

VT62/DZ11

SELF TEST
CZDZCB0

AH-A920B-MC

COPYRIGHT © 77-78

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

The main body of the document consists of a grid of 150 small, illegible data tables or test results, arranged in 10 columns and 15 rows. Each cell in the grid contains a small table with multiple columns and rows of text, which is too small to read. The grid is located on the left side of the page, with a large blank area on the right.

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

000000 .REPT 0

IDENTIFICATION

PRODUCT CODE: AC-A919B-MC
PRODUCT NAME: CZDZCB0 VT62/DZ11 TST
DATE: 1-DEC-78
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: MICHAEL DENSMORE

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

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, 1978 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

38
39
40
41
42
43
44

REVISION HISTORY

REVISION
A

BY:
M. DENSMORE ORIGINAL RELEASE

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
8.

TABLE OF CONTENTS

1. GENERAL	PG.	4
1.1 ABSTRACT	PG.	4
1.2 DEFINITIONS	PG.	4
1.3 DESIGN NOTES	PG.	4
1.4 DDCMP	PG.	5
2. EQUIPMENT REQUIREMENTS	PG.	5
3. LOAD PROCEDURE	PG.	5
4. STARTING AND RUNNING THE PROGRAM	PG.	5
4.1 START	PG.	6
4.2 RESTART	PG.	9
4.3 CONTINUE	PG.	10
4.4 ZERO FLAGS	PG.	10
4.5 ADD UNITS	PG.	10
4.6 DROP UNITS	PG.	10
4.7 DISPLAY PTABLE	PG.	11
4.8 SWITCH OPTIONS	PG.	11
5. ERROR AND INFORMATIONAL MESSAGES	PG.	12
6. RESTRICTIONS	PG.	13
7. MISC.	PG.	13
8. TEST DESCRIPTIONS	PG.	13
8.1 MICRODIAGNOSTIC	PG.	13
8.2 LOOPBACK	PG.	13
9. INDEX	PG.	15

83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

1.0
1.1

GENERAL
ABSTRACT

CZDZC RUNS THE SELFTEST FOR THE VT62 TERMINAL AND EXERCISES THE LINK BETWEEN THE TERMINAL AND THE HOST. THIS PARTICULAR VERSION SUPPORTS MULTIDROP, FULL DUPLEX COMMUNICATION VIA THE DZ-11 COMMUNICATION INTERFACE. OTHER VERSIONS (MAINDEC-11-DZDPG AND MAINDEC-11-DZDVP) SUPPORT THE DUP-11 AND DV-11 COMMUNICATION INTERFACES RESPECTIVELY. THE PROGRAM IS XXDP COMPATIBLE AND SUPPORTS SEQUENTIAL TESTING OF UP TO 8 VT62'S. PLEASE READ ALL SECTIONS OF THIS DOCUMENT CAREFULLY BEFORE USING THE PROGRAM. PAY PARTICULAR ATTENTION TO SECTIONS 3.0 THROUGH 6.0

1.2

DEFINITIONS

MODEM - EQUIPMENT, ALSO KNOWN AS A 'DATASET', WHICH PERFORMS THE MODULATION/DEMODULATION OF DATA SIGNALS. THIS UNIT PROVIDES TIMING FOR SYNCHRONOUS COMMUNICATION DEVICES.

MODEM ELIMINATOR - EQUIPMENT WHICH CONNECTS TWO SYNCHRONOUS DEVICES USING A CABLE RATHER THAN TELEPHONE VOICE LINES. THE ELIMINATOR HAS A CLOCK FOR TIMING.

DDCMP PROTOCOL - DEC'S COMMUNICATION LINE PROTOCOL.

UUT - UNIT UNDER TEST.

PTABLE - A TABLE OF INFORMATION, WITH ENTRIES FOR EACH UNIT UNDER TEST, THAT DESCRIBES THE HARDWARE (BUS ADDRESS, VECTOR, ETC.)

DUP - SINGLE LINE, SYNCHRONOUS COMMUNICATION INTERFACE.

DZ - EIGHT LINE, MULTIPLEXED, SYNCHRONOUS COMMUNICATION INTERFACE.

DV - SIXTEEN LINE, MULTIPLEXED, SYNCHRONOUS COMMUNICATION INTERFACE.

CRC CHECKSUM - CHECKSUM USED IN DEC NETWORK PROTOCOL (CYCLIC REDUNDANCY CHECK).

MICRODIAGNOSTIC - THE MICROCODE IN THE VT62 THAT PERFORMS THE DIAGNOSTIC CHECKOUT OF THE HARDWARE. (ALSO KNOWN AS 'SELF TEST')

SELF TEST - SEE 'MICRODIAGNOSTIC'.

1.3

DESIGN NOTES

THIS PROGRAM WAS DESIGNED AND IMPLEMENTED USING STRUCTURED TECHNIQUES. THE PROCEDURE CAUSED THE PROGRAM TO HAVE SOME OPERATIONAL CHARACTERISTICS WHICH ARE OF INTEREST TO THE USER. THESE CHARACTERISTICS ARE DISCUSSED IN THIS SECTION. THIS DISCUSSION IS NOT VITAL TO LEARNING HOW TO OPERATE THE PROGRAM HOWEVER.

THE MOST IMPORTANT OPERATIONAL CHARACTERISTIC OF STRUCTURED DESIGN, AS IT AFFECTS THE USER, IS HOW ERRORS ARE REPORTED. THE TOP-DOWN DESIGN OF THE PROGRAM RESULTS IN A BOTTOM-UP REPORT OF ERROR INFORMATION. SEVERAL LINES OF ERROR INFORMATION ARE PRINTED AS EACH PROGRAM SECTION REPORTS ITS LEVEL OF INFORMATION. AN EXAMPLE OF THIS TYPE OF ERROR LOG WOULD BE A DZ ERROR. THE FIRST ERROR MESSAGE WOULD GIVE SPECIFIC DEVICE INFORMATION SUCH AS REGISTER CONTENTS AND THEN RETURN CONTROL TO THE NEXT LEVEL. AT THE NEXT LEVEL, THE LOGICAL DEVICE NUMBER OR DEVICE SERIAL NUMBER WOULD BE REPORTED. AT THE LEVEL ABOVE THAT, STILL MORE GENERAL INFORMATION WOULD BE GIVEN. THE REPORTS CONTINUE UNTIL THE HIGHEST LEVEL IS REACHED.

COMMENT FIELDS IN THE LISTING WITH SQUARE BRACKETS ([]) REFER TO

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190

FUNCTIONAL BLOCKS IN THE DESIGN AND HAVE NO SIGNIFICANCE TO THE USER. THEY ARE RETAINED FOR USE BY PROGRAM MAINTAINERS.

1.4

DDCMP

THIS SECTION GIVES A VERY BRIEF DESCRIPTION OF DDCMP AS USED BY THIS DIAGNOSTIC. THE DESCRIPTION IS BY NO MEANS COMPLETE; IT IS INTENDED TO HELP THE USER UNDERSTAND ERROR REPORTS.

THERE ARE THREE TYPES OF DDCMP MESSAGES: CONTROL, DATA AND BOOTSTRAP. THE BOOTSTRAP MESSAGE IS NOT USED BY THIS DIAGNOSTIC. THE DATA MESSAGES CONSIST OF A 6-BYTE HEADER, A 2-BYTE CRC FOR THE HEADER, THE DATA ITSELF AND A 2-BYTE CRC FOR THE DATA. CONTROL MESSAGES CONSIST OF THE HEADER (AND ITS CRC) ONLY. ALL CONTROL MESSAGES HAVE AN 'ENQ' CHARACTER AS THE FIRST BYTE. ALL DATA MESSAGES ARE NUMBERED SO THAT EACH STATION ON THE LINK CAN KEEP TRACK OF WHICH MESSAGES ARE TRANSMITTED PROPERLY. ALL MESSAGES HAVE A STATION ADDRESS IN THE HEADER SO THAT MESSAGES ARE DIRECTED TO AND FROM THE PROPER STATIONS ON A MULTIDROP LINE.

THERE ARE FOUR CONTROL MESSAGES USED BY THIS DIAGNOSTIC: ACK, NAK, STRT AND STACK. ACK IS AN ACKNOWLEDGEMENT THAT THE LAST MESSAGE SENT WAS PROPERLY RECEIVED. NORMALLY, MESSAGES ARE ACKNOWLEDGED BY SENDING A DATA MESSAGE WITH THE NUMBER OF THE LAST MESSAGE IN THE HEADER. ACK'S ARE USED WHEN THERE IS NO DATA MESSAGE TO SEND BACK. NAK IS A NEGATIVE ACKNOWLEDGEMENT FOR THE LAST MESSAGE SENT AND IT INDICATES THE REASON FOR THE ERROR. STRT IS USED TO INITIATE COMMUNICATION BETWEEN TWO STATIONS. THE RECEIVING STATION SETS ITSELF UP AND ACKNOWLEDGES THE STRT WITH A STACK, OR START ACKNOWLEDGE. THE STRT-STACK SEQUENCE IS USED BEFORE AND AFTER EACH TEST. ALL MESSAGES ARE REQUIRED TO BE ACKNOWLEDGED WITH EITHER A DATA MESSAGE OR AN ACK.

2.0

EQUIPMENT REQUIREMENTS

THIS PROGRAM REQUIRES A PDP-11 WITH A MINIMUM OF 16K CORE AND A CONSOLE TERMINAL. A CLOCK OPTION IS NOT REQUIRED, BUT ONLY ROUGH TIMING IS AVAILABLE WITHOUT ONE. THE PDP-11 CONSOLE SWITCH REGISTER IS NOT USED. ONE OR MORE DZ-11'S MUST BE CONNECTED TO THE TERMINALS UNDER TEST VIA COMMUNICATION LINES THROUGH EITHER MODEM'S OR NULL MODEM'S. THE VT62'S MUST BE IN ASYNCHRONOUS MODE AND ON-LINE. THE TERMINAL ADDRESS SWITCHES ON THE CONTROL BOARD MUST BE SET TO MATCH THE TERMINAL ADDRESS ASSIGNED TO THE LOGICAL UNIT (SEE SECTION 4.0).

3.0

LOADING PROCEDURE

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM PAPER TAPE OR DISK USING XXDP PROCEDURES.

4.0

STARTING AND RUNNING THE PROGRAM

THE STARTING ADDRESS FOR THE PROGRAM IS 200. THE PROGRAM WILL FIRST ASK IF A CLOCK (KW11-L OR KW11-P) IS AVAILABLE. IT WILL THEN ASK IF A LINE PRINTER IS AVAILABLE. IF A LINE PRINTER IS SPECIFIED, THE PROGRAM WILL

191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244

ASK FOR ITS ADDRESS (DEFAULT = 177514). THE PROGRAM ALSO ASKS FOR MEMORY SIZE, Q-BUS OPTION AND LINE FREQUENCY. IF NEITHER CLOCK IS SPECIFIED, THE PROGRAM WILL ASK THE USER TO TYPE TWO CHARACTERS, FIVE SECONDS APART, IN ORDER TO PROVIDE ROUGH TIMING.

THE PROGRAM WILL THEN PROMPT THE USER BY TYPING A '>'. NOW THE USER MAY TYPE ONE OF SEVEN COMMANDS: STA(RT), RES(TART), CON(TINUE), ZFL(AG), ADD, DRO(P) OR DIS(PLAY), WHERE THE LETTERS IN PARENTHESIS NEED NOT BE TYPED. IF A TYPING MISTAKE IS MADE, THE RUBOUT KEY IS USED TO DELETE THE LAST CHARACTER(S) TYPED. THE PROGRAM WILL TYPE A MESSAGE IF THE COMMAND IS INVALID. EACH OF THESE COMMANDS IS EXPLAINED IN THE SECTIONS BELOW. SWITCHES ARE USED TO SET SOFTWARE FLAGS SUCH AS 'LOOP ON ERROR'. THE FLAGS USED BY THIS PROGRAM ARE DESCRIBED IN SECTION 4.8.

4.1

START
THE START COMMAND (STA) ZEROS ALL FLAGS (PRINT ALL ERROR MESSAGES AND DO NOT LOOP WITHIN TESTS) AND INITIATES THE OPERATOR DIALOGUE. AT THIS POINT, THE DEVICE TABLE (PTABLE) IS EMPTY AND THE OPERATOR MUST USE THE DIALOGUE TO DESCRIBE ALL UUT'S AS SHOWN BELOW. THE NUMBERS IN SQUARE BRACKETS, [], RELATE THE TYPEOUT TO THE EXPLANATION BELOW AND ARE NOT TYPED IN ACTUAL OPERATION. THE PROCEDURES FOR HANDLING TYPING ERRORS ARE ALSO DESCRIBED BELOW. CR REFERS TO THE RETURN KEY (CARRIAGE RETURN).

DIALOGUE

UNITS ? (DECIMAL) [1]
DZ ADDRESS: ? (OCTAL) NNNNNN [2]
DZ VECTOR: ? (OCTAL) NNN [3]
DZ BUS PRIORITY: ? (OCTAL) N [4]
TERMINAL ADDRESS: ? (OCTAL) N [5]
LINE NUMBER: ? (OCTAL) N [6]
BAUD RATE: ? (OCTAL) N [7]
STOP CODE/CHAR. LENGTH: ? (OCTAL) N [8]
CHANGE S.W. ? (Y OR N) [9]
HOW MANY INITIATE TRIES BEFORE ABORT ? (OCTAL) 10 [10]
USE ALL ZERO'S FOR LOOPBACK DATA ? (Y OR N) Y [11]

EXPLANATION

[1] TYPE THE TOTAL NO. OF UUT'S IN DECIMAL. QUESTIONS 2 THROUGH 5 WILL BE REPEATED UNTIL ALL UUT'S HAVE BEEN

245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298

DESCRIBED. THE USER DOES NOT HAVE TO ANSWER THESE QUESTIONS FOR EACH DZ HOWEVER, IF THERE ARE MULTIPLE UUT'S PER DZ OR PER DZ LINE. THE EXPLANATION FOR QUESTION 5 SHOWS HOW TO ENTER MULTIPLE UUT'S.

