

PDP11

BOOTSTRAP/TERMINATOR
MD-11-DZM9A-D
(M9301,M9400)

EP-DZM9A-D-DL
COPYRIGHT © 76-77
FICHE 1 OF 1

AUG 1978
digital
MADE IN USA



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
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

LIST 2

IDENTIFICATION

| | |
|---------------|-------------------------------------|
| PRODUCT CODES | -RINDIC-11-02149A-0-0 |
| PRODUCT NAME | BOOTSTRAP/TERMINATOR (M9301, M9470) |
| PROGRAM DATE | MAY 27, 1977 |
| MAINTENANCE | DIAGNOSTIC ENGINEERING |

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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, 1976 BY DIGITAL EQUIPMENT CORPORATION

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

.....
•
• SUMMARY OF OPERATING INSTRUCTIONS •
•
.....

THE FOLLOWING PROCEDURE CAN BE USED TO RUN THIS DIAGNOSTIC
IN A DEVICE VERIFICATION MODE. IF THE PROGRAM DOES NOT
RUN SUCCESSFULLY CONSULT THE FOLLOWING DOCUMENT FOR ASSISTANCE.

OPERATING PROCEDURES:

1. LOAD THE PROGRAM USING NORMAL PROCEDURES.
2. LOAD ADDRESS 20F
3. SET SWITCH REGISTER TO SELECT THE PROPER
VERSION OF THE ROM UNDER TEST.
(SEE INSTRUCTIONS IF A SOFTWARE SWITCH REGISTER
IS TO BE USED.)
4. PRESS START
5. THE PROGRAM SHOULD TAKE ABOUT 1 SEC TO
COMPLETE THE TEST AND PRINT: "END OF TEST".
6. IF THE PROGRAM DOES NOT RUN AS DESCRIBED
ABOVE, CONSULT THE FULL OPERATING INSTRUCTIONS
WHICH FOLLOW.

• • CAUTION • •

BECAUSE THE CONTENTS READ FROM LOCATION 773024
OF THE *9301 OPTION IS CONFIGURATION DEPENDANT (SWITCH
REGISTER DEPENDANT), THIS LOCATION IS NOT INCLUDED IN
THE DATA CHECK.
THIS LOCATION CAN BE VERIFIED BY EXAMINING IT OR BY USING
THE ALTERNATE STARTING ADDRESS (SEE SECTION 2.1.3) TO
PRINT OUT THIS LOCATION.

91
92 1.0 GENERAL PROGRAM INFORMATION
93
94
95 1.1 PROGRAM PURPOSE
96
97
98 THIS DIAGNOSTIC PROGRAM IS INTENDED TO VERIFY THE
99 ROM CONTENTS OF THE ROM BOOTSTRAP MODULES. THE PROGRAM
100 COMPUTES AND CHECKS A CYCLIC REDUNDANCY CHARACTER
101 AND A LONGITUDINAL PARITY CHARACTER FOR THE CONTENTS
102 OF THE ROM STORAGE AVAILABLE IN AN M9301 OR M9400 MODULE.
103
104 A SEPARATE ROUTINE INCLUDED ALLOWS THE USER TO TYPE
105 THE CONTENTS OF THE ROM STORAGE ON THE TELETYPE AS
106 AN AID TO DEBUGGING.
107
108 1.2 SYSTEM REQUIREMENTS
109
110
111 1.2.1 HARDWARE
112
113
114 PDF/11 PROCESSOR
115 TELETYPE OR EQUIVALENT
116 4K OF MEMORY
117 M9301 OR M9400 MODULE
118
119 1.2.2 SOFTWARE
120
121
122 THIS PROGRAM IS WRITTEN TO BE RUN AS A STAND-ALONE PROGRAM.
123 HOWEVER, THE PROGRAM IS DESIGNED TO RUN UNDER AUTOMATED
124 PRODUCT TEST SYSTEM (APT) IN ALL THREE MODES.
125
126 THE PROGRAM CAN ALSO BE RUN UNDER THE ACT 11 MONITOR.
127
128 1.3 RELATED DOCUMENTS AND STANDARDS
129
130
131 DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS PROGRAMMING PRACTICES
132 DOCUMENT NO. 175-803-009-00
133
134 APT INTERFACE SPECIFICATION, REV. 13
135
136 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
137
138
139 NONE, HOWEVER THE CPU IS ASSUMED TO BE FUNCTIONING
140
141 1.5 FAILURE ASSUMPTIONS
142
143
144 THE PROCESSOR IS ASSUMED TO BE FUNCTIONING PROPERLY.

145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING

USE NORMAL PROCEDURES FOR LOADING DIAGNOSTIC PROGRAMS.

2.1.2 NORMAL START

1. LOAD SOFTWARE SWITCH REGISTER (IF USED) TO SELECT THE
ROM VERSION UNDER TEST. (SEE 2.3)

2. LOAD ADDRESS 200

3. SET HARDWARE SWITCH REGISTER (IF AVAILABLE) TO SELECT THE
ROM VERSION UNDER TEST (SEE 2.3).

4. START

2.1.3 OPTIONAL START

THE OPTIONAL STARTING ADDRESS IS USED TO TYPE OUT THE CONTENTS
OF THE ROM FOR USE IN VISUAL VERIFICATION OR AS A DEBUGGING
TOOL.

USE THE SAME PROCEDURE AS A NORMAL START EXCEPT USE
ADDRESS 210 IN STEP 2.

2.2 SPECIAL ENVIRONMENTS

THIS PROGRAM IS WRITTEN TO COMPLY WITH ALL THE REQUIREMENTS
OF THE APT INTERFACE SPECIFICATION. IT WILL RUN UNDER APT
IN EITHER QUICK VERIFY, PROGRAM OR RUN-TIME MODES.

THIS PROGRAM IS WRITTEN TO COMPLY WITH THE ACT11/XXDP
INTERFACE REQUIREMENTS.

WHEN RUNNING IN ACT11 QUICK VERIFY OR XXDP CHAIN MODE
(LOC. 42 NOT #), OR APT QUICK VERIFY AND PROGRAM MODE
THE PROGRAM ATTEMPTS TO RUN WITHOUT OPERATOR
INTERVENTION OR SWITCH REGISTER SELECTION. THE
COMPUTED CRC IS COMPARED AGAINST THE CRC FOR ALL KNOWN
VERSIONS OF THE ROM BOOTSTRAP. WHEN A MATCH IS FOUND THE
VERSION OF THE MODULE IS TYPED FOR VISUAL VERIFICATION.

198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253

2.3 PROGRAM OPTIONS

THE PROGRAM AUTOMATICALLY CHECKS FOR THE PRESENCE OF A HARDWARE SWITCH REGISTER. IF NO RESPONSE IS FOUND WHEN ADDRESSING THE HARDWARE SWR (177570), THE ADDRESS OF THE SOFTWARE SWP (176) IS SUBSTITUTED.

FOR PROCESSORS WITH NO HARDWARE SWITCH REGISTER, THE OPERATOR SHOULD SET THE DESIRED SWITCH VALUE IN LOCATION 170.

WARNING... IN ORDER TO ALLOW TESTING OF M7942-YA BOARDS ON THE V171, IF LOCATION 170 IS SET TO 6, THE SOFTWARE SWITCH REGISTER WILL BE USED REGARDLESS OF HARDWARE SWITCH REGISTER AVAILABILITY.

2.3.1 SWITCH SELECTION

THE SWITCH REGISTER (HARDWARE OR SOFTWARE) IS USED TO SELECT THE VERSION OF THE ROM BOOTSTRAP TO BE TESTED ACCORDING TO THE FOLLOWING TABLE. NOTE: THESE SETTINGS ARE OCTAL NUMBERS. THEY ARE NOT PARTICULAR SWITCHES SET TO A ONE. FOR EXAMPLE, TO SELECT THE M9301-YH VERSION, SET SWITCHES 03 AND 01 IN THE SWITCH REGISTER. THIS CORRESPONDS TO AN OCTAL 12.

| SWR | MODULE VERSION |
|-----|------------------|
| --- | ----- |
| 1 | M9301-YA |
| 2 | M9301-YB |
| 3 | M9301-YC |
| 4 | M9400-YA (OR YC) |
| 5 | M9301-YD |
| 6 | M7942-YA |
| 7 | M9301-YE |
| 10 | M9400-YH (OR YK) |
| 11 | M9311 |
| 12 | M9301-YH |
| 13 | M9301-YE |
| 14 | M9301-YJ |

IF THE CRC AND LPC FOR NEW VERSIONS ARE KNOWN BUT NOT IN THE ABOVE TABLE, SET THE SWITCH REGISTER TO ZERO AND ANSWER THE TELETYPE DIALOG.

TO DETERMINE THE CRC AND LPC FOR A NEW VERSION, START THE DIAGNOSTIC AT 200 WITH SWR00. ANSWER 0 TO THE REQUESTS FOR THE LPC AND CRC. THE RESULTING MESSAGES WILL INDICATE THE CORRECT FUTURE RESPONSES FOR CRC AND LPC PROVIDED THE TEST IS RUN ON A KNOWN-GOOD MODULE.

2.3.2 TELETYPE DIALOG

SEVERAL QUESTIONS ARE ASKED OF THE OPERATOR IN ORDER

254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
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

TO OBTAIN SUFFICIENT INFORMATION FOR TESTING A ROM MODULE NOT PREVIOUSLY SUPPORTED IN THE DIAGNOSTIC, THE DIALOG IS INITIATED IF THE PROGRAM IS STARTED WITH THE SWH B U. ALL RESPONSES ARE IN OCTAL AND TERMINATED BY A CARRIAGE RETURN.

ALL RESPONSES ARE CHECKED FOR VALID OCTAL NUMBERS. IF AN ILLEGAL CHARACTER IS TYPED, THE PROGRAM WILL TYPE A "?", CARRIAGE RETURN-LINE FEED AND AWAIT THE PROPER INPUT.

IF A MISTAKE IS NOTICED BEFORE THE CARRIAGE RETURN IS USED TO TERMINATE THE INPUT, A RUBOUT CAN BE USED TO DELETE MISTYPED INPUT.

1. TYPE CRC VALUE:
THIS REQUESTS THE VALUE OF THE CYCLIC REDUNDANCY CHECK PREVIOUSLY CALCULATED FOR THIS VERSION OF THE ROM MODULE. IT IS THE VALUE AGAINST WHICH THE UNIT UNDER TEST'S CRC WILL BE COMPARED.

2. TYPE LPC VALUE:
THIS REQUESTS THE VALUE OF THE LONGITUDINAL PARITY CHECK PREVIOUSLY CALCULATED FOR THIS VERSION OF THE ROM MODULE. IT IS THE VALUE AGAINST WHICH THE UNIT UNDER TEST'S LPC WILL BE COMPARED.

3. TYPE STARTING ADDR. OF 1ST ROM ADDR. SPACE:
THIS QUESTION REFERS TO THE FACT THAT THE ROM SPACE IN AROM BOOTSTRAP MODULE IS DIVIDED INTO 2 DISTINCT ADDRESS SPACES. TYPE THE STARTING ADDRESS OF THE 1ST RANGE OF ADDRESSES. THE STANDARD M9301 & M9400 BEGIN AT 173000.

4. TYPE LENGTH (BYTES) OF 1ST ROM ADDR. SPACE:
THIS REQUESTS THE LENGTH OF THE 1ST GROUP OF ROM ADDRESSES IN BYTES. THE STANDARD M9301 & M9400 HAVE AN INITIAL ADDRESS SPACE OF 1000 BYTES. IF THIS SECTION OF ADDRESSES IS NOT USED BY THIS VERSION, ANSWER 0 TO THIS QUESTION.

5. TYPE STARTING ADDR. OF 2ND ROM ADDR. SPACE:
THIS REFERS TO THE FIRST ADDRESS IN THE SECOND DISTINCT GROUP OF ROM ADDRESSES. THE RESPONSE FOR A STANDARD M9301 & M9400 WOULD BE 165000.

6. TYPE LENGTH (BYTES) OF 2ND ROM ADDR. SPACE:
THIS REQUESTS THE LENGTH OF THE 2ND GROUP OF ROM ADDRESSES IN BYTES. THE STANDARD M9301 & M9400 HAVE A SECOND ADDRESS SPACE OF 1000 BYTES. IF THIS SECTION OF ADDRESSES IS NOT USED BY THIS VERSION, ANSWER 0 TO THIS QUESTION.

2.4 EXECUTION TIMES

THE DIAGNOSTIC COMPLETES 1 PASS IN LESS THAN 1 SEC. ONCE THE INPUT DIALOG HAS BEEN COMPLETED. THE PROGRAM WILL HALT UPON COMPLETION; HOWEVER, IF RUNNING UNDER APT THE PROGRAM WILL CYCLE CONTINUOUSLY.

F1

310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
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

3.0 ERROR INFORMATION

WHEN THE DIAGNOSTIC DETECTS A DISCREPANCY BETWEEN THE EXPECTED AND COMPUTED CRC OF LPC BOTH ARE PRINTED ON THE TELETYPE.

ANY DISCREPANCY IN THE LPC CAN ASSIST IN ISOLATING THE PROBLEM TO THE ROM IC.

UNDER APT THE ERROR IS INDICATED BY DEPOSITING ERROR INFORMATION IN THE APT MAILBOX BEFORE HALTING.

4.0 PROGRESS REPORTS

AT THE END OF EACH PASS THE PROGRAM INCREMENTS TO THE LOCATION SPASS WHICH IS IN THE APT MAILBOX. THIS LOCATION WILL ALWAYS CONTAIN THE NUMBER OF PASSES COMPLETED. SPASS IS RESET WITH EVERY RESTART.

ADDITIONALLY, THE MESSAGE "END OF TEST" IS PRINTED ON THE CONSOLE TELETYPE AFTER EACH PASS. NORMALLY ONLY ONE PASS NEEDS TO RUN TO VERIFY THE MODULE.

5.0 TROUBLE SHOOTING

THE ALGORITHM FOR COMPUTING THE CRC IS THE SAME AS THAT USED ON 9-TRACK MAGNETIC TAPE WITH ODD PARITY. THE CRC IS CALCULATED ON A BYTE-BY-BYTE BASIS. WHILE THE ALGORITHM IS SUCCESSFUL IN DETECTING MULTIPLE ERRORS, ITS USE AS A DEBUGGING AID IS LIMITED.

THE LPC IS CALCULATED BY ASSEMBLING THE XOR OF EVERY WORD IN THE ROM. WHILE ONLY USEFUL IN CATCHING AN ODD NUMBER OF ERRORS IN EACH BIT POSITION, IT IS VERY USEFUL IN ISOLATING THE PROBLEM TO A CHIP. BY LOCATING WHICH BIT POSITIONS ARE IN DISCREPANCY, THE CORRESPONDING ROM CHIPS CAN BE ISOLATED.

IF NO OTHER CLUES CAN BE OBTAINED, START THE PROGRAM AT 210 AND COMPARE THE PRINTOUT OF THE CODE WITH THE LISTING FOR THE VERSION BEING TESTED.

• • CAUTION • •

BECAUSE THE CONTENTS READ FROM LOCATION 773024 OF THE #9301 OPTION IS CONFIGURATION DEPENDANT(SWITCH REGISTER DEPENDANT), THIS LOCATION IS NOT INCLUDED IN THE DATA CHECK. THIS LOCATION CAN BE VERIFIED BY EXAMINING IT OR BY USING THE ALTERNATE STARTING ADDRESS(SEE SECTION 2.1.3) TO PRINT OUT THIS LOCATION.

