |
| 1992  | 1993  | 1996  | 1998  | 2003  | 2004  | 2006  | 2017  | 2025  | 2026  | 2028  | 2030  | 2032  |
| 2033  | 2035  | 2047  | 2049  | 2051  | 2052  | 2056  | 2058  | 2068  | 2078  | 2080  | 2087  | 2106  |
| 2110  | 2111  | 2113  | 2116  | 2117  | 2121  | 2124  | 2125  | 2129  | 2168  | 2170  | 2182  | 2189  |
| 2191  | 2205  | 2213  | 2215  | 2227  | 2245  | 2247  | 2260  | 2263  | 2265  | 2273  | 2304  | 2308  |
| 2309  | 2311  | 2314  | 2315  | 2319  | 2322  | 2323  | 2327  | 2339  | 2341  | 2342  | 2343  | 2346  |
| 2363  | 2364  | 2366  | 2367  | 2369  | 2374  | 2375  | 2381  | 2384  | 2385  | 2388  | 2390  | 2393  |
| 2394  | 2396  | 2400  | 2403  | 2404  | 2411  | 2413  | 2421  | 2422  | 2437  | 2439  | 2443  | 2549  |
| 2551  | 2553  | 2555  | 2560  | 2561  | 2566  | 2568  | 2591  | 2593  | 2597  | 2631  | 2636  | 2638  |
| 2692  | 2694  | 2699  | 2701  | 2707  | 2708  | 2712  | 2714  | 2747  | 2749  | 2754  | 2784  | 2786  |
| 2790  | 2822  | 2827  | 2831  | 2853  | 2855  | 2856  | 2857  | 2860  | 2873  | 2874  | 2877  | 2892  |
| 2893  | 2896  | 2914  | 2916  | 2924  | 2926  | 2927  | 2933  | 2935  | 2938  | 2940  | 2942  | 2945  |
| 2958  | 2963  | 2968  | 2970  | 2974  | 2976  | 3006  | 3008  | 3009  | 3010  | 3013  | 3027  | 3028  |
| 3033  | 3048  | 3049  | 3056  | 3071  | 3073  | 3078  | 3081  | 3082  | 3106  | 3111  | 3115  | 3182  |
| 3183  | 3184  | 3191  | 3193  | 3198  | 3226  | 3228  | 3232  | 3258  | 3260  | 3262  | 3264  | 3270  |
| 3275  | 3282  | 3284  | 3290  | 3316  | 3321  | 3325  | 3340  | 3341  | 3343  | 3347  | 3349  | 3354  |
| 3355  | 3356  | 3358  | 3359  | 3361  | 3364  | 3366  | 3368  | 3374  | 3376  | 3379  | 3381  | 3383  |
| 3386  | 3424  | 3458  | 3459  | 3461  | 3463  | 3467  | 3468  | 3471  | 3475  | 3477  | 3481  | 3499  |
| 3500  | 3507  | 3509  | 3512  | 3523  | 3531  | 3553  | 3555  | 3556  | 3557  | 3560  | 3565  | 3566  |
| 3573  | 3580  | 3593  | 3595  | 3596  | 3597  | 3600  | 3601  | 3603  | 3604  | 3605  | 3608  | 3609  |
| 3610  | 3612  | 3613  | 3618  | 3640  | 3655  | 3656  | 3657  | 3659  | 3660  | 3662  | 3666  | 3667  |
| 3674  | 3685  | 3686  | 3695  | 3697  | 3700  | 3702  | 3712  | 3714  | 3715  | 3717  | 3721  | 3723  |
| 3725  | 3746  | 3754  | 3790  |       |       |       |       |       |       |       |       |       |
| 1#    | 1083  | 1084  | 1085  | 1090  | 1091  | 1093  | 1094  | 1095  | 1096  | 1098  | 1099  | 1102  |
| 1103  | 1105  | 1106  | 1107  | 1108  | 1110  | 1111  | 1112  | 1113  | 1114  | 1115  | 1116  | 1117  |

SLPADR 001106  
SLPERR 001110

SLSTCN= 177777

SLSTIN= 000000

|      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1118 | 1119 | 1120 | 1121 | 1122 | 1123 | 1124 | 1125 | 1137 | 1138 | 1140 | 1141 | 1142 |
| 1143 | 1144 | 1145 | 1146 | 1147 | 1150 | 1151 | 1153 | 1154 | 1158 | 1159 | 1160 | 1167 |
| 1168 | 1169 | 1170 | 1171 | 1172 | 1173 | 1174 | 1177 | 1178 | 1179 | 1180 | 1181 | 1182 |
| 1199 | 1200 | 1201 | 1202 | 1203 | 1209 | 1210 | 1212 | 1213 | 1214 | 1223 | 1224 | 1225 |
| 1226 | 1228 | 1229 | 1230 | 1239 | 1240 | 1242 | 1243 | 1245 | 1246 | 1247 | 1256 | 1257 |
| 1259 | 1260 | 1264 | 1265 | 1266 | 1284 | 1285 | 1286 | 1287 | 1288 | 1294 | 1295 | 1297 |
| 1298 | 1299 | 1308 | 1309 | 1310 | 1311 | 1313 | 1314 | 1315 | 1324 | 1325 | 1327 | 1328 |
| 1330 | 1331 | 1332 | 1341 | 1342 | 1344 | 1345 | 1349 | 1350 | 1351 | 1376 | 1377 | 1379 |
| 1380 | 1381 | 1390 | 1391 | 1392 | 1393 | 1395 | 1396 | 1397 | 1406 | 1407 | 1409 | 1410 |
| 1412 | 1413 | 1414 | 1423 | 1424 | 1426 | 1427 | 1431 | 1432 | 1433 | 1454 | 1455 | 1457 |
| 1458 | 1459 | 1468 | 1469 | 1470 | 1471 | 1473 | 1474 | 1475 | 1484 | 1485 | 1487 | 1488 |
| 1490 | 1491 | 1492 | 1501 | 1502 | 1504 | 1505 | 1509 | 1510 | 1511 | 1538 | 1539 | 1540 |
| 1541 | 1542 | 1571 | 1572 | 1573 | 1575 | 1576 | 1578 | 1579 | 1580 | 1582 | 1583 | 1585 |
| 1586 | 1587 | 1589 | 1590 | 1592 | 1593 | 1602 | 1603 | 1605 | 1606 | 1607 | 1610 | 1611 |