- [2] TYPE THE BUS ADDRESS OF THE DZ. THIS MUST BE AN EVEN OCTAL NUMBER NOT GREATER THAN 177776. THE USER MAY TYPE CR TO USE THE DEFAULT VALUE: NNNNNN.
- [3] TYPE THE VECTOR ADDRESS FOR THE DZ. THIS MUST BE AN EVEN OCTAL NUMBER NOT GREATER THAN 376. THE USER MAY TYPE CR TO USE THE DEFAULT VALUE: NNN.
- [4] TYPE THE BUS PRIORITY LEVEL OF THE DZ. THIS MUST BE A DIGIT FROM 0 TO 7. AGAIN, THE USER MAY TYPE CR TO GET THE DEFAULT VALUE: N.
- [5] TYPE THE TERMINAL ADDRESS FOR EACH TERMINAL ATTACHED TO THE DZ LINE. SEPARATE EACH ADDRESS WITH A COMMA. THIS IS THE ADDRESS SELECTED BY THE SWITCHES ON THE I/O BOARD IN THE TERMINAL AND IS AN OCTAL NUMBER BETWEEN 0 AND 377. THE USER MAY TYPE CR TO GET THE DEFAULT VALUE: N.
- [6] TYPE THE DZ LINE NUMBER (0-7) TO WHICH THE TERMINAL IS ATTACHED. TYPE CR TO GET THE DEFAULT OF N.
- [7] TYPE THE CODE FOR THE BAUD RATE DESIRED. THE CODES ARE GIVEN IN THE TABLE BELOW. TYPE CR TO GET THE DEFAULT: N.

BAUD RATE TABLE

TYPE	TO GET THIS RATE
0	50
1	75
2	110
3	134.5
4	150
5	300
6	600
7	1200
10	1800
11	2000
12	2400
13	3600
14	4800
15	7200
16	9600
17	19,200

FOR FASTER TESTING, USE AS HIGH A BAUD RATE AS PRACTICAL. A BAUD RATE OF 19,200 IS NOT AVAILABLE ON THE VT62.

- [8] TYPE A CODE TO SELECT THE STOP CODE AND CHARACTER LENGTH. THE CODES ARE GIVEN IN THE TABLE BELOW.

TYPE	STOP CODE	CHARACTER LENGTH
0	1 BIT	5 BITS

299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351

1	::	6	::
2	::	7	::
3	::	8	::
4	2 BITS	5	::
5	::	6	::
6	::	7	::
7	::	8	::

- [9] IF THE USER WISHES TO CHANGE EITHER OF THE TWO SOFTWARE PARAMETERS DESCRIBED FOR QUESTIONS 7 OR 8, HE MUST ANSWER THIS QUESTION BY TYPING 'Y'. IF THE USER WISHES TO BYPASS QUESTIONS 7 AND 8, HE TYPES 'N'.
- [10] TYPE THE NUMBER OF TIMES THE PROGRAM IS TO ATTEMPT TO INITIALIZE A DUP BEFORE ABORTING THE UNIT. THE USER MAY TYPE ANY OCTAL NUMBER BETWEEN 0 AND 177777. TYPE CR TO DEFAULT TO 10 (OCTAL) ATTEMPTS.
- [11] IF THE USER TYPES 'Y', THE LOOPBACK DATA WILL BE 255 ZERO BYTES. IF THE USER TYPES 'N', THE LOOPBACK DATA WILL BE A PATTERN OF ALTERNATING 1 AND 0 BITS. SEE SECTION 8.2 FOR DETAILS ON THE LOOPBACK TEST.

EXAMPLE OF DIALOGUE

ASSUME THE FOLLOWING HARDWARE CONFIGURATION: A DZ (160000,300,3) WITH TWO TERMINALS (ADDRESSES 0 AND 1) ON LINE 0, A SECOND DZ (160010,320,3) WITH A SINGLE TERMINAL (ADDRESS 5) ON LINE 2 AND A THIRD DZ (170000,340,4) WITH THREE TERMINALS (ADDRESSES 0) ON LINES 1, 2 AND 3. NOTE THAT VARIABLE BAUD RATES CAN BE SPECIFIED. THE USER WOULD USE THE FOLLOWING DIALOGUE TO SET UP THE PROGRAM:

```

>STA
# UNITS ? (DECIMAL) 6
DZ ADDRESS: ? (OCTAL) 160000
DZ VECTOR: ? (OCTAL) 310 300
DZ BUS PRIORITY: ? (OCTAL) 4 3
TERMINAL ADDRESS: ? (OCTAL) 0 0,1
LINE NUMBER: ? (OCTAL) 0
BAUD RATE: ? (OCTAL) 0 16
STOP CODE/CHAR. LENGTH (SEE DOC.): ? (OCTAL) 0 7
DZ ADDRESS: ? (OCTAL) 160000 160010
DZ VECTOR: ? (OCTAL) 300 320

```

352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405

DZ BUS PRIORITY: ? (OCTAL) 3
TERMINAL ADDRESS: ? (OCTAL) 1 5
LINE NUMBER: ? (OCTAL) 0 2
BAUD RATE: ? (OCTAL) 16 7
STOP CODE/CHAR. LENGTH (SEE DOC.): ? (OCTAL) 7
DZ ADDRESS. ? (OCTAL) 160010 170000
DZ VECTOR: ? (OCTAL) 320 340
DZ BUS PRIORITY: ? (OCTAL) 3 5
TERMINAL ADDRESS: ? (OCTAL) 5 0
LINE NUMBER: ? (OCTAL) 2 1-3
BAUD RATE: ? (OCTAL) 7 7,16,7
STOP CODE/CHAR. LENGTH (SEE DOC.): ? (OCTAL) 7
CHANGE S. W. ? (Y OR N) N

(NOTE WHERE CR WAS USED TO ACCEPT THE DEFAULT)

TYPING ERRORS

- (1) THE RUBOUT KEY IS USED TO DELETE ANY CHARACTERS WHICH ARE IN ERROR.
- (2) IF THE RESPONSE IS NOT WITHIN LIMITS OR IS IN THE WRONG FORMAT, THE PROGRAM WILL NOTIFY THE USER AND ASK THE QUESTION AGAIN.
- (3) WHEN ENTERING MULTIPLE TERMINAL ADDRESSES FOR A SINGLE DZ, THE USER MAY NEGLECT TO TYPE ONE OR MORE ADDRESSES. IN SUCH A CASE, THE USER MERELY ENTERS THE SAME DZ ADDRESS, VECTOR AND PRIORITY IN THE NEXT SERIES OF QUESTIONS AND THEN ENTERS THE REMAINING TERMINAL ADDRESSES. REMEMBER; THE SERIES OF QUESTIONS (2 THROUGH 5) WILL BE RE-ASKED UNTIL ALL UUT'S HAVE BEEN ACCOUNTED FOR.

4.2

RESTART

THE RESTART COMMAND (RES) STARTS THE PROGRAM WITHOUT CLEARING THE TABLE. ALL FLAGS REVERT TO THEIR INITIAL STATE UNLESS THE FLAG SWITCH IS USED TO CHANGE THEM (SECTION 4.8). THE OPERATOR DIALOGUE DESCRIBED IN 4.1 DOES NOT TAKE PLACE UNLESS REQUESTED. WHEN THIS COMMAND IS USED, THE PROGRAM WILL ASK IF THE USER WISHES TO CHANGE THE HARDWARE TABLE AND IF HE WISHES TO CHANGE THE SOFTWARE TABLE. IF THE USER ANSWERS YES (BY TYPING 'Y') TO THE HARDWARE TABLE QUESTION, THAT PORTION OF THE DIALOGUE DESCRIBED IN

406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459

SECTION 4.1 BEGINNING WITH QUESTION 1 ENDING WITH QUESTION 5 WILL TAKE PLACE. IF THE USER ANSWERS YES TO THE SOFTWARE TABLE QUESTION, HE WILL BE ASKED QUESTIONS 7 AND 8.

THE MOST COMMON USE OF THIS COMMAND IS TO CHANGE RUN-TIME FLAGS OR TO ADD/DROP UNITS. THESE FLAGS ARE DESCRIBED IN SECTION 4.8 - SWITCH OPTIONS. TYPING A '^C' WILL FORCE THE SUPERVISOR TO ISSUE A PROMPT CHARACTER ('>') SO THAT THE USER MAY USE THE RESTART COMMAND FOR THIS PURPOSE. NOTE: IF THE USER USES THE START COMMAND AFTER A ^C, HE WILL HAVE TO REENTER ALL DEVICE INFORMATION!

4.3 CONTINUE
THE CONTINUE COMMAND WILL RESTART THE PROGRAM AT THE TEST WHICH WAS INTERRUPTED. UNLIKE THE RESTART COMMAND, THIS COMMAND DOES NOT FORCE FLAGS BACK TO THE INITIAL STATE. THE FLAGS WILL REMAIN AS SET UP DURING THE LAST START OR RESTART. THE USER HAS THE SAME DIALOGUE AND FLAG OPTIONS AS DESCRIBED FOR THE RESTART COMMAND.

4.4 ZERO FLAGS
THIS COMMAND WILL ZERO ALL FLAGS (SECTION 4.8) AND TYPE A PROMPT. SECTION 4.8 DESCRIBES THE EFFECTS OF CLEARING ALL FLAGS, BUT ESSENTIALLY, ALL LOOPING AND HALTS WILL BE DISABLED AND ALL ERROR INFORMATION WILL BE PRINTED IN THE EVENT OF AN ERROR.

4.5 ADD UNITS
WHILE TESTING MULTIPLE UNITS, THE USER MAY WISH TO DROP A PARTICULAR UNIT FROM THE TEST PROCEDURE AND, LATER ON, BEGIN TESTING THAT UNIT AGAIN. IN THIS CASE, THE USER NEED NOT STOP THE DIAGNOSTIC AND REBUILD THE PTABLE. AFTER A UNIT HAS BEEN DROPPED (SEE SECTION 4.6), THE USER MAY ADD THE UNIT BY TYPING '^C' AND USING THE ADD COMMAND AS FOLLOWS:

ADD/UNI:N

WHERE N IS A DECIMAL UNIT NUMBER THAT SPECIFIES THE UNIT TO BE ADDED. UNITS ARE NUMBERED FROM 1 TO 32 ACCORDING TO THE ORDER IN WHICH THEY WERE SPECIFIED FOR THE HARDWARE TABLE. MULTIPLE UNITS MAY BE ADDED BY TYPING EACH UNIT NUMBER SEPARATED BY COLONS (I.E., ADD/UNI:2:4:22) OR BY TYPING THE FIRST AND LAST UNIT NUMBERS OF A CONTIGUOUS GROUP OF UNITS SEPARATED BY A DASH (I.E., ADD/UNI:1-10 WILL ADD THE FIRST TEN UNITS). NOTE: AFTER A START COMMAND, ALL UNITS ARE CONSIDERS ADDED. THERE IS NO NEED TO USE THE ADD COMMAND UNLESS THE UNIT WAS DROPPED.

4.6 DROP UNITS
TO REMOVE A UNIT FROM TESTING, THE DROP COMMAND (DRO) IS USED. TO USE THIS COMMAND, TYPE '^C' AND DROP THE UNIT(S) USING THE DROP COMMAND AS FOLLOWS:

DRO/UNI:N

WHERE 'N' IS A DECIMAL UNIT NUMBER WHICH SPECIFIES THE UNIT TO BE DROPPED. UNITS ARE NUMBERED FROM 1 TO 32 ACCORDING TO THE ORDER IN WHICH THEY WERE SPECIFIED FOR THE HARDWARE TABLE. MULTIPLE UNITS MAY BE DROPPED BY TYPING EACH UNIT NUMBER SEPARATED BY COLONS (I.E., DRO/UNI:1:3:7) OR BY TYPING THE FIRST AND LAST UNIT NUMBERS OF A CONTIGUOUS GROUP SEPARATED BY A DASH (I.E., DRO/UNI:1-9 CAUSES THE FIRST

460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

NINE UNITS TO BE DROPPED). AFTER THE PROGRAM IS RESTARTED, THE SPECIFIED UNIT(S) WILL NOT BE INCLUDED FOR TESTING.

4.7 DISPLAY PTABLE
WHEN THE DISPLAY COMMAND IS TYPED, THE ENTIRE CONTENTS OF THE PTABLE WILL BE PRINTED. UNITS THAT HAVE BEEN DROPPED WILL BE SO MARKED. THIS COMMAND IS USEFUL FOR VERIFYING THE PTABLE CONTENTS.

4.8 SWITCH OPTIONS
THERE ARE THREE SWITCH OPTIONS AVAILABLE FOR USE WITH THE COMMANDS: /TES:, /PAS: AND /FLA:. THE TEST SWITCH, /TES:, IS USED ONLY WITH THE START COMMAND AND THE RESTART COMMAND. IT IS USED TO SPECIFY THE INDIVIDUAL TEST(S) TO BE RUN IN CASES WHERE ALL TESTS ARE NOT TO BE RUN. FOR EXAMPLE, IF THE USER WISHED TO RUN THE LOOPBACK TEST ONLY, HE WOULD TYPE:

>STA/TES:2

IF THIS SWITCH IS NOT USED, ALL TESTS WILL BE RUN AS THE DEFAULT CONDITION.