366
367
368
369
370
371
372
373
374

0.0 LISTING

.ENDR

```

375
376          .FNABLE ANb
377          .LIST ME
378          .LIST MC,MD,CND
379          000000          660000
380          .BPTTL BASIC DEFINITIONS
381
382          ;INITIAL ADDRESS OF THE STACK POINTER 000 1100 000
383          001100          STACK= 1100
384          .FOHIV FMT,ERROR          ;;BASIC DEFINITION OF ERROR CALL
385          .FOHIV ICT,SCOPE          ;;BASIC DEFINITION OF SCOPE CALL
386
387          ;MISCELLANEOUS DEFINITIONS
388          000111          HT= 11          ;;CODE FOR HORIZONTAL TAB
389          000112          LF= 12          ;;CODE FOR LINE FEED
390          000115          CR= 15          ;;CODE FOR CARRIAGE RETURN
391          000200          CPLF= 200          ;;CODE FOR CARRIAGE RETURN=LINE FEED
392          177770          FS= 177776          ;;PROCESSOR STATUS WORD
393          .EQUIV PS,PSW          .EQUIV PS,PSW
394          177774          STKLMT= 177774          ;;STACK LIMIT REGISTER
395          177772          PIRG= 177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
396          177570          DSWR= 177570          ;;HARDWARE SWITCH REGISTER
397          177570          DDISP= 177570          ;;HARDWARE DISPLAY REGISTER
398
399          ;GENERAL PURPOSE REGISTER DEFINITIONS
400          000000          R0= 00          ;;GENERAL REGISTER
401          000001          R1= 01          ;;GENERAL REGISTER
402          000002          R2= 02          ;;GENERAL REGISTER
403          000003          R3= 03          ;;GENERAL REGISTER
404          000004          R4= 04          ;;GENERAL REGISTER
405          000005          R5= 05          ;;GENERAL REGISTER
406          000006          R6= 06          ;;GENERAL REGISTER
407          000007          R7= 07          ;;GENERAL REGISTER
408          000006          SP= 06          ;;STACK POINTER
409          000007          PC= 07          ;;PROGRAM COUNTER
410
411          ;PRIORITY LEVEL DEFINITIONS
412          000000          PR0= 0          ;;PRIORITY LEVEL 0
413          000040          PR1= 40          ;;PRIORITY LEVEL 1
414          000100          PR2= 100          ;;PRIORITY LEVEL 2
415          000140          PR3= 140          ;;PRIORITY LEVEL 3
416          000200          PR4= 200          ;;PRIORITY LEVEL 4
417          000240          PR5= 240          ;;PRIORITY LEVEL 5
418          000300          PR6= 300          ;;PRIORITY LEVEL 6
419          000340          PR7= 340          ;;PRIORITY LEVEL 7
420
421          ;"SWITCH REGISTER" SWITCH DEFINITIONS
422          100000          SW15= 100000
423          040000          SW14= 400000
424          070000          SW13= 200000
425          010000          SW12= 100000
426          004000          SW11= 400000
427          002000          SW10= 200000
428          001000          SW09= 100000
429          000400          SW08= 400000
430          000200          SW07= 200000
  
```

| | | | |
|-----|--------|--------|----------|
| 431 | 000100 | SW06 | 100 |
| 432 | 000101 | SW05 | 00 |
| 433 | 000020 | SW04 | 20 |
| 434 | 000010 | SW03 | 10 |
| 435 | 000004 | SW02 | 4 |
| 436 | 000002 | SW01 | 2 |
| 437 | 000001 | SW00 | 1 |
| 438 | | .EQUIV | SW09,SW9 |
| 439 | | .EQUIV | SW08,SW8 |
| 440 | | .EQUIV | SW07,SW7 |
| 441 | | .EQUIV | SW06,SW6 |
| 442 | | .EQUIV | SW05,SW5 |
| 443 | | .EQUIV | SW04,SW4 |
| 444 | | .EQUIV | SW03,SW3 |
| 445 | | .EQUIV | SW02,SW2 |
| 446 | | .EQUIV | SW01,SW1 |
| 447 | | .EQUIV | SW00,SW0 |

;;DATA BIT DEFINITIONS (BIT00 TO BIT15)

| | | | |
|-----|--------|--------|------------|
| 450 | 100000 | BIT15 | 100000 |
| 451 | 010000 | BIT14 | 010000 |
| 452 | 020000 | BIT13 | 020000 |
| 453 | 010000 | BIT12 | 100000 |
| 454 | 004000 | BIT11 | 004000 |
| 455 | 002000 | BIT10 | 002000 |
| 456 | 001000 | BIT09 | 001000 |
| 457 | 000400 | BIT08 | 000400 |
| 458 | 000200 | BIT07 | 000200 |
| 459 | 000100 | BIT06 | 000100 |
| 460 | 000040 | BIT05 | 000040 |
| 461 | 000020 | BIT04 | 000020 |
| 462 | 000010 | BIT03 | 000010 |
| 463 | 000004 | BIT02 | 000004 |
| 464 | 000002 | BIT01 | 000002 |
| 465 | 000001 | BIT00 | 000001 |
| 466 | | .EQUIV | BIT09,BIT9 |
| 467 | | .EQUIV | BIT08,BIT8 |
| 468 | | .EQUIV | BIT07,BIT7 |
| 469 | | .EQUIV | BIT06,BIT6 |
| 470 | | .EQUIV | BIT05,BIT5 |
| 471 | | .EQUIV | BIT04,BIT4 |
| 472 | | .EQUIV | BIT03,BIT3 |
| 473 | | .EQUIV | BIT02,BIT2 |
| 474 | | .EQUIV | BIT01,BIT1 |
| 475 | | .EQUIV | BIT00,BIT0 |

;;BASIC "CPU" TRAP VECTOR ADDRESSES

| | | | | |
|-----|--------|---------|----|-------------------------------------|
| 478 | 000004 | ERRVEC | 4 | ;;TIME OUT AND OTHER ERRORS |
| 479 | 000010 | RESVEC | 10 | ;;RESERVED AND ILLEGAL INSTRUCTIONS |
| 480 | 000014 | TBITVEC | 14 | ;;"T" BIT |
| 481 | 000014 | TRIVEC | 14 | ;;TRACE TRAP |
| 482 | 000014 | BPTVEC | 14 | ;;BREAKPOINT TRAP (BPT) |
| 483 | 000020 | IOTVEC | 20 | ;;INPUT/OUTPUT TRAP (IOT) **SCOPE** |
| 484 | 000024 | PWFVEC | 24 | ;;POWER FAIL |
| 485 | 000030 | EMTVEC | 30 | ;;EMULATOR TRAP (EMT) **ERRRUR** |
| 486 | 000034 | TRAPVEC | 34 | ;;"TRAP" TRAP |

```

487          000F60          TRVEC= 60          ;TTY KEYBOARD VECTOR
488          000064          TPVEC= 64          ;TTY PRINTER VECTOR
489          000260          PIRGVFC=240        ;PROGRAM INTERRUPT REQUEST VECTOR
490          000000          .SP
491          .SBTTL TRAP CATCHER
492
493          .SP
494          ;=ALL UNUSED LOCATIONS FROM 4 = 776 CONTAIN A ",+2,MALT"
495          ;=SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
496          ;=LOCATION # CONTAINS # TO CATCH IMPROPERLY LOADED VECTORS
497          000174          .=174
498          000000          DISPRFG; .WORD 0          ;SOFTWARE DISPLAY REGISTER
499          000176          000000          SWREG; .WORD 0          ;SOFTWARE SWITCH REGISTER
500          000200          .=200
501          000200          005067          000624          CLP          TYP0UT
502          000204          000167          000666          JNF          START
503          000210          012767          000001          000612          MOV          01,TYP0UT
504          000216          000167          000654          JMP          START
505
506          177776          PS=177776
507          000034          TRAPVEC=34
508
509          .=1000
510          001000          177570          SWF;          177570
511          001002          177570          DISPLAY;          177570
512          001004          173000          MONSA1;          173000
513          001006          001000          DATLN1;          512.
514          001010          165000          MONSA2;          165000
515          001012          001000          DATLN2;          512.
516          001014          000000          XORB;          0
517          001016          000000          EXCRC;          0
518          001020          000000          EXLPC;          0
519          001022          000000          ACTCRC;          0
520          001024          000000          ACTLPC;          0
521          001026          000000          PAHCNT;          0
522          001030          000000          TYP0UT;          0
523
524          .SBTTL ACT11 HOOKS
525
526          ;;=====
527          ;HOOKS REQUIRED BY ACT11
528          001032          000F60          ;SAVE PC
529          000F46          .=46
530          000046          022460          ;1)SET LOC.46 TO ADDRESS OF SENDAD IN ,SLOP
531          000F52          .=52
532          000052          000000          .WORD          0          ;2)SET LOC.52 TO ZERO
533          001032          000F60          ;RESTORE PC
534          .SBTTL APT PARAMETER BLOCK
535
536          ;;=====
537          ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
538          ;;=====
539          001032          .8X=          ;GIVE CURRENT LOCATION
540          000024          .=24          ;SET POWER FAIL TO POINT TO START OF PROGRAM
541          000024          000200          200          ;FOR APT START UP
542          000044          .=44          ;POINT TO APT INDIRECT ADDRESS PTRH.

```

K1

```
543 000044 001032          SAPHDR  ;;POINT TO APT HEADER BLOCK
544          001032          .B,8X  ;;RESET LOCATION COUNTER
545          ;;*****
546          ;;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
547          ;;INTERFACE SPEC.
548
549 001032          SAPHDR:
550 001032 000000          SHIRTS: .WORD 0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
551 001034 001046          SMBADR: .WORD 8MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
552 001036 000002          STSTM: .WORD 2          ;;RUN TIME OF LONGEST TEST
553 001040 000002          SPASTM: .WORD 2          ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
554 001042 000000          SUNITM: .WORD 0          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
555 001044 000014          .WORD 8ETEND=8MAIL/2 ;;LENGTH MAILBOX=ETABLE(=WORDS)
556          .SBTTL APT MAILBOX=ETABLE
557
558          ;;*****
559          .EVEN
560 001046          SMAIL:          ;;APT MAILBOX
561 001046 000000          SMSGTY: .WORD 4MSGTY  ;;MESSAGE TYPE CODE
562 001050 000000          SFATAL: .WORD AFATAL  ;;FATAL ERROR NUMBER
563 001052 000000          STESTN: .WORD ATESTN  ;;TEST NUMBER
564 001054 000000          SPASS: .WORD APASS  ;;PASS COUNT
565 001056 000000          SDEVCT: .WORD ADEVCT  ;;DEVICE COUNT
566 001060 000000          SUNIT: .WORD AUNIT  ;;I/O UNIT NUMBER
567 001062 000000          SMSGADR: .WORD AMSGADR  ;;MESSAGE ADDRESS
568 001064 000000          SMSGLG: .WORD AMSGLG  ;;MESSAGE LENGTH
569 001066          SEIABLE:          ;;APT ENVIRONMENT TABLE
570 001066          SENV: .BYTE AENV  ;;ENVIRONMENT BYTE
571 001067          SENVM: .BYTE AENVM  ;;ENVIRONMENT MODE BITS
572 001070 000000          SSWREG: .WORD ASWREG  ;;APT SWITCH REGISTER
573 001072 000000          SUBNR: .WORD AUSNR  ;;USER SWITCHES
574 001074 000000          SCPUOPT: .WORD ACPUOPT  ;;CPU TYPE, OPTIONS
575          ;0          BITS 15-11=CPL TYPE
576          ;0          11/04001,11/05002,11/20003,11/40004,11/45005
577          ;0          11/70006,PLUG07,0010
578          ;0          BIT 10=REAL TIME CLOCK
579          ;0          BIT 9=FLOATING POINT PROCESSOR
580          ;0          BIT 8=MEMORY MANAGEMENT
581 001076          SETEND:
582          .EXIT
583
584 001076 005067 177746          START: CLH SFATAL          ;CLEAR ERROR NO.
585 001102 005067 177740          CLH SMSGTYP          ;CLEAR MESSAGE TYPE (APT)
586 001106 012767 000001 177736          MOV #1,STESTN          ;SET TEST NO.
587          .SBTTL INITIALIZE THE COMMON TAGS
588 001114 012706 000500          MOV #50H,SP          ;;SETUP THE STACK POINTER
589          ;;INITIALIZE A FEW VECTORS
590 001120 012737 004570 000034          MOV #STRAP,0STRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
591 001126 012737 000340 000036          MOV #340,0STRAPVEC+2;LEVEL 7
592 001134 012737 004412 000024          MOV #0SFWR0H,00SPWRVEC ;;POWER FAILURE VECTOR
593 001142 012737 000340 000026          MOV #340,00SPWRVEC+2 ;;LEVEL 7
594          ;;SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
595          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
596 001150 013746 000004          MOV #0ERRVEC,-(SP) ;;SAVE ERROR VECTOR
597 001154 012737 001210 000004          MOV #640,00ERRVEC ;;SET UP ERROR VECTOR
598 001162 012767 177570 177610          MOV #0DSNR,SNR          ;;SETUP FOR A HARDWARE SWICH REGISTER
```

INITIALIZE THE COMMON TAGS

SL0 0013

```

599 001170 012767 177574 177604      MOV      0DISP,DISPLAY ;AND A HARDWARE DISPLAY REGISTER
600 001176 022777 177777 177574      CMP      0=1,0BWR     ;TRY TO REFERENCE HARDWARE SWR
601 001204 001P12                      FNE      668          ;BRANCH IF NO TIMEOUT TRAP OCCURRED
602                                BR        658          ;AND THE HARDWARE SWP IS NOT 0=1
603 001206 000403                      BR        658          ;BRANCH IF NO TIMEOUT
604 001210 012716 001216      648:    MOV      0658,(SP)    ;SET UP FOR TRAP RETURN
605 001214 000P02                      RTI
606 001216 012767 000176 177554 658:    MOV      0SWPEG,SWR   ;POINT TO SOFTWARE SWP
607 001224 012767 000174 177550      MOV      0DISPEG,DISPLAY
608 001232 012637 000P04      668:    MOV      (SP)+,002P4VEC ;RESTORE ERROR VECTOR
609
610 001236 005P67 177612      CLF      0PASS        ;CLEAR PASS COUNT
611 001242 132767 000200 177617      BITB     0APTSIZE,0ENVM ;IFST USEL SIZE UNDER APT
612 001250 001403                      BEQ      678          ;YES,USE NON-APT SWITCH
613 001252 012767 001070 177520      MOV      008WREG,SWR ;NO,USE APT SWITCH REGISTER
614 001260      678:
615 001260 026727 176556 002060      CMP      42,00PNDAD   ;ACT AUTO MODE?
616 001266 001402                      BEQ      681          ;YIF 80: ON
617 001270 104401                      TYPE
618 001272 005260                      TITL
619
620 001274 123727 000176 000006 681:    RESTR1: CMPB     00176,06   ;IS SOFTWARE SWITCH REGISTER 007
621 001302 001003                      BNE     181          ;IF NOT, TEST NORMALY
622 001304 012767 000176 177466      MOV      0SWPEG,SWR   ;IF 80 USE THE SOFTWARE SWITCH REG
623 001312 017700 177462      181:    MOV      0SWP,00
624 001316 001424                      BEQ     GETIN        ;GLT SWR
625 001320 006300      8T2:    ASL      R0          ;IF ZERO: GLT INPUT
626 001322 016067 004712 177470      MOV      TXLPC(R0),EXLPC ;FETCH EXPECT, LPC
627 001330 016067 004644 177460      MOV      TXCRC(R0),EXCRC ;FETCH EXPECTED CRC
628 001336 016067 004762 177462      MOV      TDLN1(R0),DATLN1 ;FETCH 1ST LENGTH
629 001344 016067 005P30 177432      MOV      TRMSA1(R0),ROMSA1 ;FETCH 1ST STARTING ADDR,
630 001352 016067 005076 177432      MOV      TDLN2(R0),DATLN2 ;FEICH 2ND LENGTH
631 001360 016067 005144 177422      MOV      TRMSA2(R0),ROMSA2 ;FETCH 2ND STARTING ADDR
632 001366 000450                      BR      CHECK        ;GO COMPUTE
633 001370 005767 176446      6T1:    GETIN: TST      42
634 001374 001P45                      BNE     CHECK        ;UNDER ACI AUT ACCEP?
635 001376 122767 000001 177462      CMPB     01,0ENV     ;IF 80: BK USE DEFAULT PARAMETERS
636 001404 001461                      BEQ     CHECK        ;UNDER APT?
637 001406 005767 177416      TST      TYP0UT      ;IF 80: BK
638 001412 001P12                      BNE     GET2         ;FROM TYPE OPTION
639 001414 104401                      TYPE
640 001416 005273                      GETCRC
641 001420 104407                      RDOCT
642 001422 012067 177370      MOV      (SP)+,EXCRC  ;STORE EXPECT, CPC
643 001426 104401                      TYPE                ;TYPE LPC INPUT REQUEST
644 001430 005317                      GETLPC
645 001432 104407                      RDOCT
646 001434 012067 177360      MOV      (SP)+,EXLPC  ;STORE EXPECTED LPC
647 001440 104401      6T2:    GET2:  TYPE
648 001442 005477                      SA1                ;REQUEST 1ST ADDRESS SPACE
649 001444 104407                      RDOCT              ;INPUT SA
650 001446 012067 177332      MOV      (SP)+,ROMSA1
651 001452 104401                      TYPE                ;REQUEST LENGTH OF 1ST ADDR. SPACE
652 001454 005637                      SIZE1
653 001456 104407                      RDOCT              ;INPUT LENGTH
654 001460 012067 177322      MOV      (SP)+,DATLN1