| 1640 | 1641 | 1643 | 1644 | 1645 | 1673 | 1674 | 1675 | 1676 | 1677 | 1682 | 1683 | 1687 |
| 1688 | 1689 | 1690 | 1697 | 1698 | 1702 | 1703 | 1704 | 1705 | 1706 | 1707 | 1708 | 1709 |
| 1712 | 1713 | 1725 | 1726 | 1730 | 1731 | 1732 | 1734 | 1735 | 1738 | 1739 | 1744 | 1745 |
| 1746 | 1747 | 1748 | 1749 | 1750 | 1751 | 1752 | 1753 | 1761 | 1762 | 1763 | 1764 | 1765 |
| 1766 | 1770 | 1771 | 1774 | 1775 | 1777 | 1778 | 1781 | 1782 | 1783 | 1784 | 1785 | 1803 |
| 1804 | 1805 | 1806 | 1807 | 1809 | 1810 | 1816 | 1817 | 1819 | 1820 | 1823 | 1824 | 1829 |
| 1830 | 1831 | 1832 | 1833 | 1834 | 1835 | 1836 | 1839 | 1840 | 1843 | 1844 | 1852 | 1853 |
| 1859 | 1860 | 1861 | 1877 | 1878 | 1879 | 1880 | 1881 | 1883 | 1884 | 1890 | 1891 | 1892 |
| 1893 | 1898 | 1899 | 1903 | 1904 | 1905 | 1906 | 1907 | 1908 | 1909 | 1910 | 1912 | 1913 |
| 1916 | 1917 | 1921 | 1922 | 1931 | 1932 | 1934 | 1935 | 1936 | 1939 | 1940 | 1944 | 1945 |
| 1958 | 1959 | 1960 | 1961 | 1962 | 1964 | 1965 | 1967 | 1968 | 1972 | 1973 | 1974 | 1975 |
| 1976 | 1977 | 1982 | 1983 | 1986 | 1987 | 1988 | 1989 | 1990 | 1991 | 1992 | 1994 | 1995 |
| 1998 | 1999 | 2000 | 2001 | 2002 | 2004 | 2005 | 2006 | 2010 | 2011 | 2014 | 2015 | 2026 |
| 2027 | 2028 | 2030 | 2031 | 2032 | 2033 | 2034 | 2035 | 2039 | 2040 | 2044 | 2045 | 2051 |
| 2052 | 2056 | 2057 | 2058 | 2061 | 2062 | 2065 | 2066 | 2075 | 2076 | 2078 | 2079 | 2080 |
| 2083 | 2084 | 2106 | 2107 | 2108 | 2109 | 2110 | 2111 | 2112 | 2113 | 2115 | 2116 | 2118 |
| 2119 | 2123 | 2124 | 2126 | 2127 | 2132 | 2133 | 2137 | 2138 | 2144 | 2145 | 2147 | 2148 |
| 2149 | 2150 | 2151 | 2154 | 2155 | 2157 | 2158 | 2159 | 2160 | 2161 | 2164 | 2165 | 2168 |
| 2169 | 2170 | 2173 | 2174 | 2179 | 2180 | 2187 | 2188 | 2189 | 2190 | 2191 | 2195 | 2196 |
| 2202 | 2203 | 2209 | 2210 | 2213 | 2214 | 2215 | 2219 | 2220 | 2224 | 2225 | 2231 | 2232 |
| 2236 | 2237 | 2239 | 2240 | 2241 | 2242 | 2243 | 2245 | 2246 | 2247 | 2250 | 2251 | 2257 |
| 2258 | 2263 | 2264 | 2265 | 2268 | 2269 | 2304 | 2305 | 2306 | 2307 | 2308 | 2309 | 2310 |
| 2311 | 2313 | 2314 | 2316 | 2317 | 2321 | 2322 | 2324 | 2325 | 2330 | 2331 | 2332 | 2333 |
| 2334 | 2335 | 2336 | 2337 | 2339 | 2340 | 2341 | 2342 | 2343 | 2344 | 2345 | 2346 | 2347 |
| 2348 | 2350 | 2351 | 2353 | 2354 | 2356 | 2357 | 2359 | 2360 | 2361 | 2362 | 2364 | 2365 |
| 2366 | 2367 | 2368 | 2369 | 2371 | 2372 | 2373 | 2374 | 2377 | 2378 | 2379 | 2380 | 2384 |
| 2385 | 2388 | 2389 | 2390 | 2392 | 2393 | 2396 | 2397 | 2398 | 2399 | 2400 | 2402 | 2403 |
| 2409 | 2410 | 2416 | 2417 | 2418 | 2419 | 2421 | 2422 | 2424 | 2425 | 2426 | 2427 | 2428 |
| 2429 | 2430 | 2431 | 2432 | 2435 | 2436 | 2437 | 2438 | 2439 | 2499 | 2500 | 2503 | 2504 |
| 2506 | 2507 | 2509 | 2510 | 2511 | 2512 | 2513 | 2514 | 2515 | 2516 | 2518 | 2519 | 2520 |
| 2521 | 2522 | 2523 | 2524 | 2525 | 2528 | 2529 | 2538 | 2539 | 2542 | 2543 | 2544 | 2545 |
| 2546 | 2549 | 2550 | 2551 | 2553 | 2554 | 2555 | 2559 | 2560 | 2572 | 2573 | 2575 | 2576 |
| 2578 | 2579 | 2586 | 2587 | 2588 | 2589 | 2590 | 2591 | 2592 | 2593 | 2605 | 2606 | 2607 |
| 2608 | 2609 | 2610 | 2631 | 2632 | 2633 | 2634 | 2635 | 2643 | 2644 | 2645 | 2646 | 2647 |
| 2648 | 2650 | 2651 | 2652 | 2653 | 2655 | 2656 | 2658 | 2659 | 2668 | 2669 | 2673 | 2674 |
| 2675 | 2676 | 2677 | 2678 | 2679 | 2680 | 2682 | 2683 | 2685 | 2686 | 2687 | 2688 | 2689 |
| 2692 | 2693 | 2694 | 2697 | 2698 | 2699 | 2700 | 2701 | 2706 | 2707 | 2717 | 2718 | 2727 |
| 2728 | 2729 | 2730 | 2733 | 2734 | 2736 | 2737 | 2739 | 2740 | 2742 | 2743 | 2744 | 2745 |
| 2746 | 2747 | 2748 | 2749 | 2751 | 2752 | 2756 | 2757 | 2762 | 2763 | 2764 | 2765 | 2766 |
| 2767 | 2773 | 2774 | 2775 | 2776 | 2778 | 2779 | 2780 | 2781 | 2782 | 2784 | 2785 | 2786 |
| 2797 | 2798 | 2801 | 2802 | 2805 | 2806 | 2807 | 2808 | 2809 | 2810 | 2822 | 2823 | 2824 |
| 2825 | 2826 | 2828 | 2829 | 2834 | 2835 | 2837 | 2838 | 2847 | 2848 | 2849 | 2850 | 2853 |