THE PASS SWITCH IS USED TO SPECIFY THE NUMBER OF PASSES TO BE RUN. TO RUN THE LOOPBACK TEST TWENTY TIMES, THE USER WOULD TYPE:

>STA/TES:2/PAS:20

THIS SWITCH MAY BE USED WITH THE START, RESTART AND CONTINUE COMMANDS. THE DEFAULT CONDITION IS ESSENTIALLY AN INFINITE NUMBER OF PASSES.

THE FLAG SWITCH IS USED TO SET FLAGS WHICH CONTROL PROGRAM OPERATION AND ERROR PRINTOUTS. THIS SWITCH MAY BE USED WITH THE CONTINUE COMMAND AS WELL AS THE START COMMAND AND RESTART COMMAND. THE AVAILABLE FLAGS ARE SHOWN IN THE TABLE BELOW.

FLAG	EFFECT IF SET
LOE	LOOP ON ERROR
HLT	HALT ON ERROR
IER	INHIBIT ERROR REPORTS
IBE	INHIBIT BASIC ERROR REPORT*
IXE	INHIBIT EXTENDED ERROR REPORT*
PRI	SEND ALL TYPEOUTS TO THE PRINTER
BOE	BELL ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
PNT	PRINT TEST NUMBER

*SEE SECTION 5.0

EXAMPLE:
IF THE USER WISHES TO RUN THE PROGRAM WITHOUT ERROR MESSAGES AND HAVE IT LOOP ON ERROR, HE WOULD TYPE:

>RES/FLA:LOE:IER

EXAMPLE:
IF THE USER LATER WANTED TO STOP THE LOOP ON ERROR FEATURE, HE WOULD TYPE:

>CON/FLA:LOE-0

NOTE: THE ERROR REPORTS WOULD STILL BE INHIBITED.

514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567

ALL FLAGS ARE DEFAULTED TO ZERO (NOT SET).

5.0

ERROR AND INFORMATIONAL MESSAGES

AS NOTED IN THE DESIGN NOTES (SECTION 1.3), A SINGLE ERROR MAY RESULT IN SEVERAL MESSAGES BEING TYPED. THIS IS DUE TO THE STRUCTURE OF THE PROGRAM. EACH SECTION, IN THE EVENT OF AN ERROR, REPORTS THE INFORMATION AVAILABLE TO IT. THE FINAL MESSAGE FOR EACH ERROR CONTAINS THE UNIT NUMBER (DECIMAL 1 1 TO 32 - SEE SECTIONS 4.5 AND 4.6) AND THE PHRASE: "END OF ERROR REPORT".

THERE ARE THREE LEVELS OF ERROR REPORTS. BY USING THE FLAG SWITCH (SECTION 4.8), ONE OR MORE LEVELS OF REPORT MAY BE INHIBITED. THE FIRST LEVEL CONSISTS OF AN ERROR NUMBER, PC CONTENTS AND ERROR TYPE (HARD, SOFT OR FATAL). THE ERROR NUMBER CAN BE RELATED TO THE TABLE AT THE END OF THIS SECTION. THE PC CONTENTS GIVE THE ADDRESS WHERE THE ERROR REPORT MECHANISM WAS INVOKED. THE NEXT LEVEL (BASIC ERROR REPORT) GIVES AN ENGLISH DESCRIPTION OF THE ERROR. AN EXAMPLE OF THIS TYPE OF MESSAGE IS: 'HEADER HAS CRC ERROR'. THE FINAL LEVEL (EXTENDED ERROR REPORT) GIVES ADDITIONAL DATA CONCERNING THE ERROR, SUCH AS REGISTER CONTENTS.

TO INHIBIT SOME OR ALL ERROR MESSAGES, USE THE FLAGS DESCRIBED IN SECTION 4.8. THE INHIBIT-EXTENDED-ERROR-REPORT FLAG (IXE) INHIBITS THAT LEVEL ONLY. THE INHIBIT-BASIC-ERROR-REPORT FLAG (IBE) INHIBITS BOTH BASIC AND EXTENDED ERROR REPORTS. FINALLY, THE INHIBIT-ERROR-REPORT FLAG (IER) INHIBITS ALL TYPEOUTS.

TABLE OF ERROR NUMBERS

NO.	ERROR DESCRIPTION
00	CANNOT INITIALIZE DZ
01	REPLY MESSAGE FROM TERM. WAS NOT AN ENQ
02	REPLY FROM TERM. WAS NOT EXPECTED TYPE OF CONTROL MESSAGE
03	TERM. ADDRESS IN RESPONSE WAS NOT COORECT
04	RESPONSE HAD CRC ERROR IN THE HEADER
20	DZ TRANSMIT ERROR
21	DZ RECEIVE EPORR
22	MESSAGE TO TERMINAL WAS NOT ACK'ED
23	ERROR OCCURRED WHILE INITIALIZING DZ
24	ERROR OCCURRED DURING TRANSMISSION
25	DDCMP STRT MESSAGE DID NOT GET A STACK
26	ERROR DURING RECEPTION
31	TIMEOUT WHILE WAITING FOR RESPONSE
40	SEE 21
41	SEE 04
42	SEE 03
43	MESSAGE DATA HAD CRC ERROR
50	ERROR OCCURRED DURING SELFTTEST (TEST #1)
51	ERROR OCCURRED DURING LOOPBACK (TEST #2)
52	ERROR IN RECEIVED DATA

568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621

70 SELFTEST RESPONSE IN ERROR (NOT RECOGNIZED)
71 TERMINAL MEMORY ERROR
72 TERMINAL CONTROL-ROM ERROR

6.0 RESTRICTIONS

THIS DIAGNOSTIC DOES NOT ATTEMPT TO TEST THE DZ11. IF A LARGE NUMBER OF COMM ERRORS OCCUR, OR IF ATTEMPTS TO SEND OR RECEIVE DATA TO AND FROM THE TERMINAL FAIL; USE THE SPECIFIC DZ11 DIAGNOSTICS TO VERIFY THE DZ11.

7.0 MISCELLANEOUS

PLEASE REPORT ALL PROGRAM PROBLEMS VIA THE AIDS REPORTING SYSTEM. REFER ALL COMMENTS REGARDING DOCUMENTATION TO DIAGNOSTIC ENGINEERING TERMINAL GROUP IN MAYNARD, MASS.

8.0 TEST DESCRIPTIONS

8.1 MICRODIAGNOSTIC (TEST #1)

THE MICRODIAGNOSTIC, OR SELFTEST, IS ROM RESIDENT CODE IN THE VT62 THAT IS EXECUTED UPON COMMAND FROM THE HOST COMPUTER OR DURING POWER UP. A FULL DESCRIPTION OF THE TEST IS AVAILABLE IN THE HARDWARE MANUAL FOR THE VT62. ESSENTIALLY, THE TEST VERIFIES THE MICROPROCESSOR, CHECKS ROM MEMORY, EXERCISES RAM MEMORY AND EXERCISES THE SCREEN. THE HOST MAY REQUEST THE RESULTS OF THE TEST. IN ANY CASE, THE RESULTS ARE ALWAYS DISPLAYED ON THE LAST LINE OF THE SCREEN. THE FIRST CHARACTERS ARE A CODED ERROR REPORT FOR THE HOST COMPUTER AND THE REMAINDER OF THE LINE IS AN ENGLISH DESCRIPTION OF THE ERROR STATUS.

8.2 LOOPBACK (TEST #2)

IN THIS TEST, THE TERMINAL IS COMMANDED TO ENTER LOOPBACK MODE AND A SINGLE, 256(DECIMAL) BYTE DATA MESSAGE IS SENT TO THE TERMINAL. THIS DATA MESSAGE WILL HAVE A DDCMP PROTOCOL ENVELOPE. WHEN THE DATA IS RETURNED, THE TERMINAL IS TAKEN OUT OF LOOPBACK MODE USING A DDCMP 'STRT' SEQUENCE. THE CRC CHECKSUMS ARE VALIDATED FOR BOTH THE MESSAGE HEADER AND THE MESSAGE DATA. THE DATA IS THEN CHECKED, BYTE BY BYTE. THUS THERE ARE THREE POSSIBLE ERRORS DETECTED: NO TERMINAL RESPONSE, BAD CRC OR BAD DATA.

INTERPRETATION OF ANY ERROR MESSAGES REQUIRES SOME EDUCATED GUESSWORK BECAUSE OF THE NUMBER OF COMPONENTS IN THE COMMUNICATION LINK. INTERMITTENT ERRORS WOULD GENERALLY INDICATE LINE PROBLEMS UNLESS THE LINE IS PHYSICALLY SHORT OR THE ENVIRONMENT IS RELATIVELY NOISE FREE. IF EITHER OF THESE CONDITIONS IS THE CASE OR IF THE ERROR IS NOT INTERMITTENT, THEN THE TERMINAL I/O BOARD, THE MODEM AND THE DUP ARE SUSPECT. IF THERE IS NO TERMINAL RESPONSE, RUN THE TERMINAL SELF TEST OFF-LINE. THIS WILL SHOW WHETHER THE TERMINAL IS COMPLETELY 'DEAD' OR THE PORTIONS OF THE COMMUNICATION LINK MENTIONED ABOVE ARE BROKEN.

THERE ARE TWO DATA PATTERNS AVAILABLE FOR LOOPBACK: ALL ZERO'S AND ALTERNATING 1/0 BITS (I.E., BYTES OF 252,125,252,25,...). THE ZERO

622
623
624
625
626
627

PATTERN IS NOISE-SUSCEPTIBLE. HOWEVER THE ALTERNATING BITS PATTERN EXERCISES THE DATA PATHS BETTER. SECTION 4.0 SPECIFIES HOW TO SELECT THE PATTERN.

628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672

9.0

INDEX

ACK	5, 12
ADD COMMAND	10
BAUD RATE	7
CONTINUE COMMAND	10-11
CRC	4-5, 12
DDCMP	4-5
DISPLAY COMMAND	11
DROP COMMAND	10
ENQ	5, 12
ERROR NUMBER	12
ERROR REPORT	12
FLAGS	6, 9-12
LOOPBACK	8, 12-13
MULTIPLE UNITS	10
MULTIPLE UUT'S	7
NAK	5
PTABLE	4, 10-11
RESTART COMMAND	9-11
STACK	5, 12
START COMMAND	6, 10-11
STOP CODE	7
STRT	5, 12
SWITCH OPTIONS	10
SWITCH OPTIONS	11
SWITCHES	6
UNIT NUMBER	10, 12

.FNDR

673 000000'

.PRINT ;SPMAC: VERSION 1.1

674	002000		.=2000	
675				
676		:	HEADER - VT62 ACCEPTANCE TEST	
677		:		
678	002000		POINTER ALL	
679	002000		BGNMOD HEAD	
680	002000		HEADER CZDZC,B,0,0,0,0	
681	002000	103		.ASCII /C/
682	002001	132		.ASCII /Z/
683	002002	104		.ASCII /D/
684	002003	132		.ASCII /Z/
685	002004	103		.ASCII /C/
686	002005	000		.BYTE 0
687	002006	000		.BYTE 0
688	002007	000		.BYTE 0
689	002010	102		.ASCII /B/
690	002011	060		.ASCII /O/
691	002012	000000		.WORD 0
692	002014	000000		.WORD 0
693	002016	022020		.WORD L\$HARD
694	002020	022354		.WORD L\$SOFT
695	002022	011672		.WORD L\$HW
696	002024	011706		.WORD L\$SW
697	002026	022612		.WORD L\$LAST
698	002030	000000		.WORD 0
699	002032	000000		.WORD 0
700	002034	000000		.WORD 0
701	002036	000000		.WORD 0
702	002040	011716		.WORD L\$DISPATCH
703	002042	000000		.WORD 0
704	002044	000000		.WORD 0
705	002046	000000		.WORD 0
706	002050	002		.BYTE C\$REVISION
707	002051	002		.BYTE C\$EDIT
708	002052	000000		.WORD 0
709	002054	000000		.WORD 0
710	002056	000000		.WORD 0
711	002060	000000		.WORD 0
712	002062	000000		.WORD 0
713	002064	002234		.WORD L\$DVTYP
714	002066	011722		.WORD L\$RPT
715	002070	002250		.WORD L\$DR
716	002072	002254		.WORD L\$DRST
717	002074	011724		.WORD L\$AU
718	002076	011726		.WORD L\$DU
719	002100	000000		.WORD 0
720	002102	000000		.WORD 0
721	002104	011730		.WORD L\$INIT
722	002106	012322		.WORD L\$CLEAN
723	002110		ENDMOD	

```

724 002110          BGNMOD GLOB
725                :*****
726                :GLOBAL EQUATES
727                :*****
728 002110          EQUALS
729
730                :
731                :
732                :
733                :
734                :
735                :
736                :
737                :
738                :
739                :
740                :
741                :
742                :
743                :
744                :
745                :
746                :
747                :
748                :
749                :
750                :
751                :
752                :
753                :
754                :
755                :
756                :
757                :
758                :
759                :