```

M1

| Line | Address | Code | Label | Op | Opnd | Comment |
|------|---------|--------|---------------|--------|--------------|--|
| 655 | 001664 | 104401 | | TYPE | | ;REQUEST START ADDR. FOR 2ND SPACE |
| 656 | 001666 | 005557 | | BA2 | | |
| 657 | 001670 | 104407 | | MOVT | | ;INPUT SA |
| 658 | 001672 | 012067 | 177312 | MOV | (SP)+,R0MSA2 | |
| 659 | 001676 | 104401 | | TYPE | | ;REQUEST LENGTH OF 2ND SPACE |
| 660 | 001680 | 005717 | | SIZE2 | | |
| 661 | 001682 | 104407 | | MOVT | | ;INPUT LENGTH |
| 662 | 001684 | 012067 | 177302 | MOV | (SP)+,DATIN2 | |
| 663 | 001688 | 005767 | 177310 | IST | TYPCUP | ;IS TYP0UT REQUESTED? |
| 664 | 001692 | 001402 | | LEU | 18 | ;BRANCH IF NOT |
| 665 | 001696 | 000167 | 000662 | JMP | TYPROV | ;GO TYPE 001 ROM |
| 666 | 001698 | 005467 | 177274 | 181 | ACTCRC | ;CLEAR STORAGE FOR ACTUAL CRC |
| 667 | 001700 | 005467 | 177272 | CIN | ACTLPC | ;CLEAR STORAGE FOR ACTUAL LPC |
| 668 | 001702 | 010700 | 177280 | MOV | DATLN1,R0 | ;SET LENGTH OF 1ST ROM SPACE |
| 669 | 001704 | 001411 | | REG | CH1 | ;IF NO VERSION SELECTED, BR |
| 670 | 001706 | 010701 | 177240 | MOV | R0MSA1,R1 | ;POINT TO START OF 1ST ROM SPACE |
| 671 | 001708 | 000767 | 000320 | JSP | PC,CRC | ;COMPUTE FIRST HALF OF CRC |
| 672 | 001710 | 010701 | 177230 | MOV | R0MSA1,R1 | ;POINT TO START OF 1ST ROM ADDR. |
| 673 | 001712 | 010701 | 177226 | MOV | DATLN1,R0 | ;SET LENGTH OF 1ST ROM ADDR. |
| 674 | 001714 | 000200 | | ASH | FA | ;CONVERT TO WORDS |
| 675 | 001716 | 000767 | 000556 | JSP | PC,LPC | ;COMPUTE FIRST HALF OF LPC |
| 676 | 001718 | 010701 | 177210 | MOV | R0MSA2,R1 | ;POINT TO 2ND ROM ADDR. |
| 677 | 001720 | 010700 | 177210 | MOV | DATLN2,R0 | ;SET LENGTH OF 2ND ROM ADDR. |
| 678 | 001722 | 001422 | | REG | CH1 | ;BR IF THIS SPACE NOT USED |
| 679 | 001724 | 000767 | 000271 | JSP | PC,CRC | ;COMPUTE REMAINDER OF CRC |
| 680 | 001726 | 010701 | 177200 | MOV | R0MSA2,R1 | ;POINT TO START OF 2ND ROM ADDR. |
| 681 | 001728 | 010700 | 177176 | MOV | DATLN2,R0 | ;SET LENGTH OF 2ND ROM ADDR. |
| 682 | 001730 | 000200 | | ASH | RA | ;CONVERT TO WORDS |
| 683 | 001732 | 000767 | 000522 | JSP | PC,LPC | ;COMPUTE REMAINDER OF LPC |
| 684 | 001734 | 122767 | 000001 177230 | CMPR | 01,SEN4 | ;UNDER APT? |
| 685 | 001736 | 001403 | | REG | 18 | ;IF 001 BR |
| 686 | 001738 | 005767 | 170200 | IST | 42 | ;UNDER ACT AUTO ACCEPT? |
| 687 | 001740 | 001402 | | REG | CH1 | ;IF NOT1 BR |
| 688 | 001742 | 000167 | 000656 | 181 | JMP | AUTACT |
| 689 | 001744 | 020767 | 177100 177150 | CH11 | CMF | ;COMPUTED = EXPECTED ? |
| 690 | 001746 | 001431 | | REG | CH1 | ;IF 001 BR |
| 691 | | | | | | |
| 692 | 001654 | 104401 | | TYPE | | ;TYPE CRC ERROR MESSG. |
| 693 | 001656 | 005343 | | EXCRPG | | |
| 694 | 001660 | 010700 | 177132 | MOV | EXCRC,=(SP) | ;PUT EXPECT CRC ON STACK |
| 695 | 001664 | 104402 | | TYPOC | | ;TYPE EXPECTED CRC |
| 696 | 001666 | 104401 | | TYPE | | ;TYPE ACTUAL CRC MESSG |
| 697 | 001670 | 005412 | | ACCRMG | | |
| 698 | 001672 | 010700 | 177120 | MOV | ACTCRC,=(SP) | ;PUT ACTUAL CRC ON STACK |
| 699 | 001676 | 104402 | | TYPOC | | ;TYPE ACTUAL CRC |
| 700 | 001700 | 020727 | 170136 002060 | CMF | 02,08FNDA0 | ;UNDER ACT AUTO MOVE? |
| 701 | 001706 | 001400 | | REG | 18 | ;IF 001 BR |
| 702 | 001710 | 122767 | 000001 177150 | CMPR | 01,SEN4 | ;UNDER APT? |
| 703 | 001716 | 001007 | | RNE | CH1 | ;IF NOT1 BR |
| 704 | 001720 | 012767 | 000002 177122 | 181 | MOV | 02,0FATAL |
| 705 | 001726 | 012767 | 000001 177112 | MOV | 01,04SGTIP | ;MOVE TO MAILBOX ERROR NO. 0000 2 0000 |
| 706 | 001730 | 000000 | | HALT | | ;SET MAILBOX FOR FATAL ERROR |
| 707 | | | | | | ;CRC ERROR |
| 708 | 001736 | 020767 | 177062 177054 | CMF | ACTLPC,FXLPC | ;COMPARE EXPT, LPC=ACTUAL LPC |
| 709 | 001740 | 001431 | | REG | CH2 | ;IF 001 BR |
| 710 | 001746 | 104401 | | TYPE | | ;TYPE LPC ERROR MESSG. |

| | | | | | | | | | |
|-----|--------|--------|--------|--------|----------|--------|--------------|--|--|
| 711 | 001750 | 005366 | | | | EXLPMG | | | |
| 712 | 001752 | 016746 | 177042 | | | MOV | EXLPC,=(SP) | | ;PUT EXPECTED LPC ON STACK |
| 713 | 001756 | 104402 | | | | TYFDC | | | ;TYPE EXPECTED LPC |
| 714 | 001760 | 104401 | | | | TYPE | | | ;TYPE ACTUAL LPC MESSG. |
| 715 | 001762 | 005435 | | | | ACLPMG | | | |
| 716 | 001766 | 016746 | 177034 | | | MOV | ACTLPC,=(SP) | | ;PUT ACTUAL LPC ON STACK |
| 717 | 001770 | 104402 | | | | TYFDC | | | ;TYPE ACTUAL LPC |
| 718 | 001772 | 026727 | 176044 | 002060 | | CMP | 42,00ENDAD | | ;UNDEM ACT AUTO MODE? |
| 719 | 002000 | 001404 | | | | DFC | 18 | | ;IF 00: BR |
| 720 | 002002 | 122767 | 000001 | 177056 | | CMPR | 01,0ENV | | ;UNDEM API? |
| 721 | 002010 | 001007 | | | | BNE | CF2 | | ;IF NOT: BR |
| 722 | 002012 | 012767 | 000003 | 177030 | 188 | MOV | 03,0FATAL | | ;MOVE TO MAILBOX ERROR NO. 0000] 0000 |
| 723 | 002020 | 012767 | 000001 | 177020 | | MOV | 01,0MGTIP | | ;SET MAILBOX FOR FATAL ERROR |
| 724 | 002026 | 000000 | | | | HALT | | | ;LPC ERROR |
| 725 | | | | | | | | | |
| 726 | 002030 | 026727 | 176006 | 002060 | CF2: | CMP | 42,00ENDAD | | ;ACT AUTO ACCEPT? |
| 727 | 002036 | 001402 | | | | REG | 18 | | ;IF 00: BR |
| 728 | 002040 | 104401 | | | | TYPE | | | ;TYPE END OF TEST |
| 729 | 002042 | 005460 | | | | LOTST | | | |
| 730 | 002044 | 005267 | 177004 | | 188 | INC | 0PASS | | ;BUMP PASS COUNT |
| 731 | 002050 | 013700 | 000042 | | | MOV | 0042,00 | | ;CHECK API |
| 732 | 002054 | 001405 | | | | REG | GOAGIN | | ;KEEP GOING |
| 733 | 002056 | 000005 | | | | RESET | | | |
| 734 | 002060 | 001710 | | | 00ENDAD: | JSR | PC,(R0) | | ;ACT HOOKS |
| 735 | 002062 | 000260 | | | | NOP | | | |
| 736 | 002064 | 000260 | | | | NOP | | | |
| 737 | 002066 | 000260 | | | | NOP | | | |
| 738 | 002070 | 000167 | 177200 | | GOAGIN: | JMP | RESTRY | | ;DO AGAIN |
| 739 | | | | | | | | | |
| 740 | 002074 | 016767 | 176722 | 176712 | CL0: | MOV | ACTCRC,XORS | | |
| 741 | 002102 | 111104 | | | | MOVB | (R1),R4 | | ;GET CHAP. |
| 742 | 002104 | 022701 | 173024 | | | CMP | 0173024,R1 | | ;LOCATION EFFECTED BY SWITCHES |
| 743 | 002110 | 001004 | | | | BNE | CL3 | | ;IF NOT: BR |
| 744 | 002112 | 005300 | | | | DFC | R0 | | ;FIX COUNTERS |
| 745 | 002114 | 005300 | | | | DEC | R0 | | |
| 746 | 002116 | 005721 | | | | TST | (R1)+ | | ;FIX POINTER |
| 747 | 002120 | 000770 | | | | RR | CL0 | | ;CONTINUE |
| 748 | 002122 | 004767 | 000114 | | CL3: | JSP | PC,PARITY | | ;GO GET PARITY |
| 749 | 002126 | 004767 | 000166 | | | JSP | PC,XOR | | ;XOR CHAP |
| 750 | 002132 | 000261 | | | | CLC | | | |
| 751 | 002134 | 000004 | | | | ROR | R4 | | ;ROTATE ; PUB. RIGHT |
| 752 | 002136 | 103014 | | | | BCC | CL2 | | ;IF NO CARRY: BR |
| 753 | 002140 | 052704 | 000400 | | | RIS | 0400,R4 | | ;SET BIT NINE |
| 754 | 002144 | 000261 | | | | CLC | | | |
| 755 | 002146 | 010405 | | | CL1: | MOV | R4,R5 | | ;SAVE CHAN |
| 756 | 002150 | 042705 | 177703 | | | FIC | 0177703,R5 | | |
| 757 | 002154 | 005105 | | | | COM | R5 | | |
| 758 | 002156 | 042705 | 177703 | | | HIC | 0177703,R5 | | |
| 759 | 002162 | 042704 | 000074 | | | HIC | 074,R4 | | |
| 760 | 002166 | 050504 | | | | RIS | R5,R4 | | |
| 761 | 002170 | 010467 | 176620 | | CL2: | MOV | R4,XORS | | |
| 762 | 002174 | 005300 | | | | DEC | R0 | | |
| 763 | 002176 | 001402 | | | | BEO | CLLAST | | ;IF LAST CHAP.: BR |
| 764 | 002200 | 000167 | 177676 | | | JMP | CL0 | | ;GET NEXT CHAP. |
| 765 | 002204 | 016704 | 176604 | | CLLAST: | MOV | XORS,R4 | | |
| 766 | 002210 | 005167 | 176600 | | | COM | XORS | | |

| | | | | | | | | |
|-----|--------|--------|--------|--------|---------|--------|--------------|-----------------------------------|
| 767 | 002214 | 042767 | 177052 | 176572 | | PIC | 0177450,XORS | |
| 768 | 002222 | 042704 | 177727 | | | BIC | 0177727,R4 | ;COMPLEMENT ALL BUT BITS 3 & 5 |
| 769 | 002226 | 050467 | 176562 | | | R16 | R4,XORS | |
| 770 | 002232 | 016767 | 176556 | 176562 | | MOV | XORS,ACTCPC | |
| 771 | 002240 | 000207 | | | | RTS | PC | |
| 772 | 002242 | 005067 | 176560 | | PARITY: | CLR | PARCNT | ;CLEAR BIT COUNTER |
| 773 | 002246 | 012703 | 000010 | | | MOV | 010,R3 | ;SET NO. OF BITS |
| 774 | 002252 | 032704 | 000001 | | CLP0: | BIT | 01,R4 | ;SEE IF ONE BIT |
| 775 | 002256 | 001402 | | | | REQ | CLP1 | ;IF NOT, BR |
| 776 | 002260 | 005267 | 176542 | | | INC | PARCNT | ;BUMP COUNTER |
| 777 | 002264 | 000241 | | | CLP1: | CLC | | |
| 778 | 002266 | 006004 | | | | POP | R4 | ;ROTATE TO NEXT BIT |
| 779 | 002270 | 005303 | | | | DFC | R3 | |
| 780 | 002272 | 001367 | | | | BNE | CLP0 | ;CONTINUE FOR ALL BITS |
| 781 | 002274 | 112104 | | | | MOV# | (R1)+,R4 | |
| 782 | 002276 | 042704 | 177400 | | | PIC | 0177400,R4 | |
| 783 | 002302 | 032767 | 000001 | 176516 | | BIT | 01,PARCNT | ;SEE IF ODD # OF ONE BITS |
| 784 | 002310 | 001402 | | | | BNE | CLP2 | ;IF SO, BR |
| 785 | 002312 | 052704 | 000400 | | | BIS | 0400,R4 | ;SET PARITY BIT |
| 786 | 002316 | 000207 | | | CLP2: | RTS | PC | ;EXIT |
| 787 | | | | | | | | |
| 788 | 002320 | 010446 | | | XOR: | MOV | R4,=(R4) | ;XOR SUBROUTINE: R4 WITH XORS |
| 789 | 002322 | 046716 | 176466 | | | BIC | XORS,(SP) | |
| 790 | 002326 | 040467 | 176462 | | | BIC | R4,XORS | |
| 791 | 002332 | 052067 | 176456 | | | BIS | (SP)+,XORS | |
| 792 | 002336 | 016704 | 176452 | | | MOV | XORS,R4 | |
| 793 | 002342 | 000207 | | | | RTS | PC | |
| 794 | | | | | | | | |
| 795 | 002344 | 016767 | 176454 | 176442 | LPC: | MOV | ACTLPC,XORS | |
| 796 | 002352 | 012104 | | | LPC1: | MOV | (R1)+,R4 | |
| 797 | 002354 | 022701 | 173026 | | | CMF | 0173026,R1 | ;LOCATION EFFECTED BY SWITCHES |
| 798 | 002360 | 001402 | | | | REQ | LPC2 | ;IF SO, SKIP LOC. BY BRANCHING |
| 799 | 002362 | 004767 | 177732 | | | JSR | PC,XOR | |
| 800 | 002366 | 005303 | | | LPC2: | DFC | R2 | |
| 801 | 002370 | 001370 | | | | BNE | LPC1 | |
| 802 | 002372 | 016767 | 176416 | 176424 | | MOV | XORS,ACTLPC | |
| 803 | 002400 | 000207 | | | | RTS | PC | |
| 804 | | | | | | | | |
| 805 | 002402 | 104401 | | | TYPROM: | TYPE | | ;TYPE HEADER |
| 806 | 002404 | 006005 | | | | TYPHDF | | |
| 807 | 002406 | 016700 | 176372 | | | MOV | ROMSA1,R0 | ;POINT TO 1ST ROM SPACE |
| 808 | 002412 | 016701 | 176370 | | | MOV | DATLN1,R1 | ;PUT LENGTH IN R1 |
| 809 | 002416 | 006201 | | | | ASH | R1 | ;CONVERT TO WORDS |
| 810 | 002420 | 001402 | | | | BEO | TYFR1 | ;BRANCH IF 1ST ROM SPACE NOT USED |
| 811 | 002422 | 004767 | 000026 | | | JSR | PC,TYP | ;GO TYPE 1ST ADDR. SPACE |
| 812 | 002426 | 016700 | 176356 | | TYFF1: | MOV | ROMSA2,R0 | ;POINT TO 2ND ADDR. SPACE |
| 813 | 002432 | 016701 | 176354 | | | MOV | DATLN2,R1 | ;PUT LENGTH IN R1 |
| 814 | 002436 | 006201 | | | | ASH | R1 | ;CONVERT TO WORDS |
| 815 | 002440 | 001402 | | | | BEO | TYCOT | ;BR IF 2ND ADDR. SPACE NOT USED |
| 816 | 002442 | 004767 | 000006 | | | JSR | PC,TYP | ;GO TYPE 2ND ADDR. SPACE |
| 817 | 002446 | 104401 | | | ENDOT: | TYPE | | |
| 818 | 002450 | 005400 | | | | EOTST | | |
| 819 | 002452 | 000000 | | | | HALT | | |
| 820 | | | | | | | | |
| 821 | 002454 | 104401 | | | TYP: | TYPE | | |
| 822 | 002456 | 005777 | | | | CALLF | | |

INITIALIZE THE COMMON TAGS

BEG 0017

```

023 002461 000493
024 002462 032700 000003
025 002466 001000
026 002470 104401
027 002472 005777
028 002474 010040
029 002476 104402
030 002500 104401
031 002502 006002
032 002504 012000
033 002506 104402
034 002510 104401
035 002512 006002
036 002514 005301
037 002516 001361
038 002520 000207
039 002522 005000
040 002524 062700 000002
041 002530 026027 005212 177777
042 002536 001425
043 002540 026067 004644 176254
044 002546 001360
045 002550 023727 000042 002060
046 002556 001405
047 002560 016067 005212 000002
048 002566 104401
049 002570 000000
050 002572 016067 004644 176216
051 002600 016067 004712 176212
052 002606 000167 177124
053
054 002612 104401
055 002614 006000
056 002616 012767 000001 176224
057 002624 012767 000001 176214
058 002632 000000
059
060
061
062
063 002634 177560
064 002636 177562
065
066
067
068
069
070
071
072
073
074
075
076
077
078 002640 011046
  
```

```

BR TYP3
BIT 03,PC ;ADDRESS MULTIPLE OF 4?
RNF TYP2 ;IF NOT: BR
TYPE CARLF
MOV PC, -(SP) ;PUT ADDRESS ON STACK
TYP0C ;TYPE ADDR.
TYPE COLON
TYP2: MOV (R0)+, -(SP) ;PUT DATA ON STACK
TYP0C ;TYP DATA
TYPE ;TYPE 2 SPACES
SP2
DEC F1 ;FINISHED?
RNF TYP0 ;IF NOT: BR
RTS PC ;RETURN
AUCTACT: CLH R0
AUCT1: ADD 02,R0 ;BUMP VALUE INDEX
CMP TMSG(R0), 0-1 ;CHECKED ALL KNOWN VERSIONS?
BEQ AUCTER ;IF SO: BR
CMP TXCRC(R0), ACTCRC ;DOES THIS CRC AGREE?
BNE AUCT1 ;IF NOT: KEEP LOOKING
CMP 0042, 00FNDAD ;UNDER ACT AUTO ACCEPT?
BEQ AUCT3 ;IF SO: BR
MOV TMSG(PC), AUCT2 ;SET UP VERSION MESSAGE
TYPE 0
AUCT2: 0
AUCT3: MOV TXCRC(R0), EXCRC ;SET EXPECTED CRC
MOV TXLPC(R0), EXLPC ;SET EXPECTED LPC
JMP CK1 ;CHECK LPC
AUCTER: TYPE
AUCTER: AUCTER
MOV 01, 00FATAL ;MOVE TO MAILBOX ERROR NO. 0000 1 0000
MOV 01, 00MSGTYP ;SET MAILBOX FOR FATAL ERROR
HALT ;AUTO ACCEPT FAILED
  