|      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 2854 | 2855 | 2856 | 2857 | 2858 | 2859 | 2860 | 2865 | 2866 | 2867 | 2868 | 2869 | 2870 |
| 2871 | 2872 | 2873 | 2874 | 2875 | 2876 | 2881 | 2882 | 2884 | 2885 | 2886 | 2887 | 2888 |
| 2889 | 2890 | 2891 | 2892 | 2893 | 2894 | 2895 | 2899 | 2900 | 2907 | 2908 | 2909 | 2910 |
| 2911 | 2914 | 2915 | 2916 | 2919 | 2920 | 2926 | 2927 | 2931 | 2932 | 2933 | 2934 | 2935 |
| 2940 | 2941 | 2942 | 2958 | 2959 | 2960 | 2961 | 2962 | 2965 | 2966 | 2968 | 2969 | 2970 |
| 2971 | 2972 | 2979 | 2980 | 2990 | 2991 | 2999 | 3000 | 3001 | 3002 | 3003 | 3004 | 3006 |
| 3007 | 3008 | 3009 | 3010 | 3011 | 3012 | 3013 | 3019 | 3020 | 3021 | 3022 | 3023 | 3024 |
| 3025 | 3026 | 3027 | 3028 | 3030 | 3031 | 3037 | 3038 | 3040 | 3041 | 3042 | 3043 | 3044 |
| 3045 | 3046 | 3047 | 3048 | 3049 | 3053 | 3054 | 3060 | 3061 | 3064 | 3065 | 3066 | 3067 |
| 3068 | 3071 | 3072 | 3073 | 3081 | 3082 | 3087 | 3088 | 3106 | 3107 | 3108 | 3109 | 3110 |
| 3112 | 3113 | 3119 | 3120 | 3121 | 3122 | 3139 | 3140 | 3142 | 3143 | 3144 | 3145 | 3148 |
| 3149 | 3150 | 3151 | 3154 | 3155 | 3157 | 3158 | 3160 | 3161 | 3163 | 3164 | 3169 | 3170 |
| 3173 | 3174 | 3176 | 3177 | 3184 | 3185 | 3186 | 3188 | 3189 | 3191 | 3192 | 3193 | 3216 |
| 3217 | 3218 | 3220 | 3221 | 3223 | 3224 | 3226 | 3227 | 3228 | 3230 | 3231 | 3249 | 3250 |
| 3251 | 3253 | 3254 | 3255 | 3258 | 3259 | 3260 | 3262 | 3263 | 3264 | 3266 | 3267 | 3268 |
| 3269 | 3273 | 3274 | 3279 | 3280 | 3282 | 3283 | 3284 | 3286 | 3287 | 3316 | 3317 | 3318 |
| 3319 | 3320 | 3322 | 3323 | 3328 | 3329 | 3331 | 3332 | 3334 | 3335 | 3337 | 3338 | 3343 |
| 3344 | 3345 | 3347 | 3348 | 3349 | 3351 | 3352 | 3353 | 3354 | 3356 | 3357 | 3358 | 3359 |
| 3360 | 3361 | 3362 | 3363 | 3374 | 3375 | 3376 | 3381 | 3382 | 3383 | 3448 | 3449 | 3450 |
| 3451 | 3452 | 3453 | 3461 | 3462 | 3463 | 3464 | 3465 | 3466 | 3467 | 3469 | 3470 | 3475 |
| 3476 | 3477 | 3479 | 3480 | 3484 | 3485 | 3486 | 3487 | 3488 | 3492 | 3493 | 3494 | 3495 |
| 3496 | 3497 | 3498 | 3499 | 3500 | 3507 | 3508 | 3509 | 3510 | 3511 | 3514 | 3515 | 3516 |
| 3524 | 3525 | 3526 | 3527 | 3545 | 3546 | 3547 | 3548 | 3549 | 3550 | 3551 | 3553 | 3554 |
| 3555 | 3556 | 3557 | 3558 | 3559 | 3560 | 3561 | 3562 | 3563 | 3564 | 3565 | 3566 | 3568 |
| 3569 | 3570 | 3571 | 3575 | 3576 | 3591 | 3592 | 3593 | 3594 | 3595 | 3596 | 3597 | 3598 |
| 3599 | 3600 | 3601 | 3602 | 3603 | 3604 | 3605 | 3606 | 3607 | 3608 | 3609 | 3610 | 3612 |
| 3613 | 3620 | 3621 | 3634 | 3635 | 3651 | 3652 | 3653 | 3654 | 3657 | 3658 | 3659 | 3660 |
| 3661 | 3662 | 3663 | 3664 | 3665 | 3666 | 3668 | 3669 | 3672 | 3673 | 3676 | 3677 | 3678 |
| 3679 | 3680 | 3681 | 3682 | 3683 | 3684 | 3685 | 3687 | 3688 | 3689 | 3690 | 3691 | 3692 |
| 3693 | 3694 | 3697 | 3698 | 3699 | 3700 | 3701 | 3702 | 3703 | 3704 | 3705 | 3706 | 3707 |
| 3708 | 3712 | 3713 | 3714 | 3715 | 3716 | 3717 | 3719 | 3720 | 3728 | 3729 | 3730 | 3731 |
| 3732 | 3733 | 3748 | 3749 | 3792 | 3793 |      |      |      |      |      |      |      |
|      | 1082 | 1083 | 1085 | 1090 | 1091 | 1146 | 1149 | 1158 | 1160 | 1164 | 1172 | 1199 |
| 1201 | 1205 | 1212 | 1214 | 1218 | 1228 | 1230 | 1234 | 1245 | 1247 | 1251 | 1264 | 1266 |
| 1270 | 1284 | 1286 | 1290 | 1297 | 1299 | 1303 | 1313 | 1315 | 1319 | 1330 | 1332 | 1336 |
| 1349 | 1351 | 1355 | 1379 | 1381 | 1385 | 1395 | 1397 | 1401 | 1412 | 1414 | 1418 | 1431 |
| 1433 | 1437 | 1457 | 1459 | 1463 | 1473 | 1475 | 1479 | 1490 | 1492 | 1496 | 1509 | 1511 |
| 1515 | 1540 | 1542 | 1551 | 1571 | 1573 | 1575 | 1576 | 1577 | 1578 | 1580 | 1582 | 1583 |
| 1584 | 1585 | 1587 | 1589 | 1590 | 1591 | 1595 | 1597 | 1599 | 1605 | 1607 | 1622 | 1643 |
| 1645 | 1654 | 1673 | 1675 | 1679 | 1684 | 1685 | 1712 | 1713 | 1717 | 1730 | 1732 | 1738 |
| 1739 | 1740 | 1752 | 1753 | 1757 | 1759 | 1761 | 1763 | 1764 | 1766 | 1772 | 1774 | 1775 |
| 1776 | 1779 | 1781 | 1784 | 1785 | 1803 | 1808 | 1812 | 1839 | 1840 | 1847 | 1859 | 1861 |
| 1865 | 1877 | 1882 | 1886 | 1912 | 1913 | 1923 | 1934 | 1936 | 1946 | 1958 | 1962 | 1964 |
| 1965 | 1966 | 1970 | 1984 | 1985 | 1986 | 1988 | 1991 | 1992 | 1993 | 1996 | 1998 | 2004 |
| 2006 | 2017 | 2025 | 2026 | 2028 | 2030 | 2032 | 2033 | 2035 | 2047 | 2049 | 2051 | 2052 |
| 2056 | 2058 | 2068 | 2078 | 2080 | 2087 | 2106 | 2110 | 2111 | 2113 | 2115 | 2116 | 2117 |
| 2121 | 2123 | 2124 | 2125 | 2129 | 2168 | 2170 | 2182 | 2189 | 2191 | 2205 | 2213 | 2215 |
| 2227 | 2245 | 2247 | 2260 | 2263 | 2265 | 2273 | 2304 | 2308 | 2309 | 2311 | 2313 | 2314 |
| 2315 | 2319 | 2321 | 2322 | 2323 | 2327 | 2339 | 2341 | 2342 | 2343 | 2346 | 2363 | 2364 |
| 2366 | 2367 | 2369 | 2373 | 2374 | 2375 | 2381 | 2384 | 2385 | 2388 | 2390 | 2392 | 2393 |
| 2394 | 2396 | 2400 | 2402 | 2403 | 2404 | 2411 | 2413 | 2421 | 2422 | 2437 | 2439 | 2443 |
| 2549 | 2551 | 2553 | 2555 | 2559 | 2560 | 2561 | 2566 | 2568 | 2591 | 2593 | 2597 | 2631 |
| 2636 | 2638 | 2692 | 2694 | 2699 | 2701 | 2706 | 2707 | 2708 | 2712 | 2714 | 2747 | 2749 |
| 2754 | 2784 | 2786 | 2790 | 2822 | 2827 | 2831 | 2853 | 2855 | 2856 | 2857 | 2860 | 2873 |
| 2874 | 2877 | 2892 | 2893 | 2896 | 2914 | 2916 | 2924 | 2926 | 2927 | 2933 | 2935 | 2938 |
| 2940 | 2942 | 2945 | 2958 | 2963 | 2968 | 2970 | 2974 | 2976 | 3006 | 3008 | 3009 | 3010 |