```

000201	SOH=201	:START OF HEADER (DDCMP)
000001	ACK=1	:ACKNOWLEDGE
000002	NAK=2	:NEGATIVE ACKNOWLEDGE
000003	REP=3	:REPLY
000005	ENQ=5	:ENQUIRY
000006	STRT=6	:START
000007	STACK=7	:START ACKNOWLEDGE
000377	SYNC=377	:SYNCHRONIZATION CHARACTER
000220	DLE 220	:DATA LINK ESCAPE.
000030	OSOP=24	:LOOPBACK.
000200	PARITY=200	:PARITY BIT.
000002	TSTS=2	:NO. OF TESTS
000000	RCVCSR 0	:OFFSETS: RECEIVER CSR
000002	RCVBUF=2	:RECEIVER BUFFER REG.
000004	XMTCSR=4	:TRANSMITTER CSR
000006	XMTBUF 6	:TRANSMITTER BUFFER REG.
		:HARDWARE REGISTER BIT DEFINITIONS
000020	MASCLR 20	:MASTER CLEAR
000040	MSCNEN=40	:MASTER SCAN ENABLE
000100	RXINTE=100	:RECEIVE INTERRUPT ENABLE
040000	TXINTE=40000	:TRANSMIT INTERRUPT ENABLE
100000	INVDAT=100000	:INVALID DATA
010000	RENABL=10000	:RECEIVE ENABLE
000017	LINEMK 17	:LINE BITS MASK
177400	HYBYTE=177400	:MASK FOR HIGH BYTE OF A WORD
000377	LOBYTE=377	:MASK FOR LOW BYTE OF A WORD

```

760
761
762
763 002110 000000
764 002112 000000
765 002114 000000
766 002116 000000
767 002120 002000
768 002122 000000
769 002124 000010
770 002126 000000
771 002130 000000
772 002132 000001
773 002134 000000
774 002136 000000
775 002140 000000
776 002142 000000
777 002144 000000
778 002146 000000
779 002150 000000
780 002152 000000
781 002154 000000
782 002156 000000
783 002160 000000
784
785 002162
786 002162 000000
787 002164 000000
788 002166 000000
789 002170 000000
790 002172 000000
791 002174 000000
792 002176 000000
793 002200 000000
794 002202 000000
795 002204 000000
796 002206 000000
797 002210 000000
798 002212
799 002212 000000
800 002214
801 002214 000001
802 002216 000002
803 002220 000004
804 002222 000010
805 002224 000020
806 002226 000040
807 002230 000100
808 002232 000200
809
810 002234
811 002234 052126 031066 042054
812 002242 030532 000061
813
814 002246
815 002246 000001

```

```

:*****
: GLOBAL DATA *
:*****
RSSTCK: 0 ;FLAG TO INDICATE IF R5 STACK IS SET UP
SCRATCH: 0 ;SCRATCH LOCATIONS
TEMP: 0
ABORT: 0 ;TEST ABORT FLAG
TIMLIM: 2000 ;TIMEOUT LIMIT (IN 100 MICROSEC. INCREMENTS)
STIME: 0 ;SHORT TIMEOUT LIMIT (APPROX. 20 MICROSEC. INCREM.)
ILIMIT: 10 ;LIMIT ON DEVICE INITIALIZATION ATTEMPTS
INERR: 0 ;INPUT ERROR
OUTERR: 0 ;OUTPUT ERROR
PASCNT: 1
EOMSG: 0 ;END OF MESSAGES FLAG
LOGDEV: 0 ;LOGICAL DEVICE NUMBER
EXPINP: 0 ;EXPECT INPUT FLAG
OUTMSG: 0 ;OUTPUT MESSAGE POINTER AND SIZE
INMSG: 0 ;INPUT MESSAGE POINTER AND SIZE
OUTBUF: 0 ;OUTPUT BUFFER POINTER AND SIZE
OUTSIZ: 0
INBUF: 0 ;INPUT BUFFER POINTER AND SIZE
INSIZ: 0
MSGAP:
OUT: 0 ;TEST MESSAGE AREA POINTER
INITMA: 0 ;TEMPORARY MESSAGE POINTER AND SIZE
INITMS: 0 ;USED FOR INITIALIZATION OF OUTPUT ROUTINES
BAPNT: 0 ;GENERAL BUFFER AREA POINTER AND SIZE
BASIZE: 0
MSGNO: 0 ;MESSAGE NUMBER (DDCMP)
RMSGNO: 0
PTABLE: 0 ;PRESENT UNIT PTABLE DATA
0
0
0
NMOD: 0
PTEND:
DEVPRI: 0 ;DEVICE PRIORITY
LINMSK: ;MASKS FOR LINE SELECTION BITS (DZ11)
1 ;LINE 0
2 ;LINE 1
4 ;LINE 2
10 ;LINE 3
20 ;LINE 4
40 ;LINE 5
100 ;LINE 6
200 ;LINE 7
DEVTYP <VT62,DZ11>
.ASCIZ /VT62,DZ11/
.EVEN
.DEVREG 4,'',DZREG,DZMSK
.WORD 1

```

816 002250 000017
817 002252 000004
818 002254 000004

.WORD 17
.WORD 4
.BLKW 4

```
819
820
821
822
823 002264 040503 020116 047516 MSG00: .ASCIZ /CAN NOT INITIALIZE COMM DEVICE/
824 002272 020124 047111 052111
825 002300 040511 044514 042532
826 002306 041440 046517 020115
827 002314 042504 044526 042503
828 002322 000
829 002323 122 050105 054514 MSG01: .ASCIZ /REPLY WAS NOT AN ENQ/
830 002330 053440 051501 047040
831 002336 052117 040440 020116
832 002344 047105 000121
833 002350 047111 047503 051122 MSG02: .ASCIZ /INCORRECT REPLY TYPE/
834 002356 041505 020124 042522
835 002364 046120 020131 054524
836 002372 042520 000
837 002375 103 051101 044522 MSG05: .ASCIZ /CARRIER NOT DETECTED AFTER 10MSEC/
838 002402 051105 047040 052117
839 002410 042040 052105 041505
840 002416 042524 020104 043101
841 002424 042524 020122 030061
842 002432 051515 041505 000
843 002437
844 002437 122 050105 054514 MSG42:
MSG03: .ASCIZ /REPLY HAS INCORRECT STATION ADDRESS/
845 002444 044040 051501 044440
846 002452 041516 051117 042522
847 002460 052103 051440 040524
848 002466 044524 047117 040440
849 002474 042104 042522 051523
850 002502 000
851 002503
852 002503 110 040505 042504 MSG41:
MSG04: .ASCIZ /HEADER HAS CRC ERROR/
853 002510 020122 040510 020123
854 002516 051103 020103 051105
855 002524 047522 000122
856 002530 051105 047522 020122 MSG'0: .ASCIZ /ERROR DURING KEYBOARD TEST ** END OF REPORT **/
857 002536 052504 044522 043516
858 002544 045440 054505 047502
859 002552 051101 020104 042524
860 002560 052123 020040 025052
861 002566 042440 042116 047440
862 002574 020106 042522 047520
863 002602 052122 025040 000052
864 002610 040502 020104 042522 MSG11: .ASCIZ /BAD RESPONSE FROM KEYBOARD/
865 002616 050123 047117 042523
866 002624 043040 047522 020115
867 002632 042513 041131 040517
868 002640 042122 000
869 002643 124 047517 046440 MSG12: .ASCIZ /TOO MANY CHARACTERS SENT FROM KEYBOARD/
870 002650 047101 020131 044103
871 002656 051101 041501 042524
872 002664 051522 051440 047105
873 002672 020124 051106 046517
874 002700 045440 054505 047502
```

875	002706	051101	000104		
876	002712	051124	047101	046523	MSG20: .ASCIZ /TRANSMIT ERROR: NO TRANSMIT INTERRUPTS/
877	002720	052111	042440	051122	
878	002726	051117	020072	047516	
879	002734	052040	040522	051516	
880	002742	044515	020124	047111	
881	002750	042524	051122	050125	
882	002756	051524	000		
883	002761				MSG40:
884	002761	122	041505	050105	MSG21: .ASCIZ /RECEPTION ERROR/
885	002766	044524	047117	042440	
886	002774	051122	051117	000	
887	003001	116	020117	041501	MSG22: .ASCIZ /NO ACK RECEIVED AFTER TRANSMISSION/
888	003006	020113	042522	042503	
889	003014	053111	042105	040440	
890	003022	052106	051105	052040	
891	003030	040522	051516	044515	
892	003036	051523	047511	000116	
893	003044	051105	047522	020122	MSG23: .ASCIZ /ERROR WHILE INITIALIZING COMM DEVICE/
894	003052	044127	046111	020105	
895	003060	047111	052111	040511	
896	003066	044514	044532	043516	
897	003074	041440	046517	020115	
898	003102	042504	044526	042503	
899	003110	000			
900	003111	105	051122	051117	MSG24: .ASCIZ /ERROR DURING COMM DEVICE TRANSMIT/
901	003116	042040	051125	047111	
902	003124	020107	047503	046515	
903	003132	042040	053105	041511	
904	003140	020105	051124	047101	
905	003146	046523	052111	000	
906	003153	116	020117	052123	MSG25: .ASCIZ /NO STACK IN RESPONSE TO STRT/
907	003160	041501	020113	047111	
908	003166	051040	051505	047520	
909	003174	051516	020105	047524	
910	003202	051440	051124	000124	
911	003210	051105	047522	020122	MSG26: .ASCIZ /ERROR DURING COMM DEVICE RECEIVE/
912	003216	052504	044522	043516	
913	003224	041440	046517	020115	
914	003232	042504	044526	042503	
915	003240	051040	041505	044505	
916	003246	042526	000		
917	003251	103	047101	052047	MSG30: .ASCIZ /CAN'T GET CLEAR TO SEND/
918	003256	043440	052105	041440	
919	003264	042514	051101	052040	
920	003272	020117	042523	042116	
921	003300	000			
922	003301	124	046511	020105	MSG31: .ASCIZ /TIME OUT: NO CHARACTERS FROM TERMINAL/
923	003306	052517	035124	047040	
924	003314	020117	044103	051101	
925	003322	041501	042524	051522	
926	003330	043040	047522	020115	
927	003336	042524	046522	047111	
928	003344	046101	000		
929	003347	124	046511	020105	MSG32: .ASCIZ /TIME OUT: MESSAGE FROM TERMINAL INCOMPLETE/
930	003354	052517	035124	046440	

```

931 003362 051505 040523 042507
932 003370 043040 047522 020115
933 003376 042524 046522 047111
934 003404 046101 044440 041516
935 003412 046517 046120 052105
936 003420 000105
937 003422 040504 040524 041440
938 003430 041522 042440 051122
939 003436 051117 000
940 003441 116 044517 042523
941 003446 050040 052101 042524
942 003454 047122 051440 041125
943 003462 042524 052123 000
944 003467 105 051122 051117
945 003474 042040 051125 047111
946 003502 020107 042523 043114
947 003510 042524 052123 000
948 003515 105 051122 051117
949 003522 042040 051125 047111
950 003530 020107 047514 050117
951 003536 040502 045503 000
952 003543 104 052101 020101
953 003550 051105 047522 000122
954 003556 046111 042514 040507
955 003564 020114 042523 043114
956 003572 052040 051505 020124
957 003600 042522 050123 047117
958 003606 042523 000
959 003611 115 046505 051117
960 003616 020131 051105 047522
961 003624 020122 020040 054524
962 003632 042520 040
963 003635 000 000
964 003637 115 026440 041040
965 003644 047101 020113 043
966 003651 000 000
967
968 003654 051103 046517 042440
969 003662 051122 051117 000
970
971
972 003670 005
973 003671 006
974 003672 300
975 003673 000
976 003674 000
977 003675 000
978 003676 000000
979
980
981
982
983 003700
984 003700
985 003700 016746 176220
986 003704 012746 003726

```

MSG43: .ASCIZ /DATA CRC ERROR/

MSG45: .ASCIZ /NOISE PATTERN SUBTEST/

MSG50: .ASCIZ /ERROR DURING SELFTEST/

MSG51: .ASCIZ /ERROR DURING LOOPBACK/

MSG52: .ASCIZ /DATA ERROR/

MSG70: .ASCIZ /ILLEGAL SELF TEST RESPONSE/

MSG71: .ASCII /MEMORY ERROR TYPE /

MTY: .BYTE 0,0
.ASCII /M - BANK #/

BNO: .BYTE 0,0
.EVEN

MSG72: .ASCIZ /CROM ERROR/

.EVEN

STRTB: .BYTE ENQ
.BYTE STRT
.BYTE 300

.BYTE 0
.BYTE 0

STRISA: .BYTE 0

STRIC: .WORD 0

: GLOBAL ERROR REPORT *
:*****

BGNMSG AREA00
PRINTX #FORM00,ILIMIT

MOV ILIMIT, -(SP)
MOV #FORM00, -(SP)

1043	004116				AREA41:				
1044	004116				AREA43:				
1045	004116					PRINTX #FORM04,A04RC,A04EX			
1046	004116	016746	000030				MOV	A04EX,-(SP)	
1047	004122	016746	000022				MOV	A04RC,-(SP)	
1048	004126	012746	004154				MOV	#FORM04,-(SP)	
1049	004132	012746	000003				MOV	#3,-(SP)	
1050	004136	010600					MOV	SP,R0	
1051	004140	104015					EMT	C\$PNTX	
1052	004142	062706	000010				ADD	#10,SP	
1053	004146					ENDMSG			
1054	004146						L10004:		
1055	004146	104023					EMT	C\$MSG	
1056	004150	000000			A04RC: 0				
1057	004152	000000			A04EX: 0				
1058	004154	040445	042522	042503	FORM04:	.ASCIZ /%ARECEIVED %06%A -- EXPECTED %06%N/			
1059	004162	053111	042105	022440					
1060	004170	033117	040445	026440					
1061	004176	020055	054105	042520					
1062	004204	052103	042105	022440					
1063	004212	033117	047045	000					
1064		004220				.EVEN			
1065									
1066	004220				BGNMSG AREA11				
1067	004220				PRINTX #FORM04,KRC,KEX				
1068	004220	016746	000030				MOV	KEX,-(SP)	
1069	004224	016746	000022				MOV	KRC,-(SP)	
1070	004230	012746	004154				MOV	#FORM04,-(SP)	
1071	004234	012746	000003				MOV	#3,-(SP)	
1072	004240	010600					MOV	SP,R0	
1073	004242	104015					EMT	C\$PNTX	
1074	004244	062706	000010				ADD	#10,SP	
1075	004250					ENDMSG			
1076	004250						L10005:		
1077	004250	104023					EMT	C\$MSG	
1078	004252	000000			KRC: 0				
1079	004254	000000			KEX: 0				
1080	004256				BGNMSG AREA21				
1081	004256				ENDMSG				
1082	004256						L10006:		
1083	004256	104023					EMT	C\$MSG	
1084	004260				BGNMSG AREA22				
1085	004260				ENDMSG				
1086	004260						L10007:		
1087	004260	104023					EMT	C\$MSG	
1088	004262				BGNMSG AREA23				
1089	004262				AREA24:				
1090	004262				AREA40:				
1091	004262					LET TEMP2 .= LOGDEV			
1092	004262	016767	175650	000064			MOV	LOGDEV,TEMP2	
1093	004270					IF TEMP2 LT #0 THEN			
1094	004270	005767	000060				TST	TEMP2	
1095	004274	002002					BGE	50000\$	
1096	004276					LET TEMP2 := #0			
1097	004276	005067	000052				CLR	TEMP2	
1098	004302					ENDIF			