```

```

.SBTTL ITY INPUT ROUTINE
;;*****
STKS: .WORD 177562 ;;ITY KBD STATUS
STPB: .WORD 177562 ;;ITY KBD BUFFER
.ENABL LSB
.USABL LSP
;;*****
; THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE ITY
;CALL:
; R0CHR ;;INPUT A SINGLE CHARACTER FROM THE ITY
; RETURN HERE ;;CHARACTER IS ON THE STACK
; ;;WITH PARITY BIT STRIPPED OFF
;
BRDCHR: MOV (SP), -(SP) ;;PUSH DOWN THE PC
  
```

```

079 002642 016066 000004 000002      MOV      4(SF),2(SF)      ;;SAVE THE PS
080 002650 105777 177760      18:      TSTB     08TFS          ;;WAIT FOR
081 002654 100375              RPL      18              ;;A CHARACTER
082 002656 117760 177754 000004      MOVR     08TFS,4(SF)     ;;HEAD THE TTY
083 002664 042760 177600 000004      BIC      0°C<177>,4(SP) ;;GET RID OF JUNK IF ANY
084 002672 076027 000004 000023      CMP      4(SP),073      ;;IS IT A CONTROL-S?
085 002700 001013              RNE      38              ;;BRANCH IF NO
086 002702 105777 177726      28:      TSTB     08TFS          ;;WAIT FOR A CHARACTER
087 002706 100375              BPL      28              ;;LOOP UNTIL ITS THERE
088 002710 117760 177722      MOVB     08TFS,-(SP)     ;;GET CHARACTER
089 002714 042710 177600      BIC      0°C177,(SP)     ;;MAKE IT 7-BIT ASCII
090 002720 022027 000021      CMP      (SP)+,021      ;;IS IT A CONTROL-Q?
091 002724 001360              RNE      28              ;;IF NOT DISCARD IT
092 002726 000750              RR       18              ;;YES, RESUME
093 002730 076027 000004 000140 38:      CMP      4(SP),0140     ;;IS IT UPPER CASE?
094 002736 042407              BLT      48              ;;BRANCH IF YES
095 002740 026027 000004 000175      CMP      4(SP),0175     ;;IS IT A SPECIAL CHAR?
096 002746 003003              BGT      48              ;;BRANCH IF YES
097 002750 042760 000040 000004      BIC      040,4(SP)       ;;MAKE IT UPPER CASE
098 002756 000002      48:      RTI                    ;;GO BACK TO USER
099
100
101
102
103
104
105
106 002760 010340      8PDLIN: MOV      R3,-(SP)      ;;SAVE R3
107 002762 005040      CLR      -(SP)          ;;CLEAR THE RUBOUT KEY
108 002764 012703 003214      18:      MOV      08TTYIN,R3     ;;GET ADDRESS
109 002770 022703 003224      28:      CMP      08TTYIN+0,,R3  ;;BUFFER FULL?
110 002774 101450              BLOS     48              ;;BR IF YES
111 002776 104405              HDCHP                    ;;GO READ ONE CHARACTER FROM THE TTY
112 003000 112013              MOVR     (SP)+,(R3)     ;;GET CHARACTER
113 003002 122713 000177      108:     CMPB     0177,(R3)      ;;IS IT A RUBOUT
114 003006 001022              BNE     58              ;;BR IF NO
115 003010 005710              TST     (SF)           ;;IS THIS THE FIRST RUBOUT?
116 003012 001007              BNE     68              ;;BR IF NO
117 003014 112767 000134 000170      MOVR     0'\,99        ;;TYPE A BACK SLASH
118 003022 104401 003212              TYPE     ,98
119 003026 012710 177777              MOV      0-1,(SP)      ;;SET THE RUBOUT KEY
120 003032 005303      68:      DEC      R3            ;;BACKUP BY ONE
121 003034 020327 003214      CMP      R3,08TTYIN    ;;STACK EMPTY?
122 003040 103434              BLO     48              ;;BR IF YES
123 003042 111367 000144      MOVR     (R3),08       ;;SETUP TO TYPEOUT THE DELETED CHAR.
124 003046 104401 003212              TYPE     ,98          ;;GO TYPE
125 003052 000746              BR       28            ;;GO READ ANOTHER CHAR.
126 003054 005710      58:      TST     (SF)           ;;RUBOUT KEY SET?
127 003056 001400              BEQ     78              ;;BR IF NO
128 003060 112767 000134 000124      MOVB     0'\,98        ;;TYPE A BACK SLASH
129 003066 104401 003212              TYPE     ,98
130 003072 005010              CLR     (SP)           ;;CLEAR THE RUBOUT KEY
131 003074 122713 000025      78:      CMPB     075,(R3)      ;;IS CHARACTER A CTRL UP
132 003100 001003              BNE     88              ;;BR IF NO
133 003102 104401 003230              TYPE     ,8CNTLU      ;;TYPE A CONTROL "U"
134 003106 000720              BR       18            ;;GO START OVER

```

```

935 003110 122713 000022 001 CMPB 022,(R3) ;IS CHARACTER A "M"?
936 003114 001011 001 BNE 30 ;PUNCH IF NO
937 003116 105013 001 CLRB (R3) ;CLEAR THE CHARACTER
938 003120 104401 003225 001 TYPE ,SCHLF ;TYPE A "CR" & "LF"
939 003124 104401 003214 001 TYPE ,STTYIN ;TYPE THE INPUT STRING
940 003130 000717 001 MH 20 ;GO PICKUP ANOTHER CHACTER
941 003132 104401 003224 401 TYPE ,SQUER ;TYPE A "?"
942 003136 000712 001 MH 10 ;CLEAR THE BUFFER AND LOOP
943 003140 111307 000046 301 MOVB (R3),00 ;ECHO THE CHARACTER
944 003144 104401 003212 001 TYPE ,90
945 003150 122723 000015 001 CMPB 015,(R3)+ ;CHECK FOR RETURN
946 003154 001305 001 BNE 20 ;LOOP IF NOT RETURN
947 003156 105003 177777 001 CLRB -1(R3) ;CLEAR RETURN (THE 15)
948 003162 104401 003226 001 TYPE ,81F ;TYPE A LINE FEED
949 003166 005720 001 TST (SP)+ ;CLEAN PUNOUT KEI FROM THE STACK
950 003170 012603 001 MOV (SP)+,R1 ;RESTORE R1
951 003172 011646 001 MOV (SP),-(SP) ;ADJUST THE STACK AND PUT ADDRESS OF THE
952 003174 016000 000004 000002 001 MOV 4(SP),2(SP) ; FIRST ASCII CHARACTER ON IT
953 003202 012700 003214 000004 001 MOV 0STTYIN,4(SP)
954 003210 000002 001 RTI ;RETURN
955 003212 000 901 ,BYTE 0 ;STORAGE FOR ASCII CHAR. TO TYPE
956 003213 000 ,BYTE 0 ;TERMINATOR
957 003214 000010 ,BLKB 0. ;RESERVE 6 BYTES FOR TTY INPUT
958 003224 077 SQUER: ,ASCII "?" ;QUESTION MARK
959 003225 015 SCHLF: ,ASCII <15> ;CARRIAGE RETURN
960 003226 000012 SLP: ,ASCII <12> ;LINE FEED
961 003230 052530 005015 000 SCNTL1: ,ASCII /"U"/<15><12> ;CONTROL "U"
962 003235 130 006507 000012 SCNTL2: ,ASCII /"G"/<15><12> ;CONTROL "G"
963 003242 005015 053523 020127 000 S45-R: ,ASCII <15><12>/80F = /
964 003250 020075 000
965 003253 040 047040 053505 001 SMNEW: ,ASCII / NEW = /
966 003260 036440 000040
967
968
969
970 ;*****
971 ;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
972 ;CHANGE IT TO BINARY.
973 ;THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
974 ;OCTAL DIGITS. IF AN ILLGAL CHARACTER IS READ A "?" WILL BE TYPED
975 ;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
976 ;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
977 ;CALL:
978 ; RDOCT ;READ AN OCTAL NUMBER
979 ; RETURN HERE ;LOW ORDER BITS ARE ON TOP OF THE STACK
980 ; ;HIGH ORDER BITS ARE IN SHIOCT
981 003264 011646 001 BRDOCT: MOV (SP),-(SP) ;PROVIDE SPACE FOR THE
982 003266 016000 000004 000002 001 MOV 4(SP),2(SP) ;INPUT NUMBER
983 003274 010040 001 MOV R0,-(SP) ;PUSH R0 ON STACK
984 003276 010140 001 MOV R1,-(SP) ;PUSH R1 ON STACK
985 003300 010240 001 MOV R2,-(SP) ;PUSH R2 ON STACK
986 003302 104400 101 RDLIN ;READ AN ASCII LINE
987 003304 012600 001 MOV (SP)+,R0 ;GET ADDRESS OF 1ST CHARACTER
988 003306 010067 000100 001 MOV R0,50 ;AND SAVE IT
989 003312 005001 001 CLF R1 ;CLEAR DATA WORD
990 003314 005002 001 CLR R2

```

```

991 003316 112046          28:  MOVR   (R0)+,(SP)    ;;PICKUP THIS CHARACTER
992 003320 001420          BEM    38              ;;IF ZERO GET OUT
993 003322 122716 000060  CMPR   0'0,(SP)      ;;MAKE SURE THIS CHARACTER
994 003326 003020          HGT    48              ;;IS AN OCTAL DIGIT
995 003330 122716 000067  CMPB   0'7,(SP)
996 003334 002423          HIT    48
997 003336 006301          ASL    R1              ;;02
998 003340 006102          PCL    R2
999 003342 006301          ASL    R1              ;;04
1000 003344 006102          ROL    R2
1001 003346 006301          ASL    R1              ;;08
1002 003350 006102          ROL    R2
1003 003352 042716 177770  BIC    0'07,(SP)     ;;STRIP THE ASCII JUNK
1004 003356 002001          ADD    (SP)+,R1      ;;ADD IN THIS DIGIT
1005 003360 000756          BP     28              ;;LOOP
1006 003362 005726          38:  TST    (SP)+      ;;CLEAN TERMINATOR FROM STACK
1007 003364 010100 000012  MOV    R1,12(SP)     ;;SAVE THE RESULT
1008 003370 010267 000026  MOV    R2,8(0CT)
1009 003374 012002          MOV    (SP)+,R2      ;;POP STACK INTO R2
1010 003376 012601          MOV    (SP)+,R1      ;;POP STACK INTO R1
1011 003400 012000          MOV    (SP)+,R0      ;;POP STACK INTO R0
1012 003402 000002          RTI
1013 003404 005726          48:  TST    (SP)+      ;;CLEAN PARTIAL FROM STACK
1014 003406 105010          CLR   (R0)           ;;SET A TERMINATOR
1015 003410 104401          TYPE  104401         ;;TYPE UP THRU THE BAD CHAR.
1016 003412 000000          58:  .WORD  0
1017 003414 104401 003224  TYPE  ,BQUE          ;;?" "CR" & "LF"
1018 003420 000730          BF     18              ;;TYPE AGAIN
1019 003422 000000          BHICT: .WORD  0      ;;HIGH ORDER BITS GO HERE
1020          .SBTIL TYPE FOLLOWS
1021
1022          ;;*****
1023          ;;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1024          ;;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1025          ;;NOTE1: 0NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1026          ;;NOTE2: 0FILLN CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1027          ;;NOTE3: 0FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1028          ;;
1029          ;;CALL:
1030          ;;1) USING A TRAP INSTRUCTION
1031          ;;    TYPE  ,MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCII STRING
1032          ;;OP
1033          ;;    TYPE
1034          ;;    MESADR
1035          ;;
1036
1037 003424 105767 000265 8TYPE: TSTR   0TPFLG    ;;IS THERE A TERMINAL?
1038 003430 100002          BPL   18              ;;BH IF YFS
1039 003432 000000          HALT  0              ;;HALT HERE IF NO TERMINAL
1040 003434 000430          BR    38              ;;LEAVE
1041 003436 010046          18:  MOV    R0,-(SP)      ;;SAVE R0
1042 003440 017600 000002  MOV    02(SP),R0     ;;GET ADDRESS OF ASCII STRING
1043 003444 122767 000001 175414  CMPB   0APTENV,0ENV  ;;RUNNING IN APT MODE
1044 003452 001011          BNE   028             ;;NO,GO CHECK FOR APT CONSOLE
1045 003454 132767 000100 175405  BTR   0APTSPOOL,0ENVM ;;SPOOL MESSAGE TO APT
1046 003462 001405          BEM   028             ;;NO,GO CHECK FOR CONSOLE
  
```

| | | | | | | | | |
|------|--------|--------|--------|---------|-----------|----------------|--|-----------------------------|
| 1047 | 003464 | 010067 | 000004 | | MOV | R0,610 | ;;SETUP MESSAGE ADDRESS FOR APT | |
| 1048 | 003470 | 004767 | 000232 | | JBR | PC,8ATY3 | ;;SPOOL MESSAGE TO APT | |
| 1049 | 003474 | 000000 | | 610: | WORD | 0 | ;;MESSAGE ADDRESS | |
| 1050 | 003476 | 132767 | 000040 | 175363 | BITB | 0APTCSUP,0FNVM | ;;APT CONSOLE SUPPRESSED | |
| 1051 | 003504 | 001003 | | | RNE | 608 | ;;YFS,SNIP TYPE OUT | |
| 1052 | 003506 | 112046 | | 20: | MOVB | (R0)+,-(SP) | ;;PUSH CHARACTER 10 BF TYPED ONTO STACK | |
| 1053 | 003510 | 001005 | | | RNE | 48 | ;;BF IF IT ISN'T THE TERMINATOR | |
| 1054 | 003512 | 005726 | | | TST | (SP)+ | ;;IF TERMINATOR POP IT OFF THE STACK | |
| 1055 | 003514 | 012600 | | 608: | MOV | (SP)+,R0 | ;;RESTORE R0 | |
| 1056 | 003516 | 062716 | 000002 | 30: | ADD | 02,(SP) | ;;ADJUST RETURN PC | |
| 1057 | 003522 | 000002 | | | RTI | | ;;RETURN | |
| 1058 | 003524 | 122716 | 000011 | 40: | CMPB | 0HT,(SP) | ;;BRANCH IF <HT> | |
| 1059 | 003530 | 001430 | | | REQ | 08 | | |
| 1060 | 003532 | 122716 | 000200 | | CMPB | 0CHLF,(SP) | ;;BRANCH IF NOT <CHLF> | |
| 1061 | 003536 | 001006 | | | RNE | 58 | | |
| 1062 | 003540 | 005726 | | | TST | (SP)+ | ;;POP <CH><LF> EQUIV | |
| 1063 | 003542 | 104401 | | | TYPE | | ;;TYPE A CR AND LF | |
| 1064 | 003544 | 003225 | | | 0CRLF | | | |
| 1065 | 003546 | 105067 | 000130 | | CLRB | 0CHARCNT | ;;CLEAR CHARACTER COUNT | |
| 1066 | 003552 | 000755 | | | BR | 20 | ;;GET NEXT CHARACTER | |
| 1067 | 003554 | 004767 | 000056 | 50: | JBR | PC,0TYPEC | ;;GO TYPE THIS CHARACTER | |
| 1068 | 003560 | 126726 | 000130 | 60: | CMPR | 0FILLC,(SP)+ | ;;IS IT TIME FOR FILLER CHARS.? | |
| 1069 | 003564 | 001350 | | | RNE | 20 | ;;IF NO GO GET NEXT CHAR. | |
| 1070 | 003566 | 016746 | 000120 | | MOV | 0NULL,-(SP) | ;;GET 0 OF FILLER CHARS. NEEDED | |
| 1071 | | | | | | | ;;AND THE NULL CHAR. | |
| 1072 | 003572 | 105366 | 000001 | 70: | DECB | 1(SP) | ;;DOES A NULL NEED TO BE TYPED? | |
| 1073 | 003576 | 002770 | | | HLT | 08 | ;;BR IF NO--GO POP THE NULL OFF OF STACK | |
| 1074 | 003600 | 004767 | 000032 | | JBR | PC,0TYPEC | ;;GO TYPE A NULL | |
| 1075 | 003604 | 105367 | 000072 | | DECB | 0CHARCNT | ;;DU NOT COUNT AS A COUNT | |
| 1076 | 003610 | 000770 | | | BR | 76 | ;;LOOP | |
| 1077 | | | | | | | | |
| 1078 | | | | | | | | |
| 1079 | | | | | | | | |
| 1080 | 003612 | 112716 | 000040 | 80: | MOVB | 0' ,(SP) | ;;REPLACE TAB WITH SPACE | |
| 1081 | 003616 | 004767 | 000014 | 90: | JBR | PC,0TYPEC | ;;TYPE A SPACE | |
| 1082 | 003622 | 132767 | 000007 | 000052 | BITB | 07,0CHARCNT | ;;BRANCH IF NOT AT | |
| 1083 | 003630 | 001372 | | | RNE | 98 | ;;TAB STOP | |
| 1084 | 003632 | 005726 | | | TST | (SP)+ | ;;POP SPACE OFF STACK | |
| 1085 | 003634 | 000724 | | | BR | 20 | ;;GET NEXT CHARACTER | |
| 1086 | 003636 | 105777 | 000044 | 0TYPEC: | TSTB | 0STPS | ;;WAIT UNTIL PRINTER IS READY | |
| 1087 | 003642 | 100375 | | | RPL | 0TYPEC | | |
| 1088 | 003644 | 116677 | 000002 | 000036 | MOVB | 2(SP),0STPB | ;;LOAD CHAR TO BE TYPED INTO DATA REG. | |
| 1089 | 003652 | 122766 | 000015 | 000002 | CMPB | 0CR,2(SP) | ;;IS CHARACTER A CARRIAGE RETURN? | |
| 1090 | 003660 | 001003 | | | RNE | 18 | ;;BRANCH IF NO | |
| 1091 | 003662 | 105067 | 000014 | | CLRB | 0CHARCNT | ;;YFS--CLEAR CHARACTER COUNT | |
| 1092 | 003666 | 000406 | | | BR | 0TYPEX | ;;EXIT | |
| 1093 | 003670 | 122766 | 000012 | 000002 | 10: | CMPB | 0LF,2(SP) | ;;IS CHARACTER A LINE FEED? |
| 1094 | 003676 | 001402 | | | REQ | 0TYPEX | ;;BRANCH IF YES | |
| 1095 | 003700 | 105727 | | | INCR | (PC)+ | ;;COUNT THE CHARACTER | |
| 1096 | 003702 | 000000 | | | 0CHARCNT: | WORD | 0 | |
| 1097 | 003704 | 000207 | | | 0TYPEX: | PTS | PC | |
| 1098 | | | | | | | | |
| 1099 | 003706 | 177564 | | | 0STPS: | WORD | 177564 | |
| 1100 | 003710 | 177566 | | | 0STPB: | WORD | 177566 | |
| 1101 | 003712 | 000 | | | 0NULL: | BYTE | 0 | |
| 1102 | 003714 | 002 | | | 0FILLS: | BYTE | 2 | |