\$LSTST= 177777





SMTYP1 001225  
SMTYP2 001231  
SMTYP3 001235  
SMTYP4 001241  
SPDCNT 013752  
SNESTL= 177777

|      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
|------|------|------|------|------|------|------|------|------|------|------|------|------|--|--|
| 968  |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| 976  |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| 979  |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| 982  |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| 4259 | 4269 |      |      |      |      |      |      |      |      |      |      |      |  |  |
| 1    | 1082 | 1090 | 1148 | 1158 | 1164 | 1172 | 1199 | 1205 | 1212 | 1218 | 1228 | 1234 |  |  |
| 1245 | 1251 | 1264 | 1270 | 1284 | 1290 | 1297 | 1303 | 1313 | 1319 | 1330 | 1336 | 1349 |  |  |
| 1355 | 1379 | 1385 | 1395 | 1401 | 1412 | 1418 | 1431 | 1437 | 1457 | 1463 | 1473 | 1479 |  |  |
| 1490 | 1496 | 1509 | 1515 | 1540 | 1551 | 1571 | 1575 | 1578 | 1582 | 1585 | 1589 | 1595 |  |  |
| 1597 | 1599 | 1605 | 1622 | 1643 | 1654 | 1673 | 1679 | 1684 | 1712 | 1717 | 1730 | 1738 |  |  |
| 1752 | 1757 | 1759 | 1761 | 1764 | 1772 | 1774 | 1779 | 1781 | 1784 | 1803 | 1812 | 1839 |  |  |
| 1847 | 1859 | 1865 | 1877 | 1886 | 1912 | 1923 | 1934 | 1946 | 1958 | 1964 | 1970 | 1984 |  |  |
| 1986 | 1991 | 1996 | 1998 | 2004 | 2017 | 2025 | 2030 | 2033 | 2047 | 2049 | 2051 | 2056 |  |  |
| 2068 | 2078 | 2087 | 2106 | 2111 | 2115 | 2121 | 2123 | 2129 | 2168 | 2182 | 2189 | 2205 |  |  |
| 2213 | 2227 | 2245 | 2260 | 2263 | 2273 | 2304 | 2309 | 2313 | 2319 | 2321 | 2327 | 2339 |  |  |
| 2363 | 2367 | 2373 | 2381 | 2384 | 2388 | 2392 | 2396 | 2402 | 2411 | 2413 | 2421 | 2437 |  |  |
| 2443 | 2549 | 2553 | 2559 | 2566 | 2568 | 2591 | 2597 | 2631 | 2638 | 2692 | 2699 | 2706 |  |  |
| 2712 | 2714 | 2747 | 2754 | 2784 | 2790 | 2822 | 2831 | 2853 | 2873 | 2877 | 2892 | 2896 |  |  |
| 2914 | 2924 | 2926 | 2933 | 2938 | 2940 | 2945 | 2958 | 2968 | 2974 | 2976 | 3006 | 3027 |  |  |
| 3033 | 3048 | 3056 | 3071 | 3078 | 3081 | 3106 | 3115 | 3182 | 3184 | 3191 | 3198 | 3226 |  |  |
| 3232 | 3258 | 3275 | 3282 | 3290 | 3290 | 3316 | 3325 | 3340 | 3343 | 3347 | 3353 | 3374 |  |  |
| 3356 | 3359 | 3364 | 3366 | 3368 | 3374 | 3379 | 3381 | 3386 | 3424 | 3458 | 3461 | 3466 |  |  |
| 3471 | 3475 | 3481 | 3484 | 3499 | 3507 | 3512 | 3523 | 3531 | 3553 | 3565 | 3573 | 3580 |  |  |
| 3593 | 3601 | 3609 | 3612 | 3618 | 3640 | 3655 | 3657 | 3660 | 3665 | 3674 | 3684 | 3695 |  |  |
| 3697 | 3700 | 3712 | 3715 | 3721 | 3723 | 3725 | 3746 | 3754 | 3790 |      |      |      |  |  |
| 1082 | 1090 | 1148 | 1172 | 1199 | 1205 | 1212 | 1218 | 1228 | 1234 | 1245 | 1251 | 1264 |  |  |
| 1270 | 1284 | 1290 | 1297 | 1303 | 1313 | 1319 | 1330 | 1336 | 1349 | 1355 | 1379 | 1385 |  |  |
| 1395 | 1401 | 1412 | 1418 | 1431 | 1437 | 1457 | 1463 | 1473 | 1479 | 1490 | 1496 | 1509 |  |  |
| 1515 | 1540 | 1551 | 1571 | 1575 | 1599 | 1605 | 1622 | 1643 | 1654 | 1673 | 1679 | 1684 |  |  |
| 1781 | 1784 | 1803 | 1812 | 1839 | 1847 | 1859 | 1865 | 1877 | 1886 | 1912 | 1923 | 1934 |  |  |
| 1946 | 1958 | 1964 | 1970 | 1984 | 1998 | 2004 | 2017 | 2025 | 2051 | 2056 | 2068 | 2078 |  |  |
| 2087 | 2106 | 2123 | 2129 | 2168 | 2182 | 2189 | 2205 | 2213 | 2227 | 2245 | 2260 | 2263 |  |  |
| 2273 | 2304 | 2321 | 2327 | 2339 | 2421 | 2437 | 2443 | 2549 | 2568 | 2591 | 2597 | 2631 |  |  |
| 2638 | 2692 | 2714 | 2747 | 2754 | 2784 | 2790 | 2822 | 2831 | 2853 | 2926 | 2933 | 2938 |  |  |
| 2940 | 2945 | 2958 | 2976 | 3006 | 3081 | 3106 | 3115 | 3182 | 3184 | 3191 | 3198 | 3226 |  |  |
| 3232 | 3258 | 3275 | 3282 | 3290 | 3316 | 3325 | 3340 | 3343 | 3347 | 3353 | 3368 | 3374 |  |  |
| 3379 | 3381 | 3386 | 3424 | 3523 | 3531 | 3573 | 3580 | 3618 | 3640 | 3746 | 3754 | 3790 |  |  |
| 1158 | 1164 | 1578 | 1582 | 1597 | 1712 | 1717 | 1730 | 1738 | 1759 | 1761 | 1774 | 1779 |  |  |
| 1986 | 1991 | 1996 | 2030 | 2049 | 2111 | 2115 | 2121 | 2309 | 2313 | 2319 | 2363 | 2384 |  |  |
| 2388 | 2392 | 2413 | 2553 | 2559 | 2566 | 2699 | 2706 | 2712 | 2873 | 2877 | 2892 | 2896 |  |  |
| 2914 | 2924 | 2968 | 2974 | 3027 | 3033 | 3048 | 3056 | 3071 | 3078 | 3262 | 3270 | 3356 |  |  |
| 3366 | 3458 | 3484 | 3499 | 3507 | 3512 | 3553 | 3565 | 3593 | 3612 | 3655 | 3697 | 3700 |  |  |
| 3725 |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| 1585 | 1589 | 1595 | 1752 | 1757 | 1764 | 1772 | 2033 | 2047 | 2367 | 2373 | 2381 | 2396 |  |  |
| 2402 | 2411 | 3359 | 3364 | 3461 | 3466 | 3471 | 3475 | 3481 | 3601 | 3609 | 3657 | 3684 |  |  |
| 3695 | 3712 | 3723 |      |      |      |      |      |      |      |      |      |      |  |  |
| 3660 | 3665 | 3674 | 3715 | 3721 |      |      |      |      |      |      |      |      |  |  |
| 931  | 3927 | 3956 |      |      |      |      |      |      |      |      |      |      |  |  |
| 1127 | 1129 | 1189 | 1191 | 1276 | 1362 | 1444 | 1446 | 1527 | 1560 | 1629 | 1663 | 1665 |  |  |
| 1793 | 1795 | 1869 | 1949 | 1951 | 2098 | 2288 | 2290 | 2486 | 2488 | 2621 | 2623 | 2814 |  |  |
| 2816 | 2951 | 3097 | 3099 | 3305 | 3307 | 3393 |      |      |      |      |      |      |  |  |
| 4369 | 4398 | 4411 |      |      |      |      |      |      |      |      |      |      |  |  |
| 4364 | 4368 | 4373 | 4376 | 4387 | 4413 |      |      |      |      |      |      |      |  |  |
| 4223 | 4239 | 4247 | 4257 | 4266 |      |      |      |      |      |      |      |      |  |  |
| 950  | 1055 | 3715 | 3806 | 3807 | 3815 | 3828 | 4253 | 4270 |      |      |      |      |  |  |
| 895  |      |      |      |      |      |      |      |      |      |      |      |      |  |  |