1155	004536	012701	002254						MOV	#DZREG,R1
1156	004542			PRINTX	#FORM30,RCVCSR(R1)				MOV	RCVCSR(R1),-(SP)
1157	004542	016146	000000						MOV	#FORM30,-(SP)
1158	004546	012746	004666						MOV	#2,-(SP)
1159	004552	012746	000002						MOV	SP,R0
1160	004556	010600							EMT	C\$PNTX
1161	004560	104015							ADD	#6,SP
1162	004562	062706	000006							
1163	004566			PRINTX	#FRM30A,RCVBUF(R1)				MOV	RCVBUF(R1),-(SP)
1164	004566	016146	000002						MOV	#FRM30A,-(SP)
1165	004572	012746	004707						MOV	#2,-(SP)
1166	004576	012746	000002						MOV	SP,R0
1167	004602	010600							EMT	C\$PNTX
1168	004604	104015							ADD	#6,SP
1169	004606	062706	000006							
1170	004612			PRINTX	#FRM30B,XMTCSR(R1)				MOV	XMTCSR(R1),-(SP)
1171	004612	016146	000004						MOV	#FRM30B,-(SP)
1172	004616	012746	004731						MOV	#2,-(SP)
1173	004622	012746	000002						MOV	SP,R0
1174	004626	010600							EMT	C\$PNTX
1175	004630	104015							ADD	#6,SP
1176	004632	062706	000006							
1177	004636			PRINTX	#FRM30C,XMTBUF(R1)				MOV	XMTBUF(R1),-(SP)
1178	004636	016146	000006						MOV	#FRM30C,-(SP)
1179	004642	012746	004752						MOV	#2,-(SP)
1180	004646	012746	000002						MOV	SP,R0
1181	004652	010600							EMT	C\$PNTX
1182	004654	104015							ADD	#6,SP
1183	004656	062706	000006							
1184	004662			POP	R1				MOV	(SP)+,R1
1185	004662	012601								
1186	004664			ENDMSG						
1187	004664									
1188	004664	104023						L10012:	EMT	C\$MSG
1189	004666	040445	055104	051040	FORM30: .ASCIZ /%ADZ RCSR: %06%N/					
1190	004674	051503	035122	022440						
1191	004702	033117	047045	000						
1192	004707	045	042101	020132	FRM30A: .ASCIZ /%ADZ RBUF: %06%N/					
1193	004714	041122	043125	035106						
1194	004722	022440	033117	047045						
1195	004730	000								
1196	004731	045	042101	020132	FRM30B: .ASCIZ /%ADZ TCSR: %06%N/					
1197	004736	041524	051123	020072						
1198	004744	047445	022466	000116						
1199	004752	040445	055104	052040	FRM30C: .ASCIZ /%ADZ TBUF: %06%N/					
1200	004760	052502	043106	020072						
1201	004766	047445	022466	000116						
1202					.EVEN					
1203										
1204	004774				BGNMSG AREA60					
1205	004774				PRINTX #FORM60					
1206	004774	012746	005016						MOV	#FORM60,-(SP)
1207	005000	012746	000001						MOV	#1,-(SP)
1208	005004	010600							MOV	SP,R0
1209	005006	104015							EMT	C\$PNTX
1210	005010	062706	000004						ADD	#4,SP

1211	005014				ENDMSG		
1212	005014					L10013:	
1213	005014	104023					EMT C\$MSG
1214	005016	040445	040503	023516	FORM60: .ASCIZ /%ACAN'T GET INTO LOOPBACK%N/		
1215	005024	020124	042507	020124			
1216	005032	047111	047524	046040			
1217	005040	047517	041120	041501			
1218	005046	022513	000116				
1219							
1220	005052				.EVEN		
1221	005052				BGNMSG AREA61		
1222	005052	012746	005144		PRINTX #FORM61		
1223	005056	012746	000001				MOV #FORM61,-(SP)
1224	005062	010600					MOV #1,-(SP)
1225	005064	104015					MOV SP,R0
1226	005066	062706	000004				EMT C\$PNTX
1227	005072				LET TEMP2 := LOGDEV		ADD #4,SP
1228	005072	016767	175040	177254			MOV LOGDEV,TEMP2
1229	005100				IF TEMP2 LT #0 THEN		
1230	005100	005767	177250				TST TEMP2
1231	005104	002002					BGE 50001\$
1232	005106				LET TEMP2 := #0		
1233	005106	005067	177242				CLR TEMP2
1234	005112				ENDIF		
1235	005112					50001\$:	
1236	005112				LET TEMP2 := TEMP2 + #1		
1237	005112	005267	177236				INC TEMP2
1238	005116				PRINTX #FORM23,TEMP2		
1239	005116	016746	177232				MOV TEMP2,-(SP)
1240	005122	012746	004356				MOV #FORM23,-(SP)
1241	005126	012746	000002				MOV #2,-(SP)
1242	005132	010600					MOV SP,R0
1243	005134	104015					EMT C\$PNTX
1244	005136	062706	000006				ADD #6,SP
1245	005142				ENDMSG		
1246	005142					L10014:	
1247	005142	104023					EMT C\$MSG
1248	005144	040445	040503	023516	FORM61: .ASCIZ /%ACAN'T GET OUT OF LOOPBACK%N/		
1249	005152	020124	042507	020124			
1250	005160	052517	020124	043117			
1251	005166	046040	047517	041120			
1252	005174	041501	022513	000116			
1253							
1254	005202				.EVEN		
1255	005202				BGNMSG AREA52		
1256	005202	016746	000074		PRINTX #FORM52,A52A,A52B		
1257	005206	016746	000066				MOV A52B,-(SP)
1258	005212	012746	005234				MOV A52A,-(SP)
1259	005216	012746	000003				MOV #FORM52,-(SP)
1260	005222	010600					MOV #3,-(SP)
1261	005224	104015					MOV SP,R0
1262	005226	062706	000010				EMT C\$PNTX
1263	005232				ENDMSG		ADD #10,SP
1264	005232					L10015:	
1265	005232	104023					EMT C\$MSG
1266	005234	040445	054105	042520	FORM52: .ASCIZ /%AEXPECTED: %03%A -RECEIVED: %03%N/		

1267 005242 052103 042105 020072
1268 005250 047445 022463 020101
1269 005256 026440 042522 042503
1270 005264 053111 042105 020072
1271 005272 047445 022463 000116
1272
1273 005300 000000
1274 005302 000000
1275
1276 005304
1277 005304
1278 005304
1279 005304 104023
1280
1281 005306
1282 005306
1283 005306
1284 005306 104023
1285
1286 005310
1287 005310 001452
1288 010434
1289 010434 000416
1290 011470 000100
1291 011670
1292
1293 011670

.EVEN
A52A: 0
A52B: 0

BGNMSG AREA70
ENDMSG

L10016: EMT CSMSG

BGNMSG AREA71
ENDMSG

L10017: EMT CSMSG

TEMPBUF:
TMPBUF: .BLKW 810.
.BLKW 270.
.BLKW 100

SSTACK: ;STACK FOR SUBROUTINE LINKAGE.

ENDMCD

1294 011670
1295
1296
1297
1298 011670
1299 011670 000005
1300 011672 160010
1301 011674 000320
1302 011676 000640
1303 011700 007030
1304 011702 000001
1305 011704
1306 011704
1307 011704
1308 011704 000003
1309 011706 000010
1310 011710 000001
1311 011712 020000
1312 011714
1313 011714
1314
1315
1316
1317
1318 011714
1319 011714 000002
1320 011716 017466
1321 011720 020516
1322 011722

```
BGNMOD TABLE
:*****
:  DEFAULT HARDWARE P-TABLE AND SOFTWARE P-TABLE  *
:*****
  BGNHW
    160010
    320
    640
    7030
    1
  ENDPHW
  BGNSW
    000010
PATTERN:000^01
STWAIT: 20000
  ENDSW
```

```
.WORD L10020-L$HW/2
L10020:
.WORD L10021-L$SW/2
L10021:
.WORD TSTS
.WORD T1
.WORD T2
```

```
:*****
:  DISPATCH TABLE  *
:*****
DISPATCH TSTS
```

ENDMOD

```

1323 011722          BGNMOD CODE
1324
1325                :*****
1326                :STUB AREAS FOR SUPERVISOR COMPATIBILITY *
1327                :*****
1328 011722          BGNRPT
1329 011722          ENDRPT
1330
1331 011722 104025   L10022:  EMT      CSRPT
1332
1333 011724          BGNAU
1334 011724          ENDAU
1335
1336 011724 104054   L10023:  EMT      CSAU
1337
1338 011726          BGNDU
1339 011726          ENDDU
1340
1341 011726 104055   L10024:  EMT      CSDU
1342
1343                :*****
1344                :      INITIALIZE CODING
1345                :*****
1346 011730          BGNINIT
1347
1348                :SAVE R1
1349
1350                :
1351 011730 010146   PUSH R1
1352
1353                :
1354                :CHECK START EVENT FLAG TO SEE IF WE GOT HERE BECAUSE OF A START
1355                :
1356 011732 012700 000040  READEF #EF.START
1357 011736 104050   MOV      #EF.START,R0
1358                :
1359                :IF A START, THEN...
1360                :
1361 011740          IF.ERROR THEN
1362 011740 103005   BCC      50002$
1363
1364                :INITIALIZE THE SUBROUTINE LINKAGE STACK
1365                :
1366 011742          LET R5 := #SSTACK
1367 011742 012705 011670   MOV      #SSTACK,R5
1368 011746          LET R5STCK := R5STCK + #1
1369 011746 005267 170136   INC      R5STCK
1370 011752          ELSE
1371 011752 000414   BR       50003$
1372 011754          50002$:
1373
1374                :IF NOT START THEN....
1375                :
1376                :CHECK FOR RESTART AND IF IT IS:
1377                :UPDATE LOGICAL DEVICE AND STAT. TABLE ENTRY POINTER,
1378                :BUT DON'T EXCEED THE NUMBER OF UNITS SPECIFIED
  
```



```

1379      ;
1380      ;
1381      011754      ;
1382      011754      012700      000037      ;
1383      011760      104050      ;
1384      011762      103410      ;
1385      011764      ;
1386      011764      005267      170146      ;
1387      011770      ;
1388      011770      026767      170142      170014      ;
1389      011776      001002      ;
1390      012000      ;
1391      012000      005067      170132      ;
1392      012004      ;
1393      012004      ;
1394      012004      ;
1395      012004      ;
1396      012004      ;
1397      012004      ;
1398      ;
1399      ;
1400      ;
1401      012004      ;
1402      012004      012700      000035      ;
1403      012010      104050      ;
1404      012012      ;
1405      012012      103002      ;
1406      012014      ;
1407      012014      005067      170116      ;
1408      012020      ;
1409      012020      ;
1410      ;
1411      ;
1412      ;
1413      ;
1414      012020      ;
1415      012020      005067      003624      ;
1416      012024      ;
1417      012024      005067      170066      ;
1418      ;
1419      ;
1420      ;
1421      012030      ;
1422      012030      012767      015674      003634      ;
1423      012036      ;
1424      012036      016767      003630      003624      ;
1425      012044      ;
1426      012044      012767      015712      003636      ;
1427      012052      ;
1428      012052      016701      003632      ;
1429      012056      ;
1430      012056      012767      005310      170026      ;
1431      012064      ;
1432      012064      ;
1433      012064      ;
1434      012064      016721      170022      ;

      READEF #EF.RESTART
      IF .NO.ERROR THEN
          LET LOGDEV := LOGDEV + #1
          IF LOGDEV EQ L$UNIT THEN
              LET LOGDEV := #0
          ENDIF
      ENDIF
      50005$:
      50004$:
      50003$:
      ;CHECK FOR NEW PASS... IF IT IS, FORCE LOGICAL UNIT TO ZERO.
      READEF #EF.NEW
      IF .ERROR THEN
          LET LOGDEV := #0
      ENDIF
      50006$:
      ;END OF START CODE
      ;FORCE THE COMM DEVICE TO BE INITIALIZED
      LET INIT : #0
      LET ABORT := #0
      CLR INIT
      CLR ABORT
      ;SET UP THE INPUT BUFFERS QUEUE
      LET RQUE : #RQUET
      LET RQUEUE := RQUE
      LET RBQUE := #RBQUET
      LET R1 := RBQUE
      LET SCRCH := #TEMBJF
      REPEAT
          LET (R1)+ := SCRCH
      50007$:
      MOV #EF.RESTART,R0
      EMT C$REFG
      BCS 50004$
      INC LOGDEV
      CMP LOGDEV,L$UNIT
      BNE 50005$
      CLR LOGDEV
      MOV #EF.NEW,R0
      EMT C$REFG
      BCC 50006$
      CLR LOGDEV
      MOV #RQUET,RQUE
      MOV RQUE,RQUEUE
      MOV #RBQUET,RBQUE
      MOV RBQUE,R1
      MOV #TEMBUF,SCRCH
      MOV SCRCH,(R1)+
  