;;TTY PRINTER STATUS REG. ADDRESS
 ;;TTY PRINTER BUFFER REG. ADDRESS
 ;;CONTAINS NULL CHARACTER FOR FILLS
 ;;CONTAINS 0 OF FILLER CHARACTERS REQUIRED


```

1103 003714 012 SFILLC: .BYTE 12 ;INSERT FILL CHARS. AFTER A "LINE FEED"
1104 003715 000 STPLFC: .BYTE 0 ;"TERMINAL AVAILABLE" FLAG (HIT<07>=00YLS)
1105 .SBTTL APT COMMUNICATIONS ROUTINE
1106
1107 ;*****
1108 003716 112767 000001 000236 SATY1: MOVB 01,SFFLU ;ITU REPORT FATAL ERROR
1109 003724 112767 000001 000226 SATY3: MOVB 01,SFPLG ;IO TYPE A MESSAGE
1110 003732 000403 BR SATYC
1111 003734 112767 000001 000220 SATY4: MOVB 01,SFFLG ;ITO ONLY REPORT FATAL ERPOP
1112 003742 SATYC:
1113 003742 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
1114 003744 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
1115 003746 105767 000206 TSTB SMFLG ;SHOULD TYPE A MESSAGE?
1116 003752 001450 BEQ 50 ;IF NOT: BR
1117 003754 122767 000001 175104 CMPP 0APTENV,SENV ;OPERATING UNDER APT?
1118 003762 001031 BNE 30 ;IF NOT: BR
1119 003764 132767 000100 175075 BITE 0APTSPOOL,SENVH ;SHOULD SPOOL MESSAGES?
1120 003772 001425 BEQ 30 ;IF NOT: BR
1121 003774 017600 000004 MOV 04(SP),R0 ;GET MESSAGE ADDR.
1122 004000 062766 000002 000004 ADD 02,4(SP) ;BUMP RETURN ADDR.
1123 004006 005767 175034 101 TST 0MSGTYPE ;SEE IF DONE w/ LAST XMISSION?
1124 004012 001375 BNE 10 ;IF NOT: WAIT
1125 004014 010067 175042 MOV R0,0MSGAD ;PUT ADDR IN MAILBOX
1126 004020 105720 201 TSTB (R0)+ ;FIND END OF MESSAGE
1127 004022 001376 BNE 20
1128 004024 106700 175032 SUB 0MSGAD,R0 ;SUB START OF MESSAGE
1129 004030 006200 ASR R0 ;GET MESSAGE LGTH IN WORDS
1130 004032 010067 175026 MOV R0,0MSGLGT ;PUT LENGTH IN MAILBOX
1131 004036 012767 000004 175002 MOV 04,0MSGTYPE ;TELL APT TO TAKE MSG.
1132 004044 000413 BR 50
1133 004046 017667 000004 000216 301 MOV 04(SP),40 ;PUT MSG ADDR IN JSR LINKAGE
1134 004054 062766 000002 000004 ADD 02,4(SP) ;BUMP RETURN ADDRESS
1135 004062 016746 173710 MOV 177776,-(SP) ;PUSH 177776 ON STACK
1136 004066 004767 177332 JSR PC,0TYPE ;CALL TYPE MACRO
1137 004072 000000 401 .WORD 0
1138 004074 501
1139 004074 105767 000062 1001 TSTB 0FFLG ;SHOULD REPORT FATAL ERROR?
1140 004100 001416 BEQ 120 ;IF NOT: BR
1141 004102 005767 174760 TST SENV ;RUNNING UNDER APT?
1142 004106 001413 BEQ 120 ;IF NOT: BR
1143 004110 005767 174732 1101 TST 0MSGTYPEF ;FINISHED LAST MESSAGE?
1144 004114 001375 BNE 110 ;IF NOT: WAIT
1145 004116 017667 000004 174724 MOV 04(SP),0FATAL ;GET ERROR 0
1146 004124 062766 000002 000204 ADD 02,4(SP) ;BUMP RETURN ADDR.
1147 004132 005267 174710 INC 0MSGTYPE ;TELL APT TO TAKE BRPOP
1148 004136 105067 000020 1201 CLPB 0FFLG ;CLEAN FATAL FLAG
1149 004142 105067 000013 CLR0 0LFLG ;CLEAN LOG FLAG
1150 004146 105067 000006 CLR0 0MFLG ;CLEAN MESSAGE FLAG
1151 004152 012601 MOV (SP)+,P1 ;POP STACK INTO R1
1152 004154 012600 MOV (SP)+,R0 ;POP STACK INTO R0
1153 004156 000207 RTS PC ;RETURN
1154 004160 000 0MFLG: .BYTE 0 ;MESSG. FLAG
1155 004161 000 0LFLG: .BYTE 0 ;LOG FLAG
1156 004162 000 0FFLG: .BYTE 0 ;FATAL FLAG
1157 004164 .FVEN
1158 000200 APTSIZE=200
  
```

```

1159      000001      APTENVR001
1160      000100      APTSPOL0100
1161      000040      APICUP0040
1162      .RHTTL      BINARY TO OCTAL (ASC AND TYPE
1163
1164      ;;*****
1165      ;;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
1166      ;;OCTAL (ASCII) NUMBER AND TYPE IT.
1167      ;;STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1168      ;;CALL:
1169      ;;      MOV      NUM,=(SP)      ;;NUMBER TO BE TYPED
1170      ;;      TYPOS      ;;CALL FOR TYPEOUT
1171      ;;      .RYTE      N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1172      ;;      .RYTE      M      ;;N=1 OR 0
1173      ;;
1174      ;;      ;;TYPE LEADING ZEROS
1175      ;;      ;;SUPPRESS LEADING ZEROS
1176      ;;STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1177      ;;STYPOS OR STYPOC
1178      ;;CALL:
1179      ;;      MOV      NUM,=(SP)      ;;NUMBER TO BE TYPED
1180      ;;      TYPON      ;;CALL FOR TYPEOUT
1181      ;;
1182      ;;STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1183      ;;CALL:
1184      ;;      MOV      NUM,=(SP)      ;;NUMBER TO BE TYPED
1185      ;;      TYPOC      ;;CALL FOR TYPEOUT
1186
1187      004164 017646 000000      STYPOSI MOV      0(SP),=(SP)      ;;PICKUP THE MODE
1188      004170 116667 000001 000211      MOV      1(SP),80FILL      ;;LOAD ZERO FILL SWITCH
1189      004176 112667 000207      MOV      (SP)+,80MODE+1      ;;NUMBER OF DIGITS TO TYPE
1190      004202 062716 000002      AND      02,(SP)      ;;ADJUST RETURN ADDRESS
1191      004206 000406      BR      STYFON
1192      004210 112767 000001 000171      STYPOC: MOV      01,80FILL      ;;SET THE ZERO FILL SWITCH
1193      004216 112767 000006 002165      MOV      06,80MODE+1      ;;SET FOR SIX(6) DIGITS
1194      004224 112767 000005 000154      STYPON: MOV      05,80CNT      ;;SET THE ITERATION COUNT
1195      004232 010346      MOV      R3,=(SP)      ;;SAVE R3
1196      004234 010446      MOV      R4,=(SP)      ;;SAVE R4
1197      004236 010546      MOV      R5,=(SP)      ;;SAVE R5
1198      004240 116704 000145      MOV      80MODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
1199      004244 005404      NEG      R4
1200      004246 062704 000006      ADD      06,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
1201      004252 110467 000132      MOV      R4,80MODE      ;;SAVE IT FOR USE
1202      004250 116704 000125      MOV      80FILL,R4      ;;GET THE ZERO FILL SWITCH
1203      004262 010605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
1204      004266 005003      CLR      R3      ;;CLEAN THE OUTPUT WORD
1205      004270 006105      ROL      R5      ;;ROTATE MSB INTO "C"
1206      004272 000404      BR      30      ;;GO DO MSB
1207      004274 006105      ROL      R5      ;;FORM THIS DIGIT
1208      004276 006105      ROL      R5
1209      004300 006105      ROL      R5
1210      004302 010523      MOV      R5,R3
1211      004304 006143      ROL      R3      ;;GET LSH OF THIS DIGIT
1212      004306 105367 000076      DECB      80MODE      ;;TYPE THIS DIGIT?
1213      004312 100016      BPL      70      ;;BH IF NO
1214      004314 042703 177770      BIC      0177770,R3      ;;GET RID OF JUNK
  
```

```

1215 004320 001002           HNE      48           ;;TEST FOR 0
1216 004322 005704           TST      R4           ;;SUPPRESS THIS 0?
1217 004324 001403           BEQ      58           ;;BT IF YES
1218 004326 005204           INC      F4           ;;DON'T SUPPRESS ANYMORE 0'S
1219 004330 052703 000060     HIS      0',R3       ;;MAKE THIS DIGIT ASCII
1220 004334 052703 000060     LIS      0',M3       ;;MAKE ASCII IF NOT ALREADY
1221 004340 110367 000040     MOVB    R3,R8       ;;SAVE FOR TYPING
1222 004344 104401 004404     TYPF    ,R8         ;;GO TYPE THIS DIGIT
1223 004350 105367 000032     DECH    0CNT        ;;COUNT BY 1
1224 004354 003347           BGT      28           ;;JMP IF MORE TO DO
1225 004356 002402           BLT      68           ;;BT IF DONE
1226 004360 005204           INC      R4           ;;INSURE LAST DIGIT ISN'T A BLANK
1227 004362 000744           BP       28           ;;GO DO THE LAST DIGIT
1228 004364 012005     68:     MOV      (SP)+,R5     ;;RESTORE R5
1229 004366 012004           MOV      (SP)+,R4     ;;RESTORE R4
1230 004370 012003           MOV      (SP)+,R3     ;;RESTORE R3
1231 004372 016006 000002 000004     MOV      2(SP),4(SP)  ;;SET THE STACK FOR RETURNING
1232 004400 012016           MOV      (SP)+,(SP)
1233 004402 000002           RTI
1234 004404           88:     .BYTE    0           ;;STORAGE FOR ASCII DIGIT
1235 004405           .BYTE    0           ;;TERMINATOR FOR TYPE ROUTINE
1236 004406           .BYTE    0           ;;OCTAL DIGIT COUNTER
1237 004407           .BYTE    0           ;;ZERO FILL SWITCH
1238 004410 000000     80MODE: .WORD    0           ;;NUMBER OF DIGITS TO TYPE
1239
1240
1241
1242           ;;.....
1243           ;POWER DOWN ROUTINE
1243 004412 012737 004552 000024     SPWRDN: MOV      0BILLUP,0SPWRVEC ;;SET FOR FAST UP
1244 004420 012737 000340 000026     MOV      0340,0SPWRVEC+2 ;;PRIORITY
1245 004426 010040           MOV      R0,-(SP)     ;;PUSH R0 ON STACK
1246 004430 010146           MOV      R1,-(SP)     ;;PUSH R1 ON STACK
1247 004432 010246           MOV      R2,-(SP)     ;;PUSH R2 ON STACK
1248 004434 010346           MOV      R3,-(SP)     ;;PUSH R3 ON STACK
1249 004436 010446           MOV      R4,-(SP)     ;;PUSH R4 ON STACK
1250 004440 010546           MOV      R5,-(SP)     ;;PUSH R5 ON STACK
1251 004442 017740 174332     MOV      0BWR,-(SP)   ;;PUSH 0BWR ON STACK
1252 004446 010067 000104     MOV      SP,0SAVR6    ;;SAVE SP
1253 004452 012737 004464 000024     MOV      0SPWRUP,0SPWRVEC ;;SET UP VECTOR
1254 004460 000000           HALT
1255 004462 000770           BR       0=2         ;;HANG UP
1256
1257           ;;.....
1258           ;POWER UP ROUTINE
1259 004464 012737 004552 000024     SPWRUP: MOV      0BILLUP,0SPWRVEC ;;SET FOR FAST DOWN
1260 004472 016706 000060     MOV      0SAVR6,SP    ;;GET SP
1261 004476 005067 000054           CLR      0SAVR6       ;;WAIT LOOP FOR THE TTY
1262 004502 005267 000050     18:     INC      0SAVR6       ;;WAIT FOR THE INC
1263 004506 001375           BNE      18           ;;OF WORD
1264 004510 012077 174264     MOV      (SP)+,0BWR   ;;POP STACK INTO 0BWR
1265 004514 012005           MOV      (SP)+,R5     ;;POP STACK INTO R5
1266 004516 012004           MOV      (SP)+,R4     ;;POP STACK INTO R4
1267 004520 012003           MOV      (SP)+,R3     ;;POP STACK INTO R3
1268 004522 012002           MOV      (SP)+,R2     ;;POP STACK INTO R2
1269 004524 012001           MOV      (SP)+,R1     ;;POP STACK INTO R1
1270 004526 012000           MOV      (SP)+,R0     ;;POP STACK INTO R0

```

```

1271 004530 012737 004012 000024      MOV      0SPWRN,000NRVEC  ;;SET UP THE POWER DOWN VECTOR
1272 004536 012737 000300 000024      MOV      0340,000NRVEC+2 ;;PPIO:7
1273 004544 104401                      TYPE                                ;REPORT THE POWER FAILURE
1274 004546 004563 0PWRMCG: 0000 0POWER          ;POWER FAIL MESSAGE POINTER
1275 004550 000002                      RTI
1276 004552 000000 0ILLUP: 0MALT                      ;THE POWER UP SEQUENCE WAS STARTED
1277 004554 000776 0P      0=2                      ;BEFORE THE POWER DOWN WAS COMPLETE
1278 004556 000000 0SAVR6: 0                                ;PUT THE SP HERE
1279 004560 005015 047520 042527 0POWER: 0ASCIZ <15><12>"POWER"
1280 004566 000122
1281
1282      .EVEN
1283      .SBTTL TRAP DECODER
1284
1285      ;*****
1286      ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1287      ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1288      ;OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
1289      ;GO TO THAT ROUTINE.
1290 004570 010046 000002 000002 0TRAP:  MOV      R0,-(SP)          ;SAVE R0
1291 004572 016000 000002 000002 0TRAP:  MOV      2(SP),R0          ;GET TRAP ADDRESS
1292 004576 005740 000002 000002 0TRAP:  ISL      -(R0)           ;BACKUP BY 2
1293 004600 111900 000002 000002 0TRAP:  MOV8     (R0),R0          ;GET RIGHT BYTE OF TRAP
1294 004602 006300 000002 000002 0TRAP:  ASL      R0             ;POSITION FOR INDEXING
1295 004604 016000 004624 000002 0TRAP:  MOV      0TRPAB(R0),R0    ;INDEX TO TABLE
1296 004610 000200 000002 000002 0TRAP:  RTS      R0             ;GO TO ROUTINE
1297
1298
1299      ;THIS IS USE TO HANDLE THE "GETPRI" MACHO
1300
1301 004612 011046 000004 000002 0TRAP2: MOV      (SP),-(SP)       ;MOVE THE PC DOWN
1302 004614 016000 000004 000002 0TRAP2: MOV      4(SP),2(SP)      ;MOVE THE PSW DOWN
1303 004622 000002 000002 000002 0TRAP2: RTI
1304
1305      .SBTTL TRAP TABLE
1306
1307      ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1308      ;BY THE "TRAP" INSTRUCTION.
1309
1310      ;
1311      ;
1312 004624 004612 000002 000002 0TRPAB: 0WORD 0TRAP2          ;CALLTYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
1313 004626 003424 000002 000002 0TRPAB: 0TYPE  ;;CALLTYPE TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
1314 004630 000210 000002 000002 0TRPAB: 0TYPOC ;;CALLTYPOC TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
1315 004632 004104 000002 000002 0TRPAB: 0TYPOS ;;CALLTYPOS TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
1316 004634 004224 000002 000002 0TRPAB: 0STYPON ;;CALLSTYPON
1317
1318
1319 004636 002640 000002 000002 0TRPAB: 0SPDCHP ;;CALLSPDCHP TRAP+5(104405) TTY TYPEIN CHARACTER ROUTINE
1320 004640 002760 000002 000002 0TRPAB: 0SPCLIN ;;CALLSPCLIN TRAP+6(104406) TTY TYPEIN STRING ROUTINE
1321 004642 003264 000002 000002 0TRPAB: 0SPUOCT ;;CALLSPUOCT TRAP+7(104407) READ AN OCTAL NUMBER FROM TTY
1322
1323 004644 177777 000002 000002 0TRPAB: 0IXCHC: 01
1324 004646 000571 000002 000002 0TRPAB: 0571 ;M9301 = YA VERSION
1325 004650 000457 000002 000002 0TRPAB: 0457 ;M9301 = YB VERSION
1326 004652 000243 000002 000002 0TRPAB: 0243 ;M9301 = YC VERSION
  