SNSKO = 000300

SNSK1 = 000110

SNSK2 = 000110

SNSK3 = 000110

SNULL 001154

SNWTST= 000001

SOCNT 014422

SOMODE 014424

SOVER 013736

SPASS 001202

SPASTM 001006







STAGNU= 000250

|      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 10   | 1082 | 1083 | 1084 | 1085 | 1148 | 1149 | 1159 | 1160 | 1200 | 1201 | 1213 | 1214 |
| 1229 | 1230 | 1246 | 1247 | 1265 | 1266 | 1285 | 1286 | 1298 | 1299 | 1314 | 1315 | 1331 |
| 1332 | 1350 | 1351 | 1380 | 1381 | 1396 | 1397 | 1413 | 1414 | 1432 | 1433 | 1458 | 1459 |
| 1474 | 1475 | 1491 | 1492 | 1510 | 1511 | 1541 | 1542 | 1572 | 1573 | 1575 | 1577 | 1579 |
| 1580 | 1582 | 1584 | 1586 | 1587 | 1589 | 1591 | 1606 | 1607 | 1644 | 1645 | 1674 | 1675 |
| 1684 | 1685 | 1712 | 1713 | 1731 | 1732 | 1738 | 1740 | 1752 | 1753 | 1762 | 1763 | 1765 |
| 1766 | 1774 | 1776 | 1804 | 1806 | 1807 | 1808 | 1839 | 1840 | 1860 | 1861 | 1878 | 1880 |
| 1881 | 1882 | 1912 | 1913 | 1935 | 1936 | 1959 | 1961 | 1962 | 1964 | 1966 | 1984 | 1985 |
| 1987 | 1988 | 1991 | 1993 | 1999 | 2002 | 2003 | 2005 | 2006 | 2025 | 2026 | 2027 | 2028 |
| 2031 | 2032 | 2034 | 2035 | 2057 | 2058 | 2079 | 2080 | 2107 | 2109 | 2110 | 2112 | 2113 |
| 2115 | 2117 | 2123 | 2125 | 2169 | 2170 | 2190 | 2191 | 2214 | 2215 | 2246 | 2247 | 2264 |
| 2265 | 2305 | 2307 | 2308 | 2310 | 2311 | 2313 | 2315 | 2321 | 2323 | 2340 | 2341 | 2342 |
| 2345 | 2346 | 2363 | 2364 | 2365 | 2366 | 2368 | 2369 | 2373 | 2375 | 2389 | 2390 | 2392 |
| 2394 | 2397 | 2399 | 2400 | 2402 | 2404 | 2438 | 2439 | 2550 | 2551 | 2554 | 2555 | 2559 |
| 2561 | 2592 | 2593 | 2632 | 2634 | 2635 | 2636 | 2693 | 2694 | 2700 | 2701 | 2706 | 2708 |
| 2748 | 2749 | 2785 | 2786 | 2823 | 2825 | 2826 | 2827 | 2854 | 2855 | 2856 | 2859 | 2860 |
| 2873 | 2874 | 2892 | 2893 | 2915 | 2916 | 2934 | 2935 | 2941 | 2942 | 2959 | 2961 | 2962 |
| 2963 | 2969 | 2970 | 3007 | 3008 | 3009 | 3012 | 3013 | 3027 | 3028 | 3048 | 3049 | 3072 |
| 3073 | 3107 | 3109 | 3110 | 3111 | 3182 | 3183 | 3192 | 3193 | 3227 | 3228 | 3259 | 3260 |
| 3263 | 3264 | 3283 | 3284 | 3317 | 3319 | 3320 | 3321 | 3340 | 3341 | 3348 | 3349 | 3353 |
| 3355 | 3357 | 3358 | 3360 | 3361 | 3375 | 3376 | 3382 | 3383 | 3424 | 3458 | 3459 | 3462 |
| 3463 | 3466 | 3468 | 3476 | 3477 | 3508 | 3509 | 3531 | 3554 | 3555 | 3556 | 3559 | 3560 |
| 3580 | 3594 | 3595 | 3596 | 3599 | 3600 | 3602 | 3603 | 3604 | 3607 | 3608 | 3640 | 3655 |
| 3656 | 3658 | 3659 | 3661 | 3662 | 3665 | 3667 | 3684 | 3686 | 3701 | 3702 | 3713 | 3714 |
| 3716 | 3717 | 3754 |      |      |      |      |      |      |      |      |      |      |
| 1090 | 1091 | 1093 | 1094 | 1095 | 1096 | 1102 | 1103 | 1105 | 1106 | 1107 | 1108 | 1110 |
| 1111 | 1112 | 1114 | 1115 | 1117 | 1118 | 1120 | 1121 | 1123 | 1124 | 1125 | 1137 | 1138 |
| 1140 | 1141 | 1142 | 1143 | 1144 | 1145 | 1146 | 1147 | 1150 | 1151 | 1153 | 1154 | 1164 |
| 1167 | 1168 | 1169 | 1171 | 1172 | 1173 | 1177 | 1178 | 1180 | 1181 | 1182 | 1202 | 1203 |
| 1205 | 1209 | 1210 | 1218 | 1223 | 1224 | 1225 | 1226 | 1234 | 1239 | 1240 | 1242 | 1243 |
| 1251 | 1256 | 1257 | 1259 | 1260 | 1270 | 1287 | 1288 | 1290 | 1294 | 1295 | 1303 | 1308 |
| 1309 | 1310 | 1311 | 1319 | 1324 | 1325 | 1327 | 1328 | 1336 | 1341 | 1342 | 1344 | 1345 |
| 1355 | 1376 | 1377 | 1385 | 1390 | 1391 | 1392 | 1393 | 1401 | 1406 | 1407 | 1409 | 1410 |
| 1418 | 1423 | 1424 | 1426 | 1427 | 1437 | 1454 | 1455 | 1463 | 1468 | 1469 | 1470 | 1471 |
| 1479 | 1484 | 1485 | 1487 | 1488 | 1496 | 1501 | 1502 | 1504 | 1505 | 1515 | 1538 | 1539 |
| 1551 | 1575 | 1576 | 1582 | 1583 | 1584 | 1590 | 1592 | 1593 | 1595 | 1597 | 1599 | 1602 |
| 1603 | 1610 | 1611 | 1622 | 1640 | 1641 | 1654 | 1676 | 1677 | 1679 | 1682 | 1683 | 1687 |
| 1688 | 1689 | 1690 | 1697 | 1698 | 1717 | 1725 | 1726 | 1734 | 1735 | 1738 | 1739 | 1757 |
| 1759 | 1770 | 1771 | 1772 | 1774 | 1775 | 1777 | 1778 | 1779 | 1781 | 1782 | 1784 | 1785 |
| 1809 | 1810 | 1812 | 1816 | 1817 | 1819 | 1820 | 1823 | 1824 | 1843 | 1844 | 1847 | 1852 |
| 1853 | 1865 | 1883 | 1884 | 1886 | 1890 | 1891 | 1892 | 1893 | 1898 | 1899 | 1916 | 1917 |
| 1921 | 1922 | 1923 | 1931 | 1932 | 1939 | 1940 | 1944 | 1945 | 1946 | 1964 | 1965 | 1967 |
| 1968 | 1970 | 1972 | 1973 | 1974 | 1975 | 1976 | 1977 | 1982 | 1983 | 1989 | 1990 | 1991 |
| 1992 | 1994 | 1995 | 1996 | 1998 | 2001 | 2010 | 2011 | 2014 | 2015 | 2017 | 2039 | 2040 |
| 2044 | 2045 | 2047 | 2049 | 2051 | 2052 | 2061 | 2062 | 2065 | 2066 | 2068 | 2075 | 2076 |
| 2083 | 2084 | 2087 | 2115 | 2116 | 2118 | 2119 | 2121 | 2123 | 2124 | 2126 | 2127 | 2129 |
| 2132 | 2133 | 2137 | 2138 | 2144 | 2145 | 2154 | 2155 | 2164 | 2165 | 2173 | 2174 | 2179 |
| 2180 | 2182 | 2187 | 2188 | 2195 | 2196 | 2202 | 2203 | 2205 | 2209 | 2210 | 2219 | 2220 |
| 2224 | 2225 | 2227 | 2231 | 2232 | 2236 | 2237 | 2250 | 2251 | 2257 | 2258 | 2260 | 2268 |
| 2269 | 2273 | 2313 | 2314 | 2316 | 2317 | 2319 | 2321 | 2322 | 2324 | 2325 | 2327 | 2330 |
| 2331 | 2332 | 2333 | 2334 | 2335 | 2336 | 2337 | 2343 | 2346 | 2347 | 2348 | 2350 | 2351 |
| 2353 | 2354 | 2356 | 2357 | 2359 | 2360 | 2361 | 2362 | 2371 | 2372 | 2373 | 2374 | 2377 |
| 2378 | 2379 | 2380 | 2381 | 2384 | 2385 | 2392 | 2393 | 2402 | 2403 | 2409 | 2410 | 2411 |
| 2413 | 2416 | 2417 | 2418 | 2419 | 2421 | 2422 | 2424 | 2428 | 2429 | 2430 | 2431 | 2432 |
| 2435 | 2436 | 2443 | 2499 | 2500 | 2503 | 2504 | 2506 | 2507 | 2509 | 2510 | 2511 | 2512 |
| 2513 | 2514 | 2515 | 2516 | 2528 | 2529 | 2538 | 2539 | 2559 | 2560 | 2566 | 2568 | 2572 |

STEMP = 000300

# E10

MAINDEC-11-DVDVC-A  
DVDVCA.P11

MACY11 27(1006)  
08-AUG-77 09:16

08-AUG-77 09:20 PAGE 123  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0121