```

```

1435 012070          LET SCRTCH := SCRTCH + #270.
1436 012070 062767 000416 170014          ADD      #270.,SCRTCH
1437 012076          UNTIL R1 GT #RBQUEB
1438 012076 020127 015716          CMP      R1,#RBQUEB
1439 012102 003770          BLE      50007$
1440
1441          ;GET THE PTABLE ENTRY FOR THIS LOGICAL DEVICE
1442          ;
1443          REPEAT
1444          50010$:
1445          LET R1 := R1 + #1
1446          005201          INC      R1
1447          012106          GPHARD LOGDEV,SCRTCH
1448          016700 170024          MOV      LOGDEV,R0
1449          012112 104042          EMT
1450          012114 010067 167772          MOV      R0,SCRTCH
1451
1452          ;IF UNIT HAS BEEN DROPPED, UPDATE POINTERS ETC. AND TRY AGAIN
1453          ;
1454          IF.NO.ERROR THEN
1455          012120 103411          BCS      50011$
1456          012122          LET LOGDEV := LOGDEV + #1
1457          012122 005267 170010          INC      LOGDEV
1458          012126          IF LOGDEV EQ L$UNIT THEN
1459          012126 026767 170004 167656          CMP      LOGDEV,L$UNIT
1460          012134 001002          BNE      50012$
1461          012136          LET LOGDEV := #0
1462          012136 005067 167774          CLR      LOGDEV
1463          012142          ENDIF
1464          50012$:
1465          012142          LET R1 := #0
1466          012142 005001          CLR      R1
1467          012144          ENDIF
1468          50011$:
1469          012144          UNTIL R1 NE #0
1470          012144 005701          TST      R1
1471          012146 001756          BEQ      50010$
1472
1473          ;NOW GET THE PTABLE DATA FROM THE SUPERVISOR AND STORE WITHIN THE
1474          ;DIAGNOSTIC PROGRAM
1475          ;
1476          012150          LET R1 := #PTABLE
1477          012150 012701 002200          MOV      #PTABLE,R1
1478          012154          REPEAT
1479          50013$:
1480          012154          LET (R1) := @SCRTCH
1481          012154 017711 167732          MOV      @SCRTCH,(R1)
1482          012160          LET R1 := R1 + #2
1483          012160 062701 000002          ADD      #2,R1
1484          012164          LET SCRTCH := SCRTCH + #2
1485          012164 062767 000002 167720          ADD      #2,SCRTCH
1486          012172          UNTIL R1 EQ #PTEND
1487          012172 020127 002212          CMP      R1,#PTEND
1488          012176 001366          BNE      50013$
1489
1490          ;GET THE LINE NUMBER AND ISOLATE IT FOR FUTURE USE

```

```

1491
1492 012200          :      LET R1 := #PTABLE + #6
1493 012200 012701 002200          MOV      #PTABLE,R1
1494 012204 062701 000006          ADD      #6,R1
1495 012210          :      LET R1 := (R1)
1496 012210 011101          MOV      (R1),R1
1497 012212          :      LET R1 := R1 CLR.BY #177770
1498 012212 042701 177770          BIC      #177770,R1
1499 012216          :      LET R1 := R1 SHIFT #1
1500 012216 006301          ASL      R1
1501 012220          :      LET $LINE := LINMSK(R1)
1502 012220 016167 002214 003406  MOV      LINMSK(R1),$LINE
1503
1504          :RESTORE R1 AND EXIT INIT CODE
1505          :
1506 012226          :      POP R1
1507 012226 012601          MOV      (SP)+,R1
1508 012230          :      ENDINIT
1509 012230          :
1510 012230 104011          :
1511 012232 047045 040445 052126  VMSG:  .ASCIZ /%N%AVT62 ACCEPTANCE TEST  --  VERSION 10 TERMINALS%N/
1512 012240 031066 040440 041503
1513 012246 050105 040524 041516
1514 012254 020105 042524 052123
1515 012262 020040 026440 020055
1516 012270 020040 042526 051522
1517 012276 047511 020116 030061
1518 012304 052040 051105 044515
1519 012312 040516 051514 047045
1520 012320          :      G00
1521          :      012322
1522          :      .EVEN
1523          :*****
1524          :      CLEAN-UP CODING
1525          :*****
1526 012322          :      BGNCLN
1527          :
1528          :MAKE SURE DEVICE IS SHUT DOWN
1529          :
1530 012322          :      PUSH R1
1531 012322 010146          MOV      R1,-(SP)
1532 012324          :      LET R1 := PTABLE
1533 012324 016701 167650          MOV      PTABLE,R1
1534 012330          :      LET XMTCSR(R1) := #0
1535 012330 005061 000004          CLR      XMTCSR(R1)
1536 012334          :      LET RCVCSR(R1) := RCVCSR(R1) SET.BY #MASCLR
1537 012334 052761 000020 000000  BIS      #MASCLR,RCVCSR(R)
1538 012342          :      POP R1
1539 012342 012601          MOV      (SP)+,R1
1540          :
1541          :UPDATE PASS COUNT
1542          :
1543 012344          :      LET PASCNT := PASCNT + #1
1544 012344 005267 167562          INC      PASCNT
1545          :
1546          :EXIT CLEAN-UP CODE

```

1547
1548 012350
1549 012350
1550 012350 104012
1551 012352

:
ENDCLN
ENDMOD

L10026: EMT CSCLEAN

```

1552 012352          BGNMOD GSUB
1553                *****
1554                GLOBAL SUBROUTINES
1555                *****
1556                +-----+
1557                |ROUTINE TO PROCESS TERMINAL MESSAGES FOR ALL TESTS |
1558                |RETURNS POINTER TO MESSAGE ('OUTMSG') AND ITS SIZE  |
1559                |IN BYTES ('OUTSIZ')                                  |
1560                |                                                    |
1561                |SIZE MAY BE ANY NUMBER OF BYTES                      |
1562                |                                                    |
1563                |SETS A FLAG ('EXPINP') IF RESPONSE OTHER THAN 'ACK' IS |
1564                |TO BE EXPECTED                                       |
1565                |                                                    |
1566                |SETS ANOTHER FLAG ('EOMSG') IF THIS IS THE LAST MESSAGE|
1567                |                                                    |
1568                +-----+
1569 012352          ROUTINE MSG ;[3.2.14]
1570 012352
1571
1572 012352          DECLARE LOCALS
1573 012352          LOCAL <A>
1574 012352          ALLOCATE
1575 012352 010446
1576 012354 010604
1577 012356 162706 000002
1578
1579                :GET MESSAGE POINTER
1580
1581 012362          LET A(R4) := @MSGAP
1582 012362 017764 167574 177776
1583
1584                :IF MESSAGE POINTER IS ODD,
1585                :SET END_OF_MESSAGE INDICATOR
1586
1587 012370          LET SCRTCH := A(R4) AND #1
1588 012370 016467 177776 167514
1589 012376 016746 167510
1590 012402 042716 000001
1591 012406 042667 167500
1592 012412
1593 012412 005767 167474
1594 012416 001405
1595 012420          LET EOMSG := EOMSG + #1
1596 012420 005267 167510
1597 012424          LET A(R4) := A(R4) CLR.BY #1
1598 012424 042764 000001 177776
1599 012432          ENDIF
1600 012432
1601
1602                :IF MESSAGE POINTER IS NEGATIVE,
1603                :SET EXPECT_INPUT INDICATOR
1604
1605 012432          IF A(R4) LT #0 THEN
1606 012432 005764 177776
1607 012436 002006
  
```

```

MSG:
MOV R4,-(SP)
MOV SP,R4
SUB #2,SP

MOV @MSGAP,A(R4)

MOV A(R4),SCRTCH
MOV SCRTCH,-(SP)
BIC #1,(SP)
BIC (SP)+,SCRTCH

TST SCRTCH
BEQ 50002$

INC EOMSG

BIC #1,A(R4)

50002$:

TST A(R4)
BGE 50003$
  
```

```

1608 012440          LET EXPINP := EXPINP + #1
1609 012440 005267 167474          INC      EXPINP
1610 012444          LET A(R4) := A(R4) CLR.BY #100000
1611 012444 042764 100000 177776    BIC      #100000,A(R4)
1612
1613      : OTHERWISE, CLEAR EXPECT_INPUT
1614      :
1615 012452          ELSE
1616 012452 000402          BR      50003$
1617 012454          LET EXPINP := #0
1618 012454          LET EXPINP := #0
1619 012454 005067 167460          CLR      EXPINP
1620 012460          ENDIF
1621 012460          50004$:
1622
1623      : SET MESSAGE ADDRESS AND
1624      : MESSAGE_SIZE
1625      :
1626 012460          LET OUTMSG := A(R4)
1627 012460 016467 177776 167454    MOV      A(R4),OUTMSG
1628 012466          LET MSGAP := MSGAP + #2
1629 012466 062767 000002 167466    ADD      #2,MSGAP
1630 012474          LET OUTMSG+2 : @MSGAP
1631 012474 017767 167462 167442    MOV      @MSGAP,OUTMSG+2
1632
1633      : UPDATE MESSAGE_AREA_POINTER
1634      :
1635 012502          LET MSGAP := MSGAP + @MSGAP
1636 012502 067767 167454 167452    ADD      @MSGAP,MSGAP
1637 012510          LET MSGAP := MSGAP + #2
1638 012510 062767 000002 167444    ADD      #2,MSGAP
1639
1640      : IF MESSAGE AREA POINTER IS ODD
1641      : ADD 1 TO MAKE IT EVEN
1642      :
1643 012516          LET SCRTCH := MSGAP AND #1
1644 012516 016767 167440 167366    MOV      MSGAP,SCRTCH
1645 012524 016746 167362          MOV      SCRTCH,-(SP)
1646 012530 042716 000001          BIC      #1,(SP)
1647 012534 042667 167352          BIC      (SP)+,SCRTCH
1648 012540          IF SCRTCH NE #0 THEN
1649 012540 005767 167346          TST      SCRTCH
1650 012544 001402          BEQ      50005$
1651 012546          LET MSGAP := MSGAP + #1
1652 012546 005267 167410          INC      MSGAP
1653 012552          ENDIF
1654 012552          50005$:
1655 012552          ENDRTN
1656 012552          50000$:
1657 012552          50001$:
1658 012552 010406          MOV      R4,SP
1659 012554 012604          MOV      (SP)+,R4
1660 012556 000207          RTS      PC
1661
1662
1663

```

```

1664
1665
1666
1667
1668
1669
1670
1671 012560
1672 012560
1673 012560
1674 012560 005767 003064
1675 012564 001011
1676 012566
1677 012566 004767 000222
1678 012572
1679 012572 103002
1680 012574
1681 012574 000261
1682 012576 000504
1683 012600
1684 012600
1685 012600
1686 012600 005067 167370
1687 012604
1688 012604 005067 167366
1689 012610
1690 012610
1691
1692
1693
1694 012610
1695 012610 016767 167326 167346
1696 012616
1697 012616 012767 002142 167266
1698 012624 062767 000002 167260
1699 012632
1700 012632 017767 167254 167326
1701
1702 012640
1703 012640 012767 010434 167322
1704 012646
1705 012646 012767 000412 167316
1706
1707
1708
1709 012654
1710 012654 162705 000004
1711 012660 004767 000336
1712 012664 012567 167262
1713 012670 012567 167260
1714
1715
1716
1717
1718
1719 012674

```

```

*****
: ROUTINE TO HANDLE OUTPUT OF MESSAGES TO THE TERMINAL *
: UNDER TEST. MESSAGES MAY BE OF ANY LENGTH SINCE *
: ROUTINES USED BY THIS ROUTINE SPLIT THE MESSAGES INTO *
: BUFFERS OF APPROPRIATE LENGTH FOR DDCMP. *
*****
ROUTINE TEROUT ;[3.2.20]
IF INIT EQ #0 THEN
    CALL XSTRT
    IF .ERROR THEN
        RETURN ERROR
    ENDIF
    LET MSGNO := #0
    LET RMSGNO := #0
ENDIF
    TST INIT
    BNE 50002$
    JSR PC,XSTRT
    BCC 50003$
    SEC
    BR 50001$
50003$:
    CLR MSGNO
    CLR RMSGNO
50002$:
: INITIALIZE THE BUFFER ROUTINES [3.2.20.4]
:
LET INITMA := OUTMSG
MOV OUTMSG,INITMA
LET SCRTCH := #OUTMSG + #2
MOV #OUTMSG,SCRTCH
ADD #2,SCRTCH
LET INITMS := @SCRTCH
MOV @SCRTCH,INITMS
:
BUFFER #1,BAPNT
LET BAPNT := #TMPBUF
MOV #TMPBUF,BAPNT
LET BASIZE := #266.
MOV #266.,BASIZE
: GET A BUFFER OF THE MESSAGE
:
CALL GETOB OUT <OUTBUF,OUTSIZ>
SUB #2*2,R5
JSR PC,GETOB
MOV (R5)+,OUTBUF
MOV (R5)+,OUTSIZ
: PERFORM THE FOLLOWING LOOP WHILE
: A AND ABORT ARE CLEAR AND OUTSIZ
: IS GREATER THAN 0
:
WHILE OUTSIZ GT #0 AND ABORT EQ #0 DO