```

| | | | |
|------|--------|--------|--------|
| 1327 | 004654 | 000635 | 635 |
| 1328 | 004656 | 000207 | 207 |
| 1329 | 004660 | 000670 | 670 |
| 1330 | 004662 | 000132 | 132 |
| 1331 | 004664 | 000374 | 374 |
| 1332 | 004666 | 000533 | 533 |
| 1333 | 004670 | 000536 | 536 |
| 1334 | 004672 | 000752 | 752 |
| 1335 | 004674 | 000633 | 633 |
| 1336 | 004676 | 177777 | -1 |
| 1337 | 004700 | 177777 | -1 |
| 1338 | 004702 | 177777 | -1 |
| 1339 | 004704 | 177777 | -1 |
| 1340 | 004706 | 177777 | -1 |
| 1341 | 004710 | 177777 | -1 |
| 1342 | | | |
| 1343 | 004712 | 177777 | -1 |
| 1344 | 004714 | 133725 | 133725 |
| 1345 | 004716 | 017563 | 17563 |
| 1346 | 004720 | 141744 | 141744 |
| 1347 | 004722 | 047613 | 47613 |
| 1348 | 004724 | 114175 | 114175 |
| 1349 | 004726 | 146126 | 146126 |
| 1350 | 004730 | 132161 | 132161 |
| 1351 | 004732 | 143466 | 143466 |
| 1352 | 004734 | 036104 | 036104 |
| 1353 | 004736 | 125411 | 125411 |
| 1354 | 004740 | 066246 | 066246 |
| 1355 | 004742 | 132367 | 132367 |
| 1356 | 004744 | 177777 | -1 |
| 1357 | 004746 | 177777 | -1 |
| 1358 | 004750 | 177777 | -1 |
| 1359 | 004752 | 177777 | -1 |
| 1360 | 004754 | 177777 | -1 |
| 1361 | 004756 | 177777 | -1 |
| 1362 | 004760 | 177777 | -1 |
| 1363 | | | |
| 1364 | 004762 | 177777 | -1 |
| 1365 | 004764 | 001000 | 1000 |
| 1366 | 004766 | 001000 | 1000 |
| 1367 | 004770 | 001000 | 1000 |
| 1368 | 004772 | 001000 | 1000 |
| 1369 | 004774 | 001000 | 1000 |
| 1370 | 004776 | 004000 | 4000 |
| 1371 | 005000 | 001000 | 1000 |
| 1372 | 005002 | 001000 | 1000 |
| 1373 | 005004 | 001000 | 1000 |
| 1374 | 005006 | 000734 | 734 |
| 1375 | 005010 | 001000 | 1000 |
| 1376 | 005012 | 001000 | 1000 |
| 1377 | 005014 | 177777 | -1 |
| 1378 | 005016 | 177777 | -1 |
| 1379 | 005020 | 177777 | -1 |
| 1380 | 005022 | 177777 | -1 |
| 1381 | 005024 | 177777 | -1 |
| 1382 | 005026 | 177777 | -1 |

TXLPC:

TDLN1:

JM9400 - YA(OR YC) VERSION
 JM9301 - YF VERSION
 JM7942 - YB VERSION
 JM9301 - YD VERSION
 JM9400 - YH (OR YK) VERSION
 JM9311 VERSION
 JM9301 - YH VERSION
 JM9301 - YE VERSION
 JM9301 - YJ VERSION

JM9301 - YA VERSION
 JM9301 - YB VERSION
 JM9301 - YC VERSION
 JM9400 - YA(OR YC) VERSION
 JM9301 - YF VERSION
 JM7942 - YB VERSION
 JM9301 - YD VERSION
 JM9400 - YH(OR YK) VERSION
 JM9311 VERSION
 JM9301 - YH VERSION
 JM9301 - YE VERSION
 JM9301 - YJ VERSION

JM9301 - YA VERSION
 JM9301 - YB VERSION
 JM9301 - YC VERSION
 JM9400 - YA(OR YC) VERSION
 JM9301 - YF VERSION
 JM7942 - YB VERSION
 JM9301 - YD VERSION
 JM9400 - YH(OR YK) VERSION
 JM9311 VERSION
 JM9301 - YH VERSION
 JM9301 - YE VERSION
 JM9301 - YJ VERSION

| | | | | | |
|------|--------|--------|---------|--------|----------------------------|
| 1383 | | | | | |
| 1384 | 005030 | 177777 | IFMSA11 | 01 | JM9301 - YA VERSION |
| 1385 | 005032 | 173000 | | 173000 | JM9301 - YH VERSION |
| 1386 | 005034 | 173000 | | 173000 | JM9301 - YC VERSION |
| 1387 | 005036 | 171000 | | 173000 | JM9400 - YA(OR YC) VERSION |
| 1388 | 005040 | 173000 | | 173000 | JM9301 - YF VERSION |
| 1389 | 005042 | 173000 | | 173000 | JM7942 - YB VERSION |
| 1390 | 005044 | 170000 | | 170000 | JM9301 - YD VERSION |
| 1391 | 005046 | 173000 | | 173000 | JM9400 - YH(OR YK) VERSION |
| 1392 | 005050 | 173000 | | 173000 | JM9311 VERSION |
| 1393 | 005052 | 163000 | | 163000 | JM9301 - YH VERSION |
| 1394 | 005054 | 173000 | | 173000 | JM9301 - YE VERSION |
| 1395 | 005056 | 173000 | | 173000 | JM9301 - YJ VERSION |
| 1396 | 005060 | 173000 | | 173000 | |
| 1397 | 005062 | 177777 | | 01 | |
| 1398 | 005064 | 177777 | | 01 | |
| 1399 | 005066 | 177777 | | 01 | |
| 1400 | 005070 | 177777 | | 01 | |
| 1401 | 005072 | 177777 | | 01 | |
| 1402 | 005074 | 177777 | | 01 | |
| 1403 | | | | | |
| 1404 | 005076 | 177777 | TECLA21 | 01 | JM9301 - YA VERSION |
| 1405 | 005100 | 001000 | | 1000 | JM9301 - YB VERSION |
| 1406 | 005122 | 001000 | | 1000 | JM9301 - YC VERSION |
| 1407 | 005124 | 001000 | | 1000 | JM9400 - YA(OR YC) VERSION |
| 1408 | 005100 | 001000 | | 1000 | JM9301 - YF VERSION |
| 1409 | 005110 | 001000 | | 0 | JM7942 - YB VERSION |
| 1410 | 005112 | 000000 | | 1000 | JM9301 - YD VERSION |
| 1411 | 005114 | 001000 | | 1000 | JM9400 - YH(OR YK) VERSION |
| 1412 | 005116 | 001000 | | 1022 | JM9311 VERSION |
| 1413 | 005120 | 001000 | | 766 | JM9301 - YH VERSION |
| 1414 | 005122 | 000764 | | 1000 | JM9301 - YE VERSION |
| 1415 | 005124 | 001000 | | 1000 | JM9301 - YJ VERSION |
| 1416 | 005126 | 001000 | | 1000 | |
| 1417 | 005130 | 177777 | | 01 | |
| 1418 | 005132 | 177777 | | 01 | |
| 1419 | 005134 | 177777 | | 01 | |
| 1420 | 005136 | 177777 | | 01 | |
| 1421 | 005140 | 177777 | | 01 | |
| 1422 | 005142 | 177777 | | 01 | |
| 1423 | | | | | |
| 1424 | 005144 | 177777 | IFMSA21 | 01 | JM9301 - YA VERSION |
| 1425 | 005146 | 165000 | | 165000 | JM9301 - YF VERSION |
| 1426 | 005150 | 165000 | | 165000 | JM9301 - YC VERSION |
| 1427 | 005152 | 165000 | | 165000 | JM9400 - YA(OR YC) VERSION |
| 1428 | 005154 | 165000 | | 165000 | JM9301 - YF VERSION |
| 1429 | 005156 | 165000 | | 0 | JM7942 - YB VERSION |
| 1430 | 005160 | 000000 | | 0 | |
| 1431 | | | | | |
| 1432 | 005162 | 165000 | | 165000 | JM9301 - YD VERSION |
| 1433 | 005164 | 165000 | | 165000 | JM9400 - YH(OR YK) VERSION |
| 1434 | 005166 | 166000 | | 166000 | JM9311 VERSION |
| 1435 | 005170 | 165000 | | 165000 | JM9301 - YH VERSION |
| 1436 | 005172 | 165000 | | 165000 | JM9301 - YE VERSION |
| 1437 | 005174 | 165000 | | 165000 | JM9301 - YJ VERSION |
| 1438 | 005176 | 177777 | | 01 | |

AS

| | | | | | |
|------|--------|--------|-------|--|----|
| 1439 | 005200 | 177777 | | | -1 |
| 1440 | 005202 | 177777 | | | -1 |
| 1441 | 005204 | 177777 | | | -1 |
| 1442 | 005206 | 177777 | | | -1 |
| 1443 | 005210 | 177777 | | | -1 |
| 1444 | | | | | |
| 1445 | 005212 | 177777 | MSG1 | | -1 |
| 1446 | 005214 | 000070 | MSG1 | | |
| 1447 | 005216 | 000105 | MSG2 | | |
| 1448 | 005220 | 000122 | MSG3 | | |
| 1449 | 005222 | 000137 | MSG4 | | |
| 1450 | 005224 | 000157 | MSG5 | | |
| 1451 | 005226 | 000174 | MSG6 | | |
| 1452 | 005230 | 000211 | MSG7 | | |
| 1453 | 005232 | 000226 | MSG10 | | |
| 1454 | 005234 | 000252 | MSG11 | | |
| 1455 | 005236 | 000262 | MSG12 | | |
| 1456 | 005240 | 000277 | MSG13 | | |
| 1457 | 005242 | 000314 | MSG.4 | | |
| 1458 | 005244 | 177777 | | | -1 |
| 1459 | 005246 | 177777 | | | -1 |
| 1460 | 005250 | 177777 | | | -1 |
| 1461 | 005252 | 177777 | | | -1 |
| 1462 | 005254 | 177777 | | | -1 |
| 1463 | 005256 | 177777 | | | -1 |
| 1464 | | | | | |
| 1465 | | | | | |

147942 = YB VERSION

149301 = YH VERSION
149301 = YE VERSION
149301 = YJ VERSION

| | | | | | |
|------|--------|--------|--------|--------|--|
| 1466 | 005260 | 005015 | 030122 | 020115 | TITLE .ASCIZ <15><12>/PAM TEST/ |
| 1467 | 005266 | 042524 | 052123 | 000 | |
| 1468 | 005273 | 015 | 052012 | 050131 | GETCRC .ASCIZ <15><12>/TYPE CRC VALUE / |
| 1469 | 005300 | 020105 | 051103 | 020103 | |
| 1470 | 005306 | 040526 | 052514 | 035105 | |
| 1471 | 005314 | 020040 | 000 | | |
| 1472 | 005317 | 015 | 052012 | 050131 | GETLPC .ASCIZ <15><12>/TYPE LPC VALUE / |
| 1473 | 005324 | 020105 | 050114 | 020103 | |
| 1474 | 005332 | 040526 | 052514 | 035105 | |
| 1475 | 005340 | 020040 | 000 | | |
| 1476 | 005343 | 015 | 042412 | 050130 | EXCRC .ASCIZ <15><12>/EXPECTED CRC # / |
| 1477 | 005350 | 041505 | 042524 | 020104 | |
| 1478 | 005356 | 051103 | 020103 | 020075 | |
| 1479 | 005364 | 000040 | | | |
| 1480 | 005366 | 005015 | 042412 | 050130 | EXLPC .ASCIZ <15><12><12>/EXPECTED LPC # / |
| 1481 | 005374 | 041505 | 042524 | 020104 | |
| 1482 | 005402 | 050114 | 020103 | 020075 | |
| 1483 | 005410 | 000040 | | | |
| 1484 | 005412 | 005015 | 047503 | 050115 | ACCRC .ASCIZ <15><12>/COMPUTED CRC # / |
| 1485 | 005420 | 052125 | 042105 | 041440 | |
| 1486 | 005426 | 041522 | 036440 | 020040 | |
| 1487 | 005434 | 000 | | | |
| 1488 | 005435 | 015 | 041412 | 046517 | ACLPC .ASCIZ <15><12>/COMPUTED LPC # / |
| 1489 | 005442 | 052524 | 042524 | 020104 | |
| 1490 | 005450 | 050114 | 020103 | 020075 | |
| 1491 | 005456 | 000040 | | | |
| 1492 | 005460 | 005015 | 042412 | 042116 | EOTST .ASCIZ <15><12><12>/END OF TEST/ |
| 1493 | 005466 | 047440 | 020106 | 042524 | |
| 1494 | 005474 | 052123 | 000 | | |

| | | | | | | | | | |
|------|--------|--------|--------|--------|---------|--------|---------------------------|---------------------------------|----------|
| 1495 | 005477 | 015 | 052012 | 050131 | BA11 | .ASCIZ | <15><12>/TYPE | STARTING ADDR. OF 1ST ROM ADDR. | SPACE1 / |
| 1496 | 005504 | 020105 | 052123 | 051101 | | | | | |
| 1497 | 005512 | 044524 | 043516 | 040440 | | | | | |
| 1498 | 005520 | 042104 | 027122 | 047440 | | | | | |
| 1499 | 005526 | 020106 | 051461 | 020124 | | | | | |
| 1500 | 005534 | 047522 | 020115 | 042101 | | | | | |
| 1501 | 005542 | 051104 | 020056 | 050123 | | | | | |
| 1502 | 005550 | 041501 | 035105 | 020040 | | | | | |
| 1503 | 005556 | 000 | | | | | | | |
| 1504 | 005557 | 015 | 052012 | 050131 | BA21 | .ASCIZ | <15><12>/TYPE | STARTING ADDR. OF 2ND ROM ADDR. | SPACE1 / |
| 1505 | 005564 | 020105 | 052123 | 051101 | | | | | |
| 1506 | 005572 | 044524 | 043516 | 040440 | | | | | |
| 1507 | 005600 | 042104 | 027122 | 047440 | | | | | |
| 1508 | 005606 | 020106 | 047062 | 020104 | | | | | |
| 1509 | 005614 | 047522 | 020115 | 042101 | | | | | |
| 1510 | 005622 | 051104 | 020056 | 050123 | | | | | |
| 1511 | 005630 | 041501 | 035105 | 020040 | | | | | |
| 1512 | 005636 | 000 | | | | | | | |
| 1513 | 005637 | 015 | 052012 | 050131 | SIZE11 | .ASCIZ | <15><12>/TYPE | LENGTH (BYTES) OF 1ST ROM ADDR. | SPACE1 / |
| 1514 | 005644 | 020105 | 042514 | 043516 | | | | | |
| 1515 | 005652 | 044124 | 024040 | 054502 | | | | | |
| 1516 | 005660 | 042524 | 024823 | 047440 | | | | | |
| 1517 | 005666 | 020106 | 051461 | 020124 | | | | | |
| 1518 | 005674 | 047522 | 020115 | 042101 | | | | | |
| 1519 | 005702 | 051104 | 020056 | 050123 | | | | | |
| 1520 | 005710 | 041501 | 035105 | 020040 | | | | | |
| 1521 | 005716 | 000 | | | | | | | |
| 1522 | 005717 | 015 | 052012 | 050131 | SIZE21 | .ASCIZ | <15><12>/TYPE | LENGTH (BYTES) OF 2ND ROM ADDR. | SPACE1 / |
| 1523 | 005724 | 020105 | 042514 | 043516 | | | | | |
| 1524 | 005732 | 044124 | 024040 | 054502 | | | | | |
| 1525 | 005740 | 042524 | 024823 | 047440 | | | | | |
| 1526 | 005746 | 020106 | 047062 | 020104 | | | | | |
| 1527 | 005754 | 047522 | 020115 | 042101 | | | | | |
| 1528 | 005762 | 051104 | 020056 | 050123 | | | | | |
| 1529 | 005770 | 041501 | 035105 | 020040 | | | | | |
| 1530 | 005776 | 000 | | | | | | | |
| 1531 | 005777 | 015 | 000012 | | CARLF1 | .ASCIZ | <15><12> | | |
| 1532 | 006002 | 020040 | 000 | | SP21 | .ASCIZ | / / | | |
| 1533 | 006005 | 015 | 040412 | 042104 | TYPHDP1 | .ASCIZ | <15><12>/ADDRESS | | DATA / |
| 1534 | 006012 | 042522 | 051523 | 020040 | | | | | |
| 1535 | 006020 | 020040 | 020040 | 020040 | | | | | |
| 1536 | 006026 | 020040 | 020040 | 020040 | | | | | |
| 1537 | 006034 | 042040 | 052101 | 000101 | | | | | |
| 1538 | 006042 | 020072 | 000040 | | COLON1 | .ASCIZ | / / | | |
| 1539 | 006046 | 005015 | 047125 | 047113 | AUTERM1 | .ASCIZ | <15><12>/UNKNOWN MODULE / | | |
| 1540 | 006054 | 053517 | 020116 | 047515 | | | | | |
| 1541 | 006062 | 052504 | 042514 | 000040 | | | | | |
| 1542 | | | | | | | | | |
| 1543 | 006070 | 005015 | 034515 | 030063 | MSG11 | .ASCIZ | <15><12>/M9301 = YA / | | |
| 1544 | 006076 | 020061 | 020055 | 040531 | | | | | |
| 1545 | 006104 | 000 | | | | | | | |
| 1546 | 006105 | 015 | 046412 | 031471 | MSG21 | .ASCIZ | <15><12>/M9301 = YB / | | |
| 1547 | 006112 | 030460 | 026440 | 054440 | | | | | |
| 1548 | 006120 | 000102 | | | | | | | |
| 1549 | 006122 | 005015 | 034515 | 030063 | MSG31 | .ASCIZ | <15><12>/M9301 = YC / | | |
| 1550 | 006130 | 020061 | 020055 | 041531 | | | | | |