|                |       |       |       |       |       |       |       |       |       |       |       |       |       |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | 2573# | 2575# | 2576# | 2578# | 2579# | 2597# | 2605# | 2606# | 2608# | 2609# | 2610# | 2638# | 2643# |
|                | 2644# | 2645# | 2646# | 2647# | 2648# | 2650# | 2651# | 2652# | 2653# | 2655# | 2656# | 2658# | 2659# |
|                | 2668# | 2669# | 2682# | 2683# | 2697# | 2698# | 2706# | 2707# | 2712# | 2714# | 2717# | 2718# | 2727# |
|                | 2728# | 2729# | 2730# | 2733# | 2734# | 2736# | 2737# | 2739# | 2740# | 2751# | 2752# | 2754# | 2756# |
|                | 2757# | 2762# | 2763# | 2764# | 2765# | 2766# | 2767# | 2773# | 2774# | 2775# | 2776# | 2790# | 2797# |
|                | 2798# | 2801# | 2802# | 2805# | 2806# | 2808# | 2809# | 2810# | 2828# | 2829# | 2831# | 2834# | 2835# |
|                | 2837# | 2838# | 2857# | 2860# | 2875# | 2876# | 2877# | 2881# | 2882# | 2894# | 2895# | 2896# | 2899# |
|                | 2900# | 2907# | 2909# | 2910# | 2911# | 2919# | 2920# | 2924# | 2926# | 2927# | 2931# | 2932# | 2938# |
|                | 2945# | 2965# | 2966# | 2971# | 2972# | 2974# | 2976# | 2979# | 2980# | 2990# | 2991# | 3003# | 3004# |
|                | 3010# | 3013# | 3030# | 3031# | 3033# | 3037# | 3038# | 3053# | 3054# | 3056# | 3060# | 3061# | 3064# |
|                | 3066# | 3067# | 3068# | 3078# | 3081# | 3082# | 3087# | 3088# | 3112# | 3113# | 3115# | 3139# | 3140# |
|                | 3142# | 3143# | 3144# | 3145# | 3148# | 3149# | 3150# | 3151# | 3154# | 3155# | 3157# | 3158# | 3160# |
|                | 3161# | 3163# | 3164# | 3169# | 3170# | 3173# | 3174# | 3176# | 3177# | 3184# | 3185# | 3188# | 3189# |
|                | 3198# | 3216# | 3218# | 3220# | 3221# | 3223# | 3224# | 3230# | 3231# | 3232# | 3249# | 3251# | 3253# |
|                | 3255# | 3266# | 3267# | 3268# | 3269# | 3270# | 3273# | 3274# | 3275# | 3279# | 3280# | 3286# | 3287# |
|                | 3290# | 3322# | 3323# | 3325# | 3328# | 3329# | 3331# | 3332# | 3334# | 3335# | 3337# | 3338# | 3343# |
|                | 3344# | 3351# | 3352# | 3353# | 3354# | 3362# | 3363# | 3364# | 3366# | 3368# | 3379# | 3386# | 3448# |
|                | 3449# | 3450# | 3451# | 3452# | 3453# | 3464# | 3465# | 3466# | 3467# | 3469# | 3470# | 3471# | 3479# |
|                | 3480# | 3481# | 3484# | 3485# | 3487# | 3497# | 3498# | 3499# | 3500# | 3512# | 3523# | 3545# | 3546# |
|                | 3547# | 3551# | 3557# | 3560# | 3561# | 3562# | 3563# | 3564# | 3565# | 3566# | 3568# | 3569# | 3573# |
|                | 3591# | 3592# | 3597# | 3600# | 3605# | 3608# | 3609# | 3610# | 3612# | 3613# | 3618# | 3634# | 3635# |
|                | 3651# | 3652# | 3653# | 3654# | 3663# | 3664# | 3665# | 3666# | 3672# | 3673# | 3674# | 3676# | 3677# |
|                | 3678# | 3679# | 3680# | 3681# | 3682# | 3683# | 3684# | 3685# | 3687# | 3688# | 3689# | 3690# | 3691# |
|                | 3692# | 3693# | 3694# | 3695# | 3697# | 3698# | 3703# | 3704# | 3705# | 3706# | 3707# | 3708# | 3719# |
|                | 3720# | 3721# | 3723# | 3725# | 3728# | 3729# | 3730# | 3731# | 3746# | 3790# |       |       |       |
| STESTN 001200  | 949#  | 1135# | 1196# | 1281# | 1367# | 1451# | 1532# | 1565# | 1634# | 1670# | 1800# | 1874# | 1956# |
| STIMES 001160  | 2103# | 2296# | 2496# | 2629# | 2820# | 2956# | 3103# | 3314# | 3759# | 4261# |       |       |       |
|                | 935#  | 1034# | 1134# | 1195# | 1202# | 1280# | 1287# | 1366# | 1450# | 1531# | 1564# | 1592# | 1633# |
|                | 1669# | 1676# | 1799# | 1809# | 1873# | 1883# | 1955# | 1967# | 2014# | 2044# | 2065# | 2102# | 2118# |
|                | 2126# | 2179# | 2202# | 2224# | 2257# | 2295# | 2316# | 2324# | 2495# | 2628# | 2819# | 2828# | 2955# |
|                | 2965# | 2971# | 3030# | 3053# | 3102# | 3112# | 3313# | 3322# | 3397# | 3805# | 4249# | 4256# | 4259# |
|                | 4269# |       |       |       |       |       |       |       |       |       |       |       |       |
| STKB 001146    | 928#  | 3959  | 3970  | 3987  | 4041  | 4047  |       |       |       |       |       |       |       |
| STKS 001144    | 927#  | 3959  | 3968  | 3984  | 4008# | 4039  | 4045  |       |       |       |       |       |       |
| STN = 000026   | 1#    | 616   | 1127  | 1134# | 1135  | 1189  | 1195# | 1196  | 1203  | 1276  | 1280# | 1281  | 1288  |
|                | 1362  | 1366# | 1367  | 1444  | 1450# | 1451  | 1527  | 1531# | 1532  | 1560  | 1564# | 1565  | 1593  |
|                | 1629  | 1633# | 1634  | 1663  | 1669# | 1670  | 1677  | 1787  | 1793  | 1799# | 1800  | 1810  | 1869  |
|                | 1873# | 1874  | 1884  | 1949  | 1955# | 1956  | 1968  | 2015  | 2045  | 2066  | 2090  | 2098  | 2102# |
|                | 2103  | 2119  | 2127  | 2180  | 2203  | 2225  | 2258  | 2276  | 2288  | 2295# | 2296  | 2317  | 2325  |
|                | 2445  | 2486  | 2495# | 2496  | 2621  | 2628# | 2629  | 2814  | 2819# | 2820  | 2829  | 2951  | 2955# |
|                | 2956  | 2966  | 2972  | 3031  | 3054  | 3097  | 3102# | 3103  | 3113  | 3201  | 3305  | 3313# | 3314  |
|                | 3323  | 3388  | 3393  | 3397# |       |       |       |       |       |       |       |       |       |
| STPB 001152    | 930#  | 3945# | 3956  |       |       |       |       |       |       |       |       |       |       |
| STPFLG 001157  | 934#  | 3894  | 3956  |       |       |       |       |       |       |       |       |       |       |
| STPS 001150    | 929#  | 3943  | 3956  |       |       |       |       |       |       |       |       |       |       |
| STRAP 014426   | 1029  | 4422# |       |       |       |       |       |       |       |       |       |       |       |
| STRAP2 014450  | 4433# | 4444  |       |       |       |       |       |       |       |       |       |       |       |
| STRP = 000012  | 4437# | 4446# | 4447# | 4448# | 4449# | 4450# | 4451  | 4452# | 4453  | 4454# | 4455# | 4456# |       |
| STRPAD 014462  | 4427  | 4444# |       |       |       |       |       |       |       |       |       |       |       |
| STSKO = 000243 | 1083# | 1090  | 1149# | 1172  | 1201# | 1205  | 1214# | 1218  | 1230# | 1234  | 1247# | 1251  | 1266# |
|                | 1270  | 1286# | 1290  | 1299# | 1303  | 1315# | 1319  | 1332# | 1336  | 1351# | 1355  | 1381# | 1385  |
|                | 1397# | 1401  | 1414# | 1418  | 1433# | 1437  | 1459# | 1463  | 1475# | 1479  | 1492# | 1496  | 1511# |
|                | 1515  | 1542# | 1551  | 1573# | 1576  | 1577# | 1599  | 1607# | 1622  | 1645# | 1654  | 1675# | 1679  |
|                | 1685# | 1784  | 1808# | 1812  | 1840# | 1847  | 1861# | 1865  | 1882# | 1886  | 1913# | 1923  | 1936# |
|                | 1946  | 1962# | 1965  | 1966# | 1970  | 1985# | 1998  | 2006# | 2017  | 2026# | 2051  | 2058# | 2068  |
|                | 2080# | 2087  | 2110# | 2124  | 2125# | 2129  | 2170# | 2182  | 2191# | 2205  | 2215# | 2227  | 2247# |