```

```

1720 012674
1721 012674 005767 167254
1722 012700 003442
1723 012702 005767 167210
1724 012706 001037
1725
1726
1727
1728 012710
1729 012710 162705 000004
1730 012714 010546
1731 012716 016745 167232
1732 012722 016745 167224
1733 012726 004767 000416
1734 012732 012605
1735 012734 012567 002714
1736 012740 012567 002712
1737 012744
1738 012744 004767 001674
1739
1740
1741
1742 012750
1743 012750 103005
1744 012752
1745 012752 104423
1746 012754 000024
1747 012756 002712
1748 012760
1749 012760 000261
1750 012762 000412
1751 012764
1752 012764
1753
1754
1755
1756 012764
1757 012764 162705 000004
1758 012770 004767 000226
1759 012774 012567 167152
1760 013000 012567 167150
1761
1762
1763
1764 013004
1765 013004 000733
1766 013006
1767
1768
1769
1770 013006
1771 013006
1772 013006 000241
1773 013010
1774 013010 000207
1775 013012 000000

```

```

:ADD PROTOCOL ENVELOPE AND TRANSMIT

```

```

CALL PROBUF IN <OUTBUF,OUTSIZ> OUT <XBUFFER,XSIZE>

```

```

:IF TRANSMIT ERROR, REPORT IT AND SET ERRXMT

```

```

IF.ERROR THEN

```

```

ERRHRD 20,MSG20 ;[3.2.20.10]

```

```

RETURN ERROR

```

```

ENDIF

```

```

:GET THE NEXT BUFFER

```

```

CALL GETOB OUT <OUTBUF,OUTSIZ>

```

```

:END OF LOOP

```

```

ENDDO

```

```

:IF ERROR OCCURRED, DO AN ERROR RETURN

```

```

ENDRTN

```

```

TYPE: 0

```

```

;TYPE OF MESSAGE EXPECTED

```

50004\$:

```

TST OUTSIZ
BLE 50005$
TST ABORT
BNE 50005$

```

```

SUB #2*2,R5
MOV R5,-(SP)
MOV OUTSIZ,-(R5)
MOV OUTBUF,-(R5)
JSR PC,PROBUF
MOV (SP)+,R5
MOV (R5)+,XBUFFER
MOV (R5)+,XSIZE

```

```

JSR PC,XMIT

```

```

BCC 50006$

```

```

TRAP T$ERCODE
.WORD 20
.WORD MSG20

```

```

SEC BR 50001$

```

50006\$:

```

SUB #2*2,R5
JSR PC,GETOB
MOV (R5)+,OUTBUF
MOV (R5)+,OUTSIZ

```

50005\$:

```

BR 50004$

```

50000\$:

```

CLC

```

50001\$:

```

RYS PC

```



```

1776
1777
1778
1779
1780
1781
1782
1783
1784
1785 013014
1786 013014
1787 013014
1788 013014
1789 013014
1790 013014 010446
1791 013016 010604
1792 013020 162706 000002
1793
1794
1795
1796
1797
1798 013024
1799 013024 012767 000010 002624
1800 013032
1801 013032 012767 003670 002614
1802
1803
1804
1805 013040
1806 013040 012764 002200 177776
1807 013046 062764 000005 177776
1808 013054
1809 013054 117467 177776 170613
1810
1811
1812
1813
1814 013062
1815 013062 162705 000002
1816 013066 010546
1817 013070 012745 000006
1818 013074 016745 002554
1819 013100 004767 004006
1820 013104 012605
1821 013106 012567 170564
1822
1823
1824
1825 013112
1826 013112 004767 001526
1827
1828
1829
1830 013116
1831 013116 103005

```

```

:*****
:ROUTINE TO SEND A 'STRT' MESSAGE TO THE TERMINAL
:
:A 'STACK' MUST BE RETURNED BY THE TERMINAL
:*****
:ROUTINE XSTRT                                ;[3.2.20.1]
:
:DECLARE LOCALS
:LOCAL <A>
:ALLOCATE
:
:THE DDCMP STRT MESSAGE IS 10 BYTES LONG (INCLUDING CRC) AND IS IN
:A BUFFER STARTING AT 'STRTB' -- PUT THIS INFORMATION
:IN THE TRANSMIT ROUTINE'S POINTERS
:
:LET XSIZE := #10
:
:LET XBUFFER := #STRTB
:
:GET THE TERMINAL ADDRESS AND PUT INTO THE STRT MESSAGE
:
:LET A(R4) := #PTABLE + #5
:
:LET STRTSA :B= @A(R4)
:
:COMPUTE THE CRC FOR THE STRT MESSAGE AND PUT IN THE LAST TWO BYTES
:OF THE BUFFER ('STRTC')
:
:CALL CRC IN <XBUFFER,#6> OUT <STRTC>
:
:TRANSMIT THE MESSAGE
:
:CALL XMIT
:
:IF AN ERROR OCCURS, REPORT IT AND EXIT WITH THE ERROR FLAG SET.
:
:IF .ERROR THEN

```

```

XSTRT:
MOV R4,-(SP)
MOV SP,R4
SUB #2,SP
MOV #10,XSIZE
MOV #STRTB,XBUFFER
MOV #PTABLE,A(R4)
ADD #5,A(R4)
MOVB @A(R4),STRTSA
SUB #1*2,R5
MOV R5,-(SP)
MOV #6,-(R5)
MOV XBUFFER,-(R5)
JSR PC,CRC
MOV (SP)+,R5
MOV (R5)+,STRTC
JSR PC,XMIT
BCC 50002$

```

```

1832 013120          ERRHRD 20,MSG20          ;[3.2.20.10]
1833 013120 104423
1834 013122 000024          TRAP          T$ERCODE
1835 013124 002712          .WORD          20
1836 013126          RETURN ERROR          .WORD          MSG20
1837 013126 000261          SEC
1838 013130 000430          BR          50001$
1839 013132          ENDIF
1840 013132          50002$:
1841
1842 :OTHERWISE, GET THE TERMINAL'S RESPONSE
1843 :
1844 013132          CALL RCVE
1845 013132 004767 003472          JSR          PC,RCVE
1846
1847 :IF RECEPTION ERROR, REPORT IT AND EXIT WITH THE ERROR FLAG SET
1848 :
1849 013136          IF.ERROR THEN
1850 013136 103005          ERRHRD 21,MSG21          ;[3.2.20.10]
1851 013140          TRAP          T$ERCODE
1852 013140 104423          .WORD          21
1853 013142 000025          .WORD          MSG21
1854 013144 002761          RETURN ERROR
1855 013146          SEC
1856 013146 000261          BR          50001$
1857 013150 000420          ENDIF
1858 013152          50003$:
1859 013152
1860 :OTHERWISE, SET TYPE TO 'STACK' AND CHECK TO SEE THAT IT WAS THE REPLY
1861 :
1862 :
1863 013152          LET TYPE := #STACK
1864 013152 012767 000007 177632          MOV          #STACK,TYPE
1865 013160          CALL CHKREP IN <TYPE>
1866 013160 010546          MOV          R5,-(SP)
1867 013162 016745 177624          MOV          TYPE,-(R5)
1868 013166 004767 001062          JSR          PC,CHKREP
1869 013172 012605          MOV          (SP)+,R5
1870
1871 :IF REPLY WASN'T A 'STACK', REPORT THE ERROR AND EXIT WITH THE ERROR FLAG SET
1872 :
1873 013174          IF.ERROR THEN
1874 013174 103005          ERRHRD 25,MSG25          ;[3.2.20.10]
1875 013176          TRAP          T$ERCODE
1876 013176 104423          .WORD          25
1877 013200 000031          .WORD          MSG25
1878 013202 003153          RETURN ERROR
1879 013204          SEC
1880 013204 000261          BR          50001$
1881 013206 000401          ENDIF
1882 013210          50004$:
1883 013210
1884 :END OF XSTRT ROUTINE
1885 :
1886 :
1887 013210          ENDRTN
  
```

```

1888 013210
1889 013210 000241
1890 013212
1891 013212 010406
1892 013214 012604
1893 013216 000207
1894 013220 000000          DDSTRY: 0
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906 013222
1907 013222
1908 013222
1909 013222
1910 013222
1911 013222 010446
1912 013224 010604
1913 013226 162706 000002
1914
1915
1916
1917 013232
1918 013232 016765 166732 000000
1919 013240 062765 000010 000000
1920
1921
1922
1923 013246
1924 013246 005065 000002
1925
1926
1927
1928 013252
1929 013252 016767 166714 166632
1930 013260 162767 000012 166624
1931
1932
1933
1934 013266
1935 013266 016564 000000 177776
1936
1937
1938
1939 013274
1940 013274
1941 013274 026567 000002 166610
1942 013302 002017
1943 013304 005767 166656

```

```

50000$:
50001$:
50002$:
50003$:

```

```

          CLC
          MOV     R4,SP
          MOV     (SP)+,R4
          RTS     PC

          *****
          THIS ROUTINE SPLITS A MESSAGE OF ANY LENGTH INTO
          SEGMENTS OF A SIZE SUITABLE FOR TRANSMISSION OVER A DATA
          LINK -- RETURNS BUFFER ADDRESS, BUFFER SIZE AND END_OF_MESSAGE+
          (BUFFER SIZE = 0 IS END_OF_MESSAGE)
          *****
          ROUTINE GETOB <A,B> ;[3.2.20.5]
          GETOB:
          DECLARE LOCALS
          LOCAL <C>
          ALLOCATE
          MOV     R4,-(SP)
          MOV     SP,R4
          SUB     #2,SP
          ;LEAVE ENOUGH ROOM FOR A DDCMP HEADER
          LET A(R5) := BAPNT + #10
          MOV     BAPNT,A(R5)
          ADD     #10,A(R5)
          ;INIT SIZE TO ZERO
          LET B(R5) := #0
          CLR     B(R5)
          ;SCRTCH HAS MAXIMUM DDCMP MESSAGE SIZE
          LET SCRTCH := BASIZE - #12
          MOV     BASIZE,SCRTCH
          SUB     #12,SCRTCH
          ;GET A COPY OF THE BUFFER POINTER
          LET C(R4) := A(R5)
          MOV     A(R5),C(R4)
          ;
          REPEAT LOOP UNTIL MAX SIZE OR END OF DATA
          WHILE B(R5) LT SCRTCH AND INITMS GT #0 DO
          CMP     B(R5),SCRTCH
          BGE     50003$
          TST     INITMS

```

```

1944 013310 003414 BLE 50003$
1945
1946 :MOVE DATA INTO MESSAGE BUFFER
1947 :
1948 013312 LET @C(R4) :B= @INITMA ;[3.2.20.5.1 + 2]
1949 013312 117774 166646 177776 MOV @INITMA,@C(R4)
1950 013320 LET INITMS := INITMS - #1 ;[3.2.20.5.1]
1951 013320 005367 166642 DEC INITMS
1952 013324 LET INITMA := INITMA + #1
1953 013324 005267 166634 INC INITMA
1954 013330 LET B(R5) := B(R5) + #1 ;[3.2.20.5.2]
1955 013330 005265 000002 INC B(R5)
1956 013334 LET C(R4) := C(R4) + #1
1957 013334 005264 177776 INC C(R4)
1958 013340 ENDDO
1959 013340 000755 BR 50002$
1960 013342 50003$:
1961 :
1962 : END OF LOOP
1963 :
1964 013342 ENDRTN
1965 013342 50000$:
1966 013342 50001$:
1967 013342 010406 MOV R4,SP
1968 013344 012604 MOV (SP)+,R4
1969 013346 000207 RTS PC
1970
1971
1972
1973
1974
1975 :*****
1976 : ROUTINE TO PROVIDE LINE PROTOCOL HEADER AND CHECKSUMS +
1977 : CALL WITH DATA AREA ADDRESS AND SIZE (IN BYTES) +
1978 : RETURN WITH MODIFIED ADDRESS AND SIZE +
1979 :*****
1980 013350 ROUTINE PROBUF <A,B,C,D> ;[3.2.20.6]
1981 013350 PROBUF:
1982 013350 PUSH R1
1983 013350 010146 MOV R1,-(SP)
1984 013352 PUSH R2
1985 013352 010246 MOV R2,-(SP)
1986
1987 :CHANGE POINTER TO INCLUDE DDCMP HEADER
1988 :
1989 013354 LET R2 := A(R5) - #10 ;[3.2.20.6.1]
1990 013354 016502 000000 MOV A(R5),R2
1991 013360 162702 000010 SUB #10,R2
1992
1993 :FIRST BYTE IS SOH (START OF HEADER)
1994 :
1995 013364 LET (R2)+ :B- #SOH
1996 013364 112722 000201 MOVB #SOH,(R2)+
1997
1998 :NEXT TWO BYTES GLT MESSAGE SIZE
1999 :