| | | | | | | | | | | |
|------|--------|--------|--------|--------|--------|--------|-----------------|---|------------|--|
| 1551 | 006136 | 000 | | | | | | | | |
| 1552 | 006137 | 015 | 046412 | 032071 | MSG4: | .ASCIZ | <15><12>/M9400 | = | YA, YC/ | |
| 1553 | 006144 | 030060 | 026440 | 054440 | | | | | | |
| 1554 | 006152 | 026101 | 041531 | 000 | | | | | | |
| 1555 | 006157 | 015 | 046412 | 031471 | MSG5: | .ASCIZ | <15><12>/M9301 | = | YF/ | |
| 1556 | 006164 | 030460 | 026440 | 054440 | | | | | | |
| 1557 | 006172 | 000100 | | | | | | | | |
| 1558 | 006174 | 005015 | 033515 | 032071 | MSG6: | .ASCIZ | <15><12>/M7942 | = | YD/ | |
| 1559 | 006202 | 020062 | 020055 | 041131 | | | | | | |
| 1560 | 006210 | 000 | | | | | | | | |
| 1561 | 006211 | 015 | 046412 | 031471 | MSG7: | .ASCIZ | <15><12>/M9301 | = | YD/ | |
| 1562 | 006216 | 030460 | 026440 | 054440 | | | | | | |
| 1563 | 006224 | 000104 | | | | | | | | |
| 1564 | 006226 | 005015 | 034515 | 030060 | MSG10: | .ASCIZ | <15><12>/M9400 | = | YH(OR YP)/ | |
| 1565 | 006234 | 020060 | 020055 | 044131 | | | | | | |
| 1566 | 006242 | 047450 | 020122 | 045531 | | | | | | |
| 1567 | 006250 | 000051 | | | | | | | | |
| 1568 | 006252 | 005015 | 034515 | 030460 | MSG11: | .ASCIZ | <15><12>/M9311/ | | | |
| 1569 | 006260 | 000061 | | | | | | | | |
| 1570 | 006262 | 005015 | 034515 | 030060 | MSG12: | .ASCIZ | <15><12>/M9301 | = | YH/ | |
| 1571 | 006270 | 020061 | 020055 | 044131 | | | | | | |
| 1572 | 006276 | 000 | | | | | | | | |
| 1573 | 006277 | 015 | 046412 | 031471 | MSG13: | .ASCIZ | <15><12>/M9301 | = | YL/ | |
| 1574 | 006304 | 030460 | 026440 | 054440 | | | | | | |
| 1575 | 006312 | 000105 | | | | | | | | |
| 1576 | 006314 | 005015 | 034515 | 030060 | MSG14: | .ASCIZ | <15><12>/M9301 | = | YJ/ | |
| 1577 | 006322 | 020061 | 020055 | 045131 | | | | | | |
| 1578 | 006330 | 000 | | | | | | | | |
| 1579 | 000001 | | | | | .END | | | | |

| | | | | | | | | |
|---------|--------|------|-------|-------|-----|-----|------|-----|
| ABASE | 00000F | 559 | | | | | | |
| ACCPMG | 005412 | 697 | 14840 | | | | | |
| ACDN1 | 000000 | 559 | | | | | | |
| ACDN2 | 000000 | 559 | | | | | | |
| ACLPMG | 005435 | 715 | 14800 | | | | | |
| ACPUOF | 000000 | 559 | 574 | | | | | |
| ACTCRC | 001022 | 5190 | 6660 | 689 | 698 | 742 | 7700 | 843 |
| ACTLPC | 001024 | 5200 | 6670 | 700 | 716 | 795 | 8220 | |
| ADDW0 | 000000 | 559 | | | | | | |
| ADDW1 | 000000 | 559 | | | | | | |
| ADDW10 | 000000 | 559 | | | | | | |
| ADDW11 | 000000 | 559 | | | | | | |
| ADDW12 | 000000 | 559 | | | | | | |
| ADDW13 | 000000 | 559 | | | | | | |
| ADDW14 | 000000 | 559 | | | | | | |
| ADDW15 | 000000 | 559 | | | | | | |
| ADDW2 | 000000 | 559 | | | | | | |
| ADDW3 | 000000 | 559 | | | | | | |
| ADDW4 | 000000 | 559 | | | | | | |
| ADDW5 | 000000 | 559 | | | | | | |
| ADDW6 | 000000 | 559 | | | | | | |
| ADDW7 | 000000 | 559 | | | | | | |
| ADDW8 | 000000 | 559 | | | | | | |
| ADDW9 | 000000 | 559 | | | | | | |
| ADEVCT | 000000 | 559 | 565 | | | | | |
| ADEVH | 00000F | 559 | | | | | | |
| AENV | 000000 | 559 | 574 | | | | | |
| AENVH | 000000 | 559 | 571 | | | | | |
| AFATAL | 000000 | 559 | 502 | | | | | |
| AMADF1 | 000000 | 559 | | | | | | |
| AMADR2 | 000000 | 559 | | | | | | |
| AMADP3 | 000000 | 559 | | | | | | |
| AMADF4 | 000000 | 559 | | | | | | |
| AMAMS1 | 000000 | 559 | | | | | | |
| AMAMS2 | 000000 | 559 | | | | | | |
| AMAMS3 | 000000 | 559 | | | | | | |
| AMAMS4 | 000000 | 559 | | | | | | |
| AMSGAD | 000000 | 559 | 507 | | | | | |
| AMSGLG | 000000 | 559 | 508 | | | | | |
| AMSGTY | 000000 | 559 | 501 | | | | | |
| ANTYP1 | 000000 | 559 | | | | | | |
| ANTYP2 | 000000 | 559 | | | | | | |
| ANTYP3 | 000000 | 559 | | | | | | |
| ANTYP4 | 000000 | 559 | | | | | | |
| APASS | 000000 | 559 | 564 | | | | | |
| APRIOP | 000000 | 559 | | | | | | |
| APTCBU | 000040 | 1050 | 11610 | | | | | |
| APTEHV | 000001 | 1043 | 1117 | 11590 | | | | |
| APTSIZ | 000200 | 611 | 11500 | | | | | |
| APTSPO | 000100 | 1045 | 1119 | 11600 | | | | |
| ASHPREG | 000000 | 559 | 572 | | | | | |
| ATLBTN | 000000 | 559 | 563 | | | | | |
| AUNIT | 000000 | 559 | 566 | | | | | |
| AUSHH | 000000 | 559 | 573 | | | | | |
| AUTACT | 002522 | 600 | 8390 | | | | | |
| AUTERM | 006046 | 855 | 15390 | | | | | |

| | | | | | | | | |
|--------|--------|-----|------|-------|------|-----|-----|--|
| AUTERR | 022612 | 842 | 854 | | | | | |
| AUT1 | 02524 | 840 | 844 | | | | | |
| AUT2 | 02570 | 847 | 849 | | | | | |
| AUT3 | 02572 | 846 | 850 | | | | | |
| AVECT1 | 000000 | 559 | | | | | | |
| AVECT2 | 000000 | 559 | | | | | | |
| BIT0 | 000001 | 475 | | | | | | |
| BIT00 | 000001 | 465 | 475 | | | | | |
| BIT01 | 000002 | 464 | 474 | | | | | |
| BIT02 | 000004 | 463 | 473 | | | | | |
| BIT03 | 000010 | 462 | 472 | | | | | |
| BIT04 | 000020 | 461 | 471 | | | | | |
| BIT05 | 000040 | 460 | 470 | | | | | |
| BIT06 | 000100 | 459 | 469 | | | | | |
| BIT07 | 000200 | 458 | 468 | | | | | |
| BIT08 | 000400 | 457 | 467 | | | | | |
| BIT09 | 001000 | 456 | 466 | | | | | |
| BIT1 | 000002 | 474 | | | | | | |
| BIT10 | 002000 | 455 | | | | | | |
| BIT11 | 004000 | 454 | | | | | | |
| BIT12 | 010000 | 453 | | | | | | |
| BIT13 | 020000 | 452 | | | | | | |
| BIT14 | 040000 | 451 | | | | | | |
| BIT15 | 100000 | 450 | | | | | | |
| BIT2 | 000004 | 473 | | | | | | |
| BIT3 | 000010 | 472 | | | | | | |
| BIT4 | 000020 | 471 | | | | | | |
| BIT5 | 000040 | 470 | | | | | | |
| BIT6 | 000100 | 469 | | | | | | |
| BIT7 | 000200 | 468 | | | | | | |
| BIT8 | 000400 | 467 | | | | | | |
| BIT9 | 001000 | 466 | | | | | | |
| BPIVEC | 000010 | 482 | | | | | | |
| CARLF | 005777 | 822 | 827 | 15310 | | | | |
| CHECK | 001510 | 632 | 636 | 636 | 6630 | | | |
| CH0 | 001500 | 669 | 670 | | | | | |
| CH1 | 001644 | 670 | 687 | 689 | | | | |
| CK1 | 001736 | 690 | 703 | 700 | 852 | | | |
| CK2 | 002030 | 709 | 721 | 726 | | | | |
| CLLAST | 002204 | 763 | 765 | | | | | |
| CLP0 | 002252 | 774 | 790 | | | | | |
| CLP1 | 002264 | 775 | 777 | | | | | |
| CLP2 | 002316 | 784 | 786 | | | | | |
| CL0 | 002102 | 741 | 747 | 764 | | | | |
| CL1 | 002146 | 755 | | | | | | |
| CL2 | 002170 | 752 | 761 | | | | | |
| CL3 | 002122 | 743 | 740 | | | | | |
| COLON | 000042 | 831 | 1530 | | | | | |
| CP | 000015 | 390 | 1099 | 1099 | | | | |
| CRC | 002074 | 671 | 670 | 740 | | | | |
| CRLF | 000200 | 391 | 1000 | 1099 | | | | |
| DATLN1 | 001006 | 513 | 620 | 654 | 660 | 673 | 800 | |
| DATLN2 | 001012 | 515 | 630 | 662 | 677 | 681 | 813 | |
| DDISP | 177570 | 397 | 599 | | | | | |
| DISPLA | 001002 | 511 | 599 | 607 | | | | |
| DISPRT | 000174 | 490 | 647 | | | | | |

| | | | | | | | | | | |
|--------|------------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| DSWP | = 177570 | 3060 | 590 | | | | | | | |
| ENTVEC | = 000030 | 4050 | | | | | | | | |
| ENDOT | = 002046 | 015 | 0170 | | | | | | | |
| EOTST | = 005060 | 729 | 010 | 14020 | | | | | | |
| ERRVEC | = 000004 | 4700 | 596 | 5970 | 6000 | | | | | |
| EXCPC | = 001016 | 5170 | 6270 | 6420 | 689 | 694 | 0500 | | | |
| EXCRMG | = 005343 | 693 | 14760 | | | | | | | |
| EXLPC | = 001020 | 5100 | 6200 | 6460 | 700 | 712 | 0510 | | | |
| EXLPNG | = 005366 | 711 | 14000 | | | | | | | |
| GETCPC | = 005273 | 640 | 14600 | | | | | | | |
| GETIN | = 001370 | 624 | 6130 | | | | | | | |
| GETLPC | = 005317 | 644 | 14720 | | | | | | | |
| GET2 | = 001440 | 630 | 6470 | | | | | | | |
| GNS | = 000000 U | 497 | 1313 | 1314 | 1315 | 1316 | 1319 | 1320 | 1321 | |
| GOAGIN | = 002070 | 732 | 7300 | | | | | | | |
| HT | = 000011 | 3000 | 1050 | 1009 | | | | | | |
| IOTVEC | = 000020 | 4030 | | | | | | | | |
| LF | = 000012 | 3090 | 1093 | 1099 | | | | | | |
| LPC | = 002344 | 675 | 603 | 7950 | | | | | | |
| LPC1 | = 002352 | 7000 | 001 | | | | | | | |
| LPC2 | = 002366 | 790 | 0000 | | | | | | | |
| MSG1 | = 000070 | 1446 | 15430 | | | | | | | |
| MSG10 | = 000226 | 1453 | 15640 | | | | | | | |
| MSG11 | = 000252 | 1454 | 15600 | | | | | | | |
| MSG12 | = 000262 | 1455 | 15700 | | | | | | | |
| MSG13 | = 000277 | 1456 | 15730 | | | | | | | |
| MSG14 | = 000314 | 1457 | 15760 | | | | | | | |
| MSG2 | = 000105 | 1447 | 15460 | | | | | | | |
| MSG3 | = 000122 | 1448 | 15490 | | | | | | | |
| MSG4 | = 000137 | 1449 | 15520 | | | | | | | |
| MSG5 | = 000157 | 1450 | 15550 | | | | | | | |
| MSG6 | = 000174 | 1451 | 15500 | | | | | | | |
| MSG7 | = 000211 | 1452 | 15010 | | | | | | | |
| PARCNT | = 001026 | 9210 | 7720 | 7760 | 743 | | | | | |
| PARITY | = 002262 | 740 | 7720 | | | | | | | |
| PIRO | = 177772 | 3980 | | | | | | | | |
| PIROVE | = 000260 | 4090 | | | | | | | | |
| PR0 | = 000000 | 4120 | | | | | | | | |
| PR1 | = 000040 | 4130 | | | | | | | | |
| PR2 | = 000100 | 4140 | | | | | | | | |
| PR3 | = 000140 | 4150 | | | | | | | | |
| PR4 | = 000200 | 4160 | | | | | | | | |
| PR5 | = 000260 | 4170 | | | | | | | | |
| PR6 | = 000300 | 4180 | | | | | | | | |
| PR7 | = 000360 | 4190 | | | | | | | | |
| PS | = 177776 | 3920 | 393 | 5060 | | | | | | |
| PS0 | = 177776 | 3930 | | | | | | | | |
| PRVVEC | = 000024 | 4040 | 5920 | 5930 | 12430 | 12440 | 12530 | 12590 | 12710 | 12720 |
| RDCHA | = 104405 | 911 | 13190 | | | | | | | |
| RDLIN | = 104406 | 906 | 13200 | | | | | | | |
| RDOC1 | = 104407 | 641 | 645 | 640 | 653 | 657 | 661 | 13210 | | |
| REBTHT | = 001274 | 616 | 6200 | 730 | | | | | | |
| REBVEC | = 000010 | 4700 | | | | | | | | |
| ROMBA1 | = 001004 | 5120 | 6290 | 6500 | 670 | 672 | 007 | | | |
| ROMBA2 | = 001010 | 5140 | 6310 | 6500 | 676 | 680 | 012 | | | |
| BA1 | = 005477 | 640 | 14950 | | | | | | | |

| | | | | | | | | | | | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|------|------|-------|------|-------|-----|-----|-----|-----|--|--|--|--|--|
| SA2 | 005557 | 656 | 15040 | | | | | | | | | | | | | | | | |
| SIZE1 | 005637 | 652 | 15130 | | | | | | | | | | | | | | | | |
| SIZE2 | 005717 | 660 | 15220 | | | | | | | | | | | | | | | | |
| SP2 | 006002 | 835 | 15320 | | | | | | | | | | | | | | | | |
| STACK | 001100 | 3030 | | | | | | | | | | | | | | | | | |
| START | 001076 | 502 | 504 | 5040 | | | | | | | | | | | | | | | |
| STKLMT | 177774 | 3940 | | | | | | | | | | | | | | | | | |
| ST2 | 001320 | 6250 | | | | | | | | | | | | | | | | | |
| SWR | 001000 | 5100 | 5900 | 600 | 6060 | 6130 | 6220 | 623 | 1251 | 17040 | | | | | | | | | |
| SWREG | 000176 | 4990 | 606 | 622 | | | | | | | | | | | | | | | |
| SW0 | 000001 | 4470 | | | | | | | | | | | | | | | | | |
| SW00 | 000001 | 4370 | 447 | | | | | | | | | | | | | | | | |
| SW01 | 000002 | 4360 | 446 | | | | | | | | | | | | | | | | |
| SW02 | 000004 | 4350 | 445 | | | | | | | | | | | | | | | | |
| SW03 | 000010 | 4340 | 444 | | | | | | | | | | | | | | | | |
| SW04 | 000020 | 4330 | 443 | | | | | | | | | | | | | | | | |
| SW05 | 000040 | 4320 | 442 | | | | | | | | | | | | | | | | |
| SW06 | 000100 | 4310 | 441 | | | | | | | | | | | | | | | | |
| SW07 | 000200 | 4300 | 440 | | | | | | | | | | | | | | | | |
| SW08 | 000400 | 4290 | 439 | | | | | | | | | | | | | | | | |
| SW09 | 001000 | 4280 | 438 | | | | | | | | | | | | | | | | |
| SW1 | 000002 | 4460 | | | | | | | | | | | | | | | | | |
| SW10 | 002000 | 4270 | | | | | | | | | | | | | | | | | |
| SW11 | 004000 | 4260 | | | | | | | | | | | | | | | | | |
| SW12 | 010000 | 4250 | | | | | | | | | | | | | | | | | |
| SW13 | 020000 | 4240 | | | | | | | | | | | | | | | | | |
| SW14 | 040000 | 4230 | | | | | | | | | | | | | | | | | |
| SW15 | 100000 | 4220 | | | | | | | | | | | | | | | | | |
| SW2 | 000001 | 4450 | | | | | | | | | | | | | | | | | |
| SW3 | 000010 | 4440 | | | | | | | | | | | | | | | | | |
| SW4 | 000020 | 4430 | | | | | | | | | | | | | | | | | |
| SW5 | 000040 | 4420 | | | | | | | | | | | | | | | | | |
| SW6 | 000100 | 4410 | | | | | | | | | | | | | | | | | |
| SW7 | 000200 | 4400 | | | | | | | | | | | | | | | | | |
| SW8 | 000400 | 4390 | | | | | | | | | | | | | | | | | |
| SW9 | 001000 | 4380 | | | | | | | | | | | | | | | | | |
| TBITVE | 000014 | 4000 | | | | | | | | | | | | | | | | | |
| TDLN1 | 004762 | 620 | 13040 | | | | | | | | | | | | | | | | |
| TDLN2 | 005076 | 630 | 14040 | | | | | | | | | | | | | | | | |
| TITL | 005260 | 610 | 14060 | | | | | | | | | | | | | | | | |
| TKVEC | 000060 | 4070 | | | | | | | | | | | | | | | | | |
| TMSG | 005212 | 841 | 847 | 14450 | | | | | | | | | | | | | | | |
| TPVEC | 000064 | 4000 | | | | | | | | | | | | | | | | | |
| TRAFVE | 000034 | 4060 | 5070 | 5000 | 5010 | | | | | | | | | | | | | | |
| TRMSA1 | 005036 | 620 | 13040 | | | | | | | | | | | | | | | | |
| TRMSA2 | 005144 | 631 | 14240 | | | | | | | | | | | | | | | | |
| TRTVEC | 000014 | 4010 | | | | | | | | | | | | | | | | | |
| TXCPC | 004644 | 627 | 843 | 850 | 13230 | | | | | | | | | | | | | | |
| TXLPC | 004712 | 626 | 851 | 13430 | | | | | | | | | | | | | | | |
| TYP | 002454 | 811 | 816 | 8210 | | | | | | | | | | | | | | | |
| TYPE | 104401 | 617 | 639 | 643 | 647 | 651 | 655 | 659 | 692 | 696 | 710 | 714 | 720 | 805 | | | | | |
| | | 817 | 821 | 826 | 830 | 834 | 840 | 854 | 910 | 924 | 929 | 933 | 936 | 939 | | | | | |
| | | 941 | 944 | 948 | 1015 | 1017 | 1003 | 1272 | 1273 | 13130 | | | | | | | | | |
| TYPHDR | 006005 | 806 | 15330 | | | | | | | | | | | | | | | | |
| TYPOC | 104402 | 695 | 699 | 713 | 717 | 829 | 833 | 13140 | | | | | | | | | | | |
| TYPON | 104404 | 13160 | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | |
|--------|--------|---|-------|-------|-------|-------|-------|-------|-------|------|-------|------|------|-----|------|--|--|--|--|--|
| TYPOS | 104403 | | 13150 | | | | | | | | | | | | | | | | | |
| TYPOUT | 001030 | | 5010 | 5030 | 5220 | 637 | 663 | | | | | | | | | | | | | |
| TYPRM | 002402 | | 665 | 8050 | | | | | | | | | | | | | | | | |
| TYPR1 | 002420 | | 010 | 0120 | | | | | | | | | | | | | | | | |
| TYPE | 002462 | | 0240 | 037 | | | | | | | | | | | | | | | | |
| TYP2 | 002504 | | 025 | 0320 | | | | | | | | | | | | | | | | |
| TYP3 | 002470 | | 023 | 0260 | | | | | | | | | | | | | | | | |
| XOR | 002320 | | 749 | 7000 | 799 | | | | | | | | | | | | | | | |
| XORS | 001014 | | 5160 | 7400 | 7610 | 765 | 7660 | 7670 | 7690 | 770 | 789 | 7900 | 7910 | 792 | 7950 | | | | | |
| | | | 802 | | | | | | | | | | | | | | | | | |
| SAPTHD | 001032 | | 543 | 5490 | | | | | | | | | | | | | | | | |
| SASTAT | 000000 | U | 1139 | 1150 | | | | | | | | | | | | | | | | |
| SATYC | 003742 | | 1110 | 11120 | | | | | | | | | | | | | | | | |
| SATY1 | 003710 | | 11000 | | | | | | | | | | | | | | | | | |
| SATY3 | 003724 | | 1040 | 11090 | | | | | | | | | | | | | | | | |
| SATY4 | 003734 | | 11110 | | | | | | | | | | | | | | | | | |
| SCHARC | 003702 | | 10050 | 10750 | 1092 | 10910 | 10960 | | | | | | | | | | | | | |
| SCKSRM | 000000 | U | 1319 | | | | | | | | | | | | | | | | | |
| SCMTAG | 000000 | U | 500 | | | | | | | | | | | | | | | | | |
| SCNTLC | 003235 | | 9620 | | | | | | | | | | | | | | | | | |
| SCNTLU | 003230 | | 933 | 9610 | | | | | | | | | | | | | | | | |
| SCPUOP | 001074 | | 5740 | | | | | | | | | | | | | | | | | |
| SCRLE | 003225 | | 930 | 9590 | 1020 | 1064 | 1105 | | | | | | | | | | | | | |
| SDEVCT | 001056 | | 5650 | | | | | | | | | | | | | | | | | |
| SENDAD | 002000 | | 930 | 615 | 700 | 710 | 720 | 7300 | 845 | | | | | | | | | | | |
| SENV | 001066 | | 5700 | 635 | 684 | 702 | 720 | 1043 | 1117 | 1141 | | | | | | | | | | |
| SENVN | 001007 | | 5710 | 611 | 1045 | 1050 | 1119 | | | | | | | | | | | | | |
| SETABL | 001060 | | 5690 | | | | | | | | | | | | | | | | | |
| SETEND | 001076 | | 555 | 5010 | | | | | | | | | | | | | | | | |
| SFATAL | 001050 | | 5620 | 5040 | 7040 | 7220 | 8560 | 11450 | | | | | | | | | | | | |
| SFFLG | 000102 | | 11000 | 11110 | 1139 | 11100 | 11560 | | | | | | | | | | | | | |
| SFILLC | 003714 | | 1060 | 11030 | | | | | | | | | | | | | | | | |
| SFILLS | 003713 | | 11020 | | | | | | | | | | | | | | | | | |
| SGTSMF | 000000 | U | 1310 | | | | | | | | | | | | | | | | | |
| SHIBTS | 001032 | | 5500 | | | | | | | | | | | | | | | | | |
| SHIOCT | 003022 | | 10080 | 10190 | | | | | | | | | | | | | | | | |
| SILLUP | 004552 | | 1243 | 1259 | 12760 | | | | | | | | | | | | | | | |
| SLF | 003220 | | 940 | 9000 | 1020 | 1105 | | | | | | | | | | | | | | |
| SLFLG | 000101 | | 11490 | 11550 | | | | | | | | | | | | | | | | |
| SMAIL | 001046 | | 551 | 555 | 5000 | 610 | 1043 | | | | | | | | | | | | | |
| SMBADR | 001034 | | 5510 | | | | | | | | | | | | | | | | | |
| SNFLG | 000100 | | 11090 | 1115 | 11500 | 11540 | | | | | | | | | | | | | | |
| SNNEL | 003253 | | 9650 | | | | | | | | | | | | | | | | | |
| SNSGAD | 001002 | | 5670 | 11250 | 1120 | | | | | | | | | | | | | | | |
| SNSGLC | 001064 | | 5680 | 11300 | | | | | | | | | | | | | | | | |
| SNSGTY | 001046 | | 5610 | 5050 | 7050 | 7230 | 8570 | 1123 | 11310 | 1143 | 11670 | | | | | | | | | |
| SNSWP | 003242 | | 9630 | | | | | | | | | | | | | | | | | |
| SNULL | 003712 | | 1070 | 11010 | | | | | | | | | | | | | | | | |
| SOCHT | 004406 | | 11940 | 12230 | 12360 | | | | | | | | | | | | | | | |
| SOMODE | 004410 | | 11090 | 11330 | 1190 | 12010 | 12120 | 12300 | | | | | | | | | | | | |
| SPASS | 001054 | | 5640 | 6100 | 7300 | | | | | | | | | | | | | | | |
| SPASTM | 001040 | | 5530 | | | | | | | | | | | | | | | | | |
| SPOWLF | 004560 | | 1274 | 12790 | | | | | | | | | | | | | | | | |
| SPWRDN | 004412 | | 592 | 12430 | 1271 | | | | | | | | | | | | | | | |
| SPWRMC | 004540 | | 12740 | | | | | | | | | | | | | | | | | |
| SPWRUP | 004464 | | 1253 | 12590 | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | |
|---------|----------|-------|-------|-------|-------|-------|------|-------|-------|-------|------|------|------|------|--|--|--|--|--|--|
| BOUES | 003224 | 941 | 9500 | 1017 | 1020 | 1105 | | | | | | | | | | | | | | |
| BRDCH | 002640 | 8700 | 1319 | | | | | | | | | | | | | | | | | |
| BRDECB | 000000 U | 1322 | | | | | | | | | | | | | | | | | | |
| BRDLIN | 002760 | 9060 | 1320 | | | | | | | | | | | | | | | | | |
| BRDOCT | 003264 | 9010 | 1321 | | | | | | | | | | | | | | | | | |
| BRDSZ | 000010 | 8990 | | | | | | | | | | | | | | | | | | |
| BR2A | 000000 U | 1322 | | | | | | | | | | | | | | | | | | |
| BSAVPE | 000000 U | 1322 | | | | | | | | | | | | | | | | | | |
| BSAVH | 004556 | 12520 | 1260 | 12610 | 12620 | 12700 | | | | | | | | | | | | | | |
| BSSETUP | 000014 | 5000 | 509 | 590 | 592 | 594 | 867 | 967 | | | | | | | | | | | | |
| BSIIP | 177777 | 5000 | | | | | | | | | | | | | | | | | | |
| BSVPC | 001032 | 5200 | 533 | | | | | | | | | | | | | | | | | |
| BSWP | 000000 | 3790 | 1275 | | | | | | | | | | | | | | | | | |
| BSWREG | 001070 | 5720 | 613 | | | | | | | | | | | | | | | | | |
| BTESTN | 001052 | 5630 | 5060 | | | | | | | | | | | | | | | | | |
| BTKB | 002636 | 8640 | 802 | 800 | | | | | | | | | | | | | | | | |
| BTKS | 002634 | 8630 | 800 | 806 | | | | | | | | | | | | | | | | |
| BTPB | 003710 | 10000 | 11000 | | | | | | | | | | | | | | | | | |
| BTPFLG | 003715 | 1037 | 11040 | | | | | | | | | | | | | | | | | |
| BTPS | 003706 | 1000 | 10990 | | | | | | | | | | | | | | | | | |
| BTRAP | 004570 | 590 | 12900 | | | | | | | | | | | | | | | | | |
| BTRAP2 | 004612 | 13010 | 1312 | | | | | | | | | | | | | | | | | |
| BTRP | 000010 | 13050 | 13140 | 13150 | 13160 | 13170 | 1319 | 13200 | 13210 | 13220 | | | | | | | | | | |
| BTRPAD | 004624 | 1295 | 13120 | | | | | | | | | | | | | | | | | |
| BSTN | 001036 | 5520 | | | | | | | | | | | | | | | | | | |
| BSTYIN | 003214 | 900 | 909 | 921 | 939 | 953 | 9570 | | | | | | | | | | | | | |
| BSTYUN | 000000 U | 1317 | | | | | | | | | | | | | | | | | | |
| BSTYUN | 000000 U | 1317 | | | | | | | | | | | | | | | | | | |
| BTYPE | 003424 | 10370 | 1136 | 1305 | 1313 | | | | | | | | | | | | | | | |
| BTYPEC | 003636 | 1067 | 1074 | 1001 | 10000 | 1007 | | | | | | | | | | | | | | |
| BTYPEX | 003704 | 1002 | 1094 | 10970 | | | | | | | | | | | | | | | | |
| BTYPOC | 004210 | 11920 | 1310 | | | | | | | | | | | | | | | | | |
| BTYPON | 004224 | 1191 | 11940 | 1316 | | | | | | | | | | | | | | | | |
| BTYPOS | 004164 | 11070 | 1315 | | | | | | | | | | | | | | | | | |
| BUNIT | 001060 | 5660 | | | | | | | | | | | | | | | | | | |
| BUNITM | 001042 | 5540 | | | | | | | | | | | | | | | | | | |
| BUSWP | 001072 | 5730 | | | | | | | | | | | | | | | | | | |
| BSFILL | 004407 | 11000 | 11920 | 1202 | 12370 | | | | | | | | | | | | | | | |
| . | 006331 | 4900 | 4930 | 4970 | 5000 | 5090 | 520 | 5290 | 5310 | 5330 | 539 | 5400 | 5420 | 5440 | | | | | | |
| | | 863 | 864 | 9570 | 958 | 959 | 960 | 967 | 1020 | 1099 | 1100 | 1101 | 1102 | 1103 | | | | | | |
| | | 1104 | 1105 | 11570 | 1255 | 1277 | | | | | | | | | | | | | | |
| BSASTA | 000000 U | 1109 | 1112 | | | | | | | | | | | | | | | | | |
| BSX | 001032 | 5390 | 544 | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | |
|--------|-------|------|------|------|------|------|------|------|-----|-----|-----|------|------|------|------|--|--|--|--|
| COMPEN | 4900 | | | | | | | | | | | | | | | | | | |
| ENDCOM | 4900 | | | | | | | | | | | | | | | | | | |
| ERROR | 3840 | | | | | | | | | | | | | | | | | | |
| ESCAPE | 4900 | | | | | | | | | | | | | | | | | | |
| GETPRI | 4900 | | | | | | | | | | | | | | | | | | |
| GETSWR | 4900 | | | | | | | | | | | | | | | | | | |
| MULT | 4900 | | | | | | | | | | | | | | | | | | |
| NEWST | 4900 | | | | | | | | | | | | | | | | | | |
| POP | 3790 | 4900 | 1009 | 1151 | 1152 | 1264 | 1265 | | | | | | | | | | | | |
| PUSH | 3790 | 4900 | 983 | 1112 | 1114 | 1135 | 1245 | 1251 | | | | | | | | | | | |
| REPORT | 4900 | | | | | | | | | | | | | | | | | | |
| SCOPE | 3050 | | | | | | | | | | | | | | | | | | |
| SETPRI | 4900 | | | | | | | | | | | | | | | | | | |
| SETTFA | 13050 | 1314 | 1315 | 1316 | 1319 | 1320 | 1321 | | | | | | | | | | | | |
| SETUP | 3790 | 4900 | 507 | | | | | | | | | | | | | | | | |
| SKIP | 4900 | | | | | | | | | | | | | | | | | | |
| SLASH | 4900 | | | | | | | | | | | | | | | | | | |
| SPACE | 4900 | | | | | | | | | | | | | | | | | | |
| STARS | 3790 | 4900 | 526 | 536 | 538 | 545 | 550 | 862 | 870 | 899 | 909 | 1022 | 1107 | 1164 | 1241 | | | | |
| | 1257 | 1284 | | | | | | | | | | | | | | | | | |
| SWRSU | 4900 | 5940 | | | | | | | | | | | | | | | | | |
| TPMTP | 13050 | | | | | | | | | | | | | | | | | | |
| TYPRIN | 4900 | | | | | | | | | | | | | | | | | | |
| TYPDEC | 4900 | | | | | | | | | | | | | | | | | | |
| TYPNAM | 4900 | | | | | | | | | | | | | | | | | | |
| TYPNUM | 4900 | | | | | | | | | | | | | | | | | | |
| TYPOCS | 4900 | | | | | | | | | | | | | | | | | | |
| TYPOCT | 4900 | | | | | | | | | | | | | | | | | | |
| TYPTAT | 4900 | | | | | | | | | | | | | | | | | | |
| SSBSCA | 4900 | | | | | | | | | | | | | | | | | | |
| SSNEWT | 4900 | | | | | | | | | | | | | | | | | | |
| SSSET | 13050 | 1314 | 1315 | 1316 | 1319 | 1320 | 1321 | | | | | | | | | | | | |
| SSSETH | 6100 | | | | | | | | | | | | | | | | | | |
| SSSKIP | 4900 | | | | | | | | | | | | | | | | | | |
| .EQUAT | 3790 | 300 | | | | | | | | | | | | | | | | | |
| .SETUP | 3790 | 500 | | | | | | | | | | | | | | | | | |
| .SACT1 | 3790 | 524 | | | | | | | | | | | | | | | | | |
| .SAPTR | 3790 | 556 | | | | | | | | | | | | | | | | | |
| .SAPTH | 3790 | 534 | | | | | | | | | | | | | | | | | |
| .SAPTY | 3790 | 1105 | | | | | | | | | | | | | | | | | |
| .SCATC | 3790 | 491 | | | | | | | | | | | | | | | | | |
| .SPOWE | 3790 | 1239 | | | | | | | | | | | | | | | | | |
| .SRDOC | 3790 | 967 | | | | | | | | | | | | | | | | | |
| .SREAU | 3790 | 860 | | | | | | | | | | | | | | | | | |
| .STRAP | 3790 | 1282 | | | | | | | | | | | | | | | | | |
| .STYPE | 3790 | 1020 | | | | | | | | | | | | | | | | | |
| .SIYPO | 3790 | 1102 | | | | | | | | | | | | | | | | | |

. ABS. 006331 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DZM9AD.0IN,DZM9AD.LST/CRF/SOL/NL:UC=DZM9AD.P11

.MAIN. MACY11 27(1026) 24-MAY-77 15124 PAGE 4#
DZM9AU.P11 24-MAY-77 15122 CROSS REFERENCE TABLE -- MACRO NAMES

SLU 4030

RUN-TIME: 11 6 .5 SECONDS
RUN-TIME RATIO: 632/16=39.6
CORE USED: 20K (40 PAGES)