SSDST = 000067  
SSFLAG= 000001

|      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1395 | 1412 | 1431 | 1457 | 1473 | 1490 | 1509 | 1540 | 1571 | 1578 | 1585 | 1605 | 1643 |
| 1673 | 1730 | 1761 | 1764 | 1803 | 1805 | 1859 | 1877 | 1879 | 1934 | 1958 | 1960 | 1986 |
| 2004 | 2026 | 2030 | 2033 | 2056 | 2078 | 2106 | 2108 | 2111 | 2168 | 2189 | 2213 | 2245 |
| 2263 | 2304 | 2306 | 2309 | 2364 | 2367 | 2388 | 2396 | 2398 | 2437 | 2549 | 2553 | 2591 |
| 2631 | 2633 | 2692 | 2699 | 2747 | 2784 | 2822 | 2824 | 2914 | 2933 | 2940 | 2958 | 2960 |
| 2968 | 3071 | 3106 | 3108 | 3191 | 3226 | 3258 | 3262 | 3282 | 3316 | 3318 | 3347 | 3356 |
| 3359 | 3374 | 3381 | 3461 | 3475 | 3507 | 3657 | 3660 | 3700 | 3712 | 3715 |      |      |
| 2425 | 3548 |      |      |      |      |      |      |      |      |      |      |      |
| 1082 | 1083 | 1085 | 1158 | 1160 | 1164 | 1199 | 1201 | 1205 | 1212 | 1214 | 1218 | 1228 |
| 1230 | 1234 | 1245 | 1247 | 1251 | 1264 | 1266 | 1270 | 1284 | 1286 | 1290 | 1297 | 1299 |
| 1303 | 1313 | 1315 | 1319 | 1330 | 1332 | 1336 | 1349 | 1351 | 1355 | 1379 | 1381 | 1385 |
| 1395 | 1397 | 1401 | 1412 | 1414 | 1418 | 1431 | 1433 | 1437 | 1457 | 1459 | 1463 | 1473 |
| 1475 | 1479 | 1490 | 1492 | 1496 | 1509 | 1511 | 1515 | 1540 | 1542 | 1551 | 1571 | 1573 |
| 1578 | 1580 | 1585 | 1587 | 1595 | 1597 | 1599 | 1605 | 1607 | 1622 | 1643 | 1645 | 1654 |
| 1673 | 1675 | 1679 | 1712 | 1717 | 1730 | 1732 | 1752 | 1757 | 1759 | 1761 | 1763 | 1764 |
| 1766 | 1772 | 1779 | 1803 | 1805 | 1808 | 1812 | 1839 | 1847 | 1859 | 1861 | 1865 | 1877 |
| 1879 | 1882 | 1886 | 1912 | 1923 | 1934 | 1936 | 1946 | 1958 | 1960 | 1962 | 1970 | 1986 |
| 1988 | 1996 | 2004 | 2006 | 2017 | 2025 | 2026 | 2028 | 2030 | 2032 | 2033 | 2035 | 2047 |
| 2049 | 2056 | 2058 | 2068 | 2078 | 2080 | 2087 | 2106 | 2108 | 2110 | 2111 | 2113 | 2121 |
| 2129 | 2168 | 2170 | 2182 | 2189 | 2191 | 2205 | 2213 | 2215 | 2227 | 2245 | 2247 | 2260 |
| 2263 | 2265 | 2273 | 2304 | 2306 | 2308 | 2309 | 2311 | 2319 | 2327 | 2363 | 2364 | 2366 |
| 2367 | 2369 | 2381 | 2388 | 2390 | 2396 | 2398 | 2400 | 2411 | 2413 | 2437 | 2439 | 2443 |
| 2549 | 2551 | 2553 | 2555 | 2566 | 2568 | 2591 | 2593 | 2597 | 2631 | 2633 | 2636 | 2638 |
| 2692 | 2694 | 2699 | 2701 | 2712 | 2714 | 2747 | 2749 | 2754 | 2784 | 2786 | 2790 | 2822 |
| 2824 | 2827 | 2831 | 2873 | 2877 | 2892 | 2896 | 2914 | 2916 | 2924 | 2933 | 2935 | 2938 |
| 2940 | 2942 | 2945 | 2958 | 2960 | 2963 | 2968 | 2970 | 2974 | 2976 | 3027 | 3033 | 3048 |
| 3056 | 3071 | 3073 | 3078 | 3106 | 3108 | 3111 | 3115 | 3191 | 3193 | 3198 | 3226 | 3228 |
| 3232 | 3258 | 3260 | 3262 | 3264 | 3270 | 3275 | 3282 | 3284 | 3290 | 3316 | 3318 | 3321 |
| 3325 | 3347 | 3349 | 3356 | 3358 | 3359 | 3361 | 3364 | 3366 | 3368 | 3374 | 3376 | 3379 |
| 3381 | 3383 | 3386 | 3461 | 3463 | 3471 | 3475 | 3477 | 3481 | 3507 | 3509 | 3512 | 3657 |
| 3659 | 3660 | 3662 | 3674 | 3695 | 3700 | 3702 | 3712 | 3714 | 3715 | 3717 | 3721 | 3723 |
| 3725 |      |      |      |      |      |      |      |      |      |      |      |      |
| 1098 | 1702 | 1744 | 1829 | 1903 | 2147 | 2157 | 2239 | 2518 | 2542 | 2586 | 2673 | 2685 |
| 2742 | 2778 | 2847 | 2865 | 2884 | 2999 | 3019 | 3040 | 3119 | 3492 | 3668 |      |      |
| 3820 |      |      |      |      |      |      |      |      |      |      |      |      |
| 1084 | 1085 | 1159 | 1160 | 1173 | 1174 | 1200 | 1201 | 1213 | 1214 | 1229 | 1230 | 1246 |
| 1247 | 1265 | 1266 | 1285 | 1286 | 1298 | 1299 | 1314 | 1315 | 1331 | 1332 | 1350 | 1351 |
| 1380 | 1381 | 1396 | 1397 | 1413 | 1414 | 1432 | 1433 | 1458 | 1459 | 1474 | 1475 | 1491 |
| 1492 | 1510 | 1511 | 1541 | 1542 | 1572 | 1573 | 1579 | 1580 | 1586 | 1587 | 1606 | 1607 |
| 1644 | 1645 | 1674 | 1675 | 1712 | 1713 | 1731 | 1732 | 1752 | 1753 | 1762 | 1763 | 1765 |
| 1766 | 1806 | 1807 | 1839 | 1840 | 1860 | 1861 | 1880 | 1881 | 1912 | 1913 | 1935 | 1936 |
| 1959 | 1960 | 1961 | 1962 | 1987 | 1988 | 2001 | 2002 | 2005 | 2006 | 2027 | 2028 | 2031 |
| 2032 | 2034 | 2035 | 2057 | 2058 | 2079 | 2080 | 2107 | 2108 | 2109 | 2110 | 2112 | 2113 |
| 2169 | 2170 | 2190 | 2191 | 2214 | 2215 | 2246 | 2247 | 2264 | 2265 | 2305 | 2306 | 2307 |
| 2308 | 2310 | 2311 | 2365 | 2366 | 2368 | 2369 | 2389 | 2390 | 2397 | 2398 | 2399 | 2400 |
| 2438 | 2439 | 2550 | 2551 | 2554 | 2555 | 2592 | 2593 | 2634 | 2635 | 2693 | 2694 | 2700 |
| 2701 | 2748 | 2749 | 2785 | 2786 | 2825 | 2826 | 2873 | 2874 | 2892 | 2893 | 2915 | 2916 |
| 2934 | 2935 | 2941 | 2942 | 2961 | 2962 | 2969 | 2970 | 3027 | 3028 | 3048 | 3049 | 3072 |
| 3073 | 3109 | 3110 | 3185 | 3186 | 3192 | 3193 | 3227 | 3228 | 3259 | 3260 | 3263 | 3264 |
| 3283 | 3284 | 3319 | 3320 | 3344 | 3345 | 3348 | 3349 | 3357 | 3358 | 3360 | 3361 | 3375 |
| 3376 | 3382 | 3383 | 3462 | 3463 | 3476 | 3477 | 3508 | 3509 | 3658 | 3659 | 3661 | 3662 |
| 3698 | 3699 | 3701 | 3702 | 3713 | 3714 | 3716 | 3717 |      |      |      |      |      |
| 3424 | 3526 | 3531 | 3575 | 3580 | 3620 | 3640 | 3748 | 3754 | 3792 |      |      |      |
| 3424 | 3510 | 3514 | 3524 | 3531 | 3574 | 3580 | 3619 | 3640 | 3747 | 3754 | 3791 |      |
| 3424 | 3510 | 3523 | 3531 | 3570 | 3573 | 3580 | 3618 | 3640 | 3732 | 3746 | 3754 | 3790 |
| 3424 | 3515 | 3525 | 3531 | 3574 | 3580 | 3619 | 3640 | 3747 | 3754 | 3791 |      |      |

SSFROM= 000000

SSGET4= 000000  
SSLOC = 011226

SSLOCN= 000000  
SSRETU= 000000  
SSRTN1= 000246  
SSRTN2= 000247

\$\$\$RC = 000027  
\$\$TO = 000000

|       |       |       |       |       |       |       |       |       |       |       |       |       |  |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| 2425# | 3548# |       |       |       |       |       |       |       |       |       |       |       |  |
| 1098# | 1099  | 1702# | 1708  | 1744# | 1750  | 1829# | 1835  | 1903# | 1909  | 2147# | 2150  | 2157# |  |
| 2160  | 2239# | 2242  | 2518# | 2524  | 2542# | 2545  | 2586# | 2589  | 2673# | 2679  | 2685# | 2688  |  |
| 2742# | 2745  | 2778# | 2781  | 2847# | 2848  | 2849  | 2865# | 2871  | 2884# | 2890  | 2999# | 3000  |  |
| 3001  | 3019# | 3025  | 3040# | 3046  | 3119# | 3120  | 3121  | 3492# | 3495  | 3668# | 3669  |       |  |
| 4363# | 4367# | 4377  | 4412# |       |       |       |       |       |       |       |       |       |  |
| 1082# | 1090  |       |       |       |       |       |       |       |       |       |       |       |  |
| 1246  | 1251# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2112  | 2116# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2115  | 2121# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2123  | 2129# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2169  | 2182# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2190  | 2205# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2214  | 2227# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2246  | 2260# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2264  | 2273# |       |       |       |       |       |       |       |       |       |       |       |  |
| 1265  | 1270# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2305  | 2307  | 2322# |       |       |       |       |       |       |       |       |       |       |  |
| 2310  | 2314# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2313  | 2319# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2321  | 2327# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2340  | 2343# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2341# | 2421  |       |       |       |       |       |       |       |       |       |       |       |  |
| 2345  | 2422# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2363# | 2384  |       |       |       |       |       |       |       |       |       |       |       |  |
| 1285  | 1290# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2365  | 2385# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2368  | 2374# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2373  | 2381# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2389  | 2393# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2392  | 2413# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2397  | 2399  | 2403# |       |       |       |       |       |       |       |       |       |       |  |
| 2402  | 2411# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2438  | 2443# |       |       |       |       |       |       |       |       |       |       |       |  |
| 1298  | 1303# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2550  | 2568# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2554  | 2560# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2559  | 2566# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2592  | 2597# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2632  | 2635# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2634  | 2638# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2693  | 2714# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2700  | 2707# |       |       |       |       |       |       |       |       |       |       |       |  |
| 1314  | 1319# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2706  | 2712# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2748  | 2754# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2785  | 2790# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2823  | 2826# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2825  | 2831# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2854  | 2857# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2855# | 2926  |       |       |       |       |       |       |       |       |       |       |       |  |
| 2859  | 2927# |       |       |       |       |       |       |       |       |       |       |       |  |
| 1331  | 1336# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2873  | 2877# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2892  | 2896# |       |       |       |       |       |       |       |       |       |       |       |  |
| 2915  | 2924# |       |       |       |       |       |       |       |       |       |       |       |  |

\$OFILL 014423  
 \$1 001712  
 \$10 002552  
 \$100 004740  
 \$101 004750  
 \$102 004762  
 \$103 005070  
 \$104 005124  
 \$105 005162  
 \$106 005240  
 \$107 005260  
 \$11 002602  
 \$110 005410  
 \$111 005376  
 \$112 005406  
 \$113 005420  
 \$114 005454  
 \$115 005452  
 \$116 005626  
 \$117 005512  
 \$12 002640  
 \$120 005546  
 \$121 005534  
 \$122 005544  
 \$123 005560  
 \$124 005610  
 \$125 005602  
 \$126 005610  
 \$127 005674  
 \$13 002660  
 \$130 006154  
 \$131 006152  
 \$132 006154  
 \$133 006232  
 \$134 006320  
 \$135 006324  
 \$136 006514  
 \$137 006512  
 \$14 002706  
 \$140 006514  
 \$141 006606  
 \$142 006704  
 \$143 007004  
 \$144 007014  
 \$145 007060  
 \$146 007056  
 \$147 007222  
 \$15 002734  
 \$150 007126  
 \$151 007172  
 \$152 007220



|       |        |       |       |
|-------|--------|-------|-------|
| \$153 | 007242 | 2934  | 2938# |
| \$154 | 007254 | 2941  | 2945# |
| \$155 | 007312 | 2959  | 2962# |
| \$156 | 007342 | 2961  | 2976# |
| \$157 | 007342 | 2969  | 2974# |
| \$16  | 002764 | 1350  | 1355# |
| \$160 | 007414 | 3007  | 3010# |
| \$161 | 007412 | 3008# | 3081  |
| \$162 | 007560 | 3012  | 3082# |
| \$163 | 007466 | 3027  | 3033# |
| \$164 | 007536 | 3048  | 3056# |
| \$165 | 007556 | 3072  | 3078# |
| \$166 | 007624 | 3107  | 3110# |
| \$167 | 007634 | 3109  | 3115# |
| \$17  | 003034 | 1380  | 1385# |
| \$170 | 007742 | 3182# | 3185  |
| \$171 | 007766 | 3192  | 3198# |
| \$172 | 010032 | 3227  | 3232# |
| \$173 | 010110 | 3259  | 3275# |
| \$174 | 010104 | 3263  | 3270# |
| \$175 | 010126 | 3283  | 3290# |
| \$176 | 010172 | 3317  | 3320# |
| \$177 | 010202 | 3319  | 3325# |
| \$2   | 002122 | 1084  | 1091# |
| \$20  | 003062 | 1396  | 1401# |
| \$200 | 010230 | 3340# | 3344  |
| \$201 | 010256 | 3348  | 3354# |
| \$202 | 010304 | 3353  | 3368# |
| \$203 | 010304 | 3357  | 3366# |
| \$204 | 010304 | 3360  | 3364# |
| \$205 | 010320 | 3375  | 3379# |
| \$206 | 010332 | 3382  | 3386# |
| \$207 | 010634 | 3510  | 3523# |
| \$21  | 003110 | 1413  | 1418# |
| \$210 | 010636 | 3515  | 3525# |
| \$211 | 010504 | 3458# | 3499  |
| \$212 | 010610 | 3485  | 3487  |
| \$213 | 010524 | 3462  | 3467# |
| \$214 | 010532 | 3466  | 3471# |
| \$215 | 010550 | 3476  | 3481# |
| \$216 | 010622 | 3508  | 3512# |
| \$217 | 010736 | 3570  | 3573# |
| \$22  | 003140 | 1432  | 1437# |
| \$220 | 010736 | 3574# |       |
| \$221 | 010702 | 3554  | 3557# |
| \$222 | 010676 | 3555# | 3565  |
| \$223 | 010726 | 3559  | 3566# |
| \$224 | 011016 | 3618# |       |
| \$225 | 011016 | 3619# |       |
| \$226 | 010764 | 3594  | 3597# |
| \$227 | 010760 | 3595# | 3612  |
| \$23  | 003176 | 1458  | 1463# |
| \$230 | 011010 | 3599  | 3613# |
| \$231 | 010776 | 3602  | 3605# |
| \$232 | 010774 | 3603# | 3609  |
| \$233 | 011006 | 3607  | 3610# |

3500#



|          |         |       |       |
|----------|---------|-------|-------|
| \$234    | 011272  | 3732  | 3746# |
| \$235    | 011272  | 3747# |       |
| \$236    | 011044  | 3655# | 3698  |
| \$237    | 011130  | 3658  | 3685# |
| \$24     | 003224  | 1474  | 1479# |
| \$240    | 011070  | 3661  | 3666# |
| \$241    | 011076  | 3665  | 3674# |
| \$242    | 011150  | 3684  | 3695# |
| \$243    | 011236  | 3701  | 3725# |
| \$244    | 011236  | 3713  | 3723# |
| \$245    | 011236  | 3716  | 3721# |
| \$246    | 011462  | 3790# |       |
| \$247    | 011462  | 3791# |       |
| \$25     | 003252  | 1491  | 1496# |
| \$26     | 003302  | 1510  | 1515# |
| \$27     | 003342  | 1541  | 1551# |
| \$3      | 002320  | 1148# | 1173  |
| \$30     | 003372  | 1572  | 1576# |
| \$31     | 003426  | 1575  | 1599# |
| \$32     | 003404  | 1579  | 1583# |
| \$33     | 003426  | 1582  | 1597# |
| \$34     | 003416  | 1586  | 1590# |
| \$35     | 003426  | 1589  | 1595# |
| \$36     | 003456  | 1606  | 1622# |
| \$37     | 003516  | 1644  | 1654# |
| \$4      | 002344  | 1159  | 1164# |
| \$40     | 003554  | 1674  | 1679# |
| \$40CAT= | ##### U | 4180  | 4222  |
| \$41     | 003562  | 1684# | 1784  |
| \$42     | 003772  | 1782  | 1785# |
| \$43     | 003636  | 1712  | 1717# |
| \$44     | 003664  | 1731  | 1739# |
| \$45     | 003720  | 1738  | 1759# |
| \$46     | 003720  | 1752  | 1757# |
| \$47     | 003752  | 1762  | 1775# |
| \$5      | 002456  | 1200  | 1205# |
| \$50     | 003750  | 1765  | 1772# |
| \$51     | 003760  | 1774  | 1779# |
| \$52     | 004040  | 1804  | 1807# |
| \$53     | 004050  | 1806  | 1812# |
| \$54     | 004132  | 1839  | 1847# |
| \$55     | 004154  | 1860  | 1865# |
| \$56     | 004212  | 1878  | 1881# |
| \$57     | 004222  | 1880  | 1886# |
| \$6      | 002476  | 1213  | 1218# |
| \$60     | 004312  | 1912  | 1923# |
| \$61     | 004344  | 1935  | 1946# |
| \$62     | 004404  | 1959  | 1961  |
| \$63     | 004414  | 1964  | 1970# |
| \$64     | 004440  | 1984# | 2001  |
| \$65     | 004460  | 1987  | 1992# |
| \$66     | 004464  | 1991  | 1996# |
| \$67     | 004504  | 1999  | 2002# |
| \$7      | 002524  | 1229  | 1234# |
| \$70     | 004534  | 2005  | 2017# |
| \$71     | 004534  | 2025# | 2051  |

1965#



L10