```

```

2000 013370          LET R1 := B(R5)
2001 013370 016501 000002          MOV      B(R5),R1
2002 013374          LET (R2)+ :B= R1
2003 013374 110122          MOV      R1,(R2)+
2004 013376          LET R1 := SWAP R1
2005 013376 000301          SWAB    R1
2006 013400          LET (R2) :B= R1
2007 013400 110112          MOV      R1,(R2)
2008
2009          ;SET SELECT BIT
2010
2011 013402          LET (R2) :B= (R2) SET.BY #200
2012 013402 152712 000200          BISB    #200,(R2)
2013 013406          LET R2 := R2 + #1
2014 013406 005202          INC     R2
2015
2016          ;NEXT BYTE IS LAST GOOD RESPONSE
2017
2018 013410          LET (R2)+ :B= RMSGNO
2019 013410 116722 166562          MOV      RMSGNO,(R2)+
2020
2021          ;NEXT BYTE GETS MESSAGE NUMBER --- POP NUMBER FOR NEXT MESSAGE (IF ANY)
2022
2023 013414          LET MSGNO := MSGNO + #1
2024 013414 005267 166554          INC     MSGNO
2025 013420          LET (R2)+ :B= MSGNO
2026 013420 116722 166550          MOV      MSGNO,(R2)+
2027
2028          ;GET THE TERMINAL ADDRESS FOR THE NEXT HEADER BYTE
2029
2030 013424          LET R1 :- #PTABLE + #5
2031 013424 012701 002200          MOV      #PTABLE,R1
2032 013430 062701 000005          ADD     #5,R1
2033 013434          LET (R2)+ :B= (R1)
2034 013434 111122          MOV      (R1),(R2)+
2035
2036          ;GET HEADER START ADDRESS AND SIZE IN BYTES
2037
2038 013436          LET SCRTCH := A(R5) - #10
2039 013436 016567 000000 166446          MOV      A(R5),SCRTCH
2040 013444 162767 000010 166440          SUB     #10,SCRTCH
2041 013452          LET TEMP := #6
2042 013452 012767 000006 166434          MOV      #6,TEMP
2043
2044          ;COMPUTE HEADER CRC
2045
2046 013460          CALL CRC IN <SCRTCH,TEMP> OUT <TEMP>          ;[3.2.20.6.2]
2047 013460 162705 000002          SUB     #1*2,R5
2048 013464 010546          MOV      R5,-(SP)
2049 013466 016745 166422          MOV      TEMP,-(R5)
2050 013472 016745 166414          MOV      SCRTCH,-(R5)
2051 013476 004767 003410          JSR     PC,CRC
2052 013502 012605          MOV      (SP)+,R5
2053 013504 012567 166404          MOV      (R5)+,TEMP
2054
2055          ;PUT CRC INTO BUFFER

```

```

2056
2057 013510          ;      LET (R2) := TEMP                ;[3.2.20.6.3]
2058 013510 016712 166400          ;      MOV      TEMP,(R2)
2059
2060          ;GET ADDRESS OF MESSAGE AND MESSAGE SIZE IN BYTES
2061
2062 013514          ;      LET SCRTCH := A(R5)
2063 013514 016567 000000 166370          ;      MOV      A(R5),SCRTCH
2064 013522          ;      LET TEMP := B(R5)
2065 013522 016567 000002 166364          ;      MOV      B(R5),TEMP
2066
2067          ;COMPUTE DATA CRC
2068
2069 013530          ;      CALL CRC IN <SCRTCH,TEMP> OUT <TEMP>          ;[3.2.20.6.4]
2070 013530 162705 000002          ;      SUB      #1*2,R5
2071 013534 010546          ;      MOV      R5,-(SP)
2072 013536 016745 166352          ;      MOV      TEMP,-(R5)
2073 013542 016745 166344          ;      MOV      SCRTCH,-(R5)
2074 013546 004767 003340          ;      JSR      PC,CRC
2075 013552 012605          ;      MOV      (SP)+,R5
2076 013554 012567 166334          ;      MOV      (R5)+,TEMP
2077
2078          ;PUT DATA CRC IN LAST TWO BYTES OF BUFFER
2079
2080 013560          ;      LET R2 := A(R5) + B(R5)                ;[3.2.20.6.5]
2081 013560 016502 000000          ;      MOV      A(R5),R2
2082 013564 066502 000002          ;      ADD      B(R5),R2
2083 013570          ;      LET (R2)+ :=B= TEMP
2084 013570 116722 166320          ;      MOV      TEMP,(R2)+
2085 013574          ;      LET TEMP := SWAP TEMP
2086 013574 000367 166314          ;      SWAB     TEMP
2087 013600          ;      LET (R2) :=B= TEMP
2088 013600 116712 166310          ;      MOV      TEMP,(R2)
2089
2090          ;RETURN WITH MODIFIED ADDRESS AND SIZE
2091
2092 013604          ;      LET D(R5) := B(R5) + #12
2093 013604 016565 000002 000006          ;      MOV      B(R5),D(R5)
2094 013612 062765 000012 000006          ;      ADD      #12,D(R5)
2095 013620          ;      LET C(R5) := A(R5) - #10
2096 013620 016565 000000 000004          ;      MOV      A(R5),C(R5)
2097 013626 162765 000010 000004          ;      SUB      #10,C(R5)
2098 013634          ;      POP R2
2099 013634 012602          ;      MOV      (SP)+,R2
2100 013636          ;      POP R1
2101 013636 012601          ;      MOV      (SP)+,R1
2102 013640          ;      ENDRTN
2103 013640
2104 013640          ;      50000$:
2105 013640 000207          ;      50001$:
2106          ;      RTS      PC
2107
2108
2109
2110
2111          ;*****

```

```

2112          :.....ROUTINE TO GET TERMINAL INPUT
2113          :*****+
2114 013642    :.....ROUTINE TERIN <A> ;[3.2.1]
2115 013642    :.....TERIN:
2116          :
2117          :SAVE R1
2118          :
2119 013642    :
2120 013642 010146 :
2121          :
2122          :GET TERMINAL RESPONSE
2123          :
2124 013644    :
2125 013644 004767 002760 :
2126          :
2127          :IF RECEIVE ERROR, REPORT IT AND EXIT WITH THE ERROR FLAG SET
2128          :
2129 013650    :
2130 013650 103007 :
2131 013652    :
2132 013652 104463 :
2133 013654 000050 :
2134 013656 002761 :
2135 013660 004262 :
2136          :
2137          :.....RESTORE R1 FIRST
2138          :
2139 013662    :
2140 013662 012601 :
2141 013664    :
2142 013664 000261 :
2143 013666 000571 :
2144 013670    :
2145 013670    :
2146          :
2147          :IF MESSAGE IS ONLY 10 BYTES, SEE IF IT IS AN ACK.
2148          :IT'S OK TO BE AN ACK, BUT RECEPTION IS INCOMPLETE (CLEAR FLAG).
2149          :
2150 013670    :
2151 013670 026727 166264 000010 :
2152 013676 001027 :
2153 013700    :
2154 013700 005065 000000 :
2155 013704    :
2156 013704 012767 000001 177100 :
2157 013712    :
2158 013712 010546 :
2159 013714 016745 177072 :
2160 013720 004767 000330 :
2161 013724 012605 :
2162 013726    :
2163 013726 103003 :
2164 013730    :
2165 013730 012601 :
2166 013732    :
2167 013732 000261 :
  
```

2168	013734	000546						
2169	013736			ENDIF			BR	50001\$
2170	013736					50004\$:		
2171	013736			LET R1 := INBUF + #4				
2172	013736	016701	166214				MOV	INBUF,R1
2173	013742	062701	000004				ADD	#4,R1
2174	013746			LET RMSGNO :B= (R1)			MOVB	(R1),RMSGNO
2175	013746	111167	166224				MOV	(SP)+,R1
2176	013752			POP R1				
2177	013752	012601		RETURN			BR	50000\$
2178	013754							
2179	013754	000535		ENDIF		50003\$:		
2180	013756							
2181	013756							
2182								
2183				:IF RECEIVE OK, CHECK THE HEADER CRC				
2184				:COMPUTE EXPECTED CRC				
2185				:				
2186	013756			CALL CRC IN <INBUF,#6> OUT <A04EX>				
2187	013756	162705	000002				SUB	#1*2,R5
2188	013762	010546					MOV	R5,-(SP)
2189	013764	012745	000006				MOV	#6,-(R5)
2190	013770	016745	166162				MOV	INBUF,-(R5)
2191	013774	004767	003112				JSR	PC,CRC
2192	014000	012605					MOV	(SP)+,R5
2193	014002	012567	170144				MOV	(R5)+,A04EX
2194								
2195				:GET THE CRC RECEIVED				
2196				:				
2197	014006			LET R1 := INBUF + #10				
2198	014006	016701	166144				MOV	INBUF,R1
2199	014012	062701	000010				ADD	#10,R1
2200	014016			LET A04RC :B= -(R1)				
2201	014016	114167	170126				MOVB	-(R1),A04RC
2202	014022			LET A04RC := SWAP A04RC				
2203	014022	000367	170122				SWAB	A04RC
2204	014026			LET A04RC :B= -(R1)				
2205	014026	114167	170116				MOVB	-(R1),A04RC
2206								
2207				:IF THEY AREN'T THE SAME, REPORT THE ERROR, RESTORE R1 AND EXIT WITH				
2208				:THE ERROR FLAG SET				
2209				:				
2210	014032			IF A04EX NE A04RC THEN				
2211	014032	026767	170114 170110				CMP	A04EX,A04RC
2212	014040	001407					BEQ	50005\$
2213	014042			ERRHRD 41,MSG41,AREA41				
2214	014042	104463					TRAP	T\$ERCODE
2215	014044	000051					.WORD	41
2216	014046	002503					.WORD	MSG41
2217	014050	004116					.WORD	AREA41
2218	014052			POP R1				
2219	014052	012601		RETURN ERROR			MOV	(SP)+,R1
2220	014054							
2221	014054	000261					SEC	
2222	014056	000475					BR	50001\$
2223	014060			ENDIF				


```

2224 014060                                     50005$:
2225
2226      : COMPARE TERMINAL ADDRESS RECEIVED WITH THAT EXPECTED
2227      : AND IF THEY AREN'T THE SAME, REPORT THE ERROR, RESTORE R1 AND
2228      : EXIT WITH THE ERROR FLAG SET
2229      :
2230      IFB -(R1) NE STRTSA THEN
2231 014060 124167 167611                       CMPB   -(R1), STRTSA
2232 014064 001407                               BEQ    50006$
2233 014066                                     ERRHRD 42,MSG42,AREA42
2234 014066 104463                               TRAP   T$ERCODE
2235 014070 000052                               .WORD  42
2236 014072 002437                               .WORD  MSG42
2237 014074 004064                               .WORD  AREA42
2238 014076                                     POP R1
2239 014076 012601                               MOV    (SP)+,R1
2240 014100                                     RETURN ERROR
2241 014100 000261
2242 014102 000463                               SEC
2243 014104                                     BR     50001$
2244 014104
2245                                     50006$:
2246      : MOVE THE POINTERS PAST THE HEADER AND COMPUTE THE DATA CRC
2247      :
2248 014104                                     LET INBUF := INBUF + #10
2249 014104 062767 000010 166044                 ADD    #10, INBUF
2250 014112                                     LET INSIZ := INSIZ - #12
2251 014112 162767 000012 166040                 SUB    #12, INSIZ
2252 014120                                     CALL CRC IN <INBUF,INSIZ> OUT <A04EX>
2253 014120 162705 000002
2254 014124 010546
2255 014126 016745 166026
2256 014132 016745 166020
2257 014136 004767 002750
2258 014142 012605
2259 014144 012567 170002
2260
2261      : GET THE DATA CRC RECEIVED
2262      :
2263 014150                                     LET R1 := INBUF + INSIZ
2264 014150 016701 166002
2265 014154 066701 166000
2266 014160                                     LET R1 : R1 + #2
2267 014160 062701 000002
2268 014164                                     LET A04RC :B= -(R1)
2269 014164 114167 167760
2270 014170                                     LET A04RC := SWAP A04RC
2271 014170 000367 167754
2272 014174                                     LET A04RC .B= -(R1)
2273 014174 114167 167750
2274
2275      : IF THEY AREN'T THE SAME; REPORT THE ERROR, RESTORE R1 AND
2276      : EXIT WITH THE ERROR FLAG SET
2277      :
2278 014200                                     IF A04RC NE A04EX THEN
2279 014200 026767 167744 167744                 CMP    A04RC,A04EX

```

```

2280 014206 001407
2281 014210 ERRHRD 43,MSG43,AREA43
2282 014210 104463
2283 014212 000053
2284 014214 003422
2285 014216 004116
2286 014220 POP R1
2287 014220 012601
2288 014222 RETURN ERROR
2289 014222 000261
2290 014224 000412
2291 014226 ENDIF
2292 014226
2293
2294
2295
2296 014226 OTHERWISE; SET COMPLETE FLAG, RESTORE R1 AND EXIT NORMALLY
2297 014226 005265 000000 LET A(R5) := A(R5) + #1
2298 014232 LET R1 := INBUF - #4
2299 014232 016701 165720
2300 014236 162701 000004
2301 014242 LET RMSGNO :B= (R1)
2302 014242 111167 165730
2303 014246 POP R1
2304 014246 012601
2305 014250 ENDRTN
2306 014250
2307 014250 000241
2308 014252
2309 014252 000207
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326 014254
2327 014254
2328 014254
2329 014254
2330 014254
2331 014254 010446
2332 014256 010604
2333 014260 162706 000002
2334
2335

```

OTHERWISE; SET COMPLETE FLAG, RESTORE R1 AND EXIT NORMALLY

```

*****
ROUTINE TO CHECK AN INPUT BUFFER FOR A STANDARD
RESPONSE ('ACK' OR 'STACK')
LOCATION 'TYPE' CONTAINS THE RESPONSE CODE EXPECTED
INPUT BUFFER POINTER IS IN 'INBUF'
INPUT BUFFER SIZE (IN BYTES) IS IN 'INSIZ'
*****
ROUTINE CHKREP <B> ;[3.2.20.3, 3.2.20.9]
CHKREP:

```

IF MESSAGE IS NOT 10 BYTES LONG, IT CAN'T BE A CONTROL MESSAGE

```

BEQ 50007$
TRAP T$ERCODE
.WORD 43
.WORD MSG43
.WORD AREA43
MOV (SP)+,R1
SEC
BR 50001$
50007$:
INC A(R5)
MOV INBUF,R1
SUB #4,R1
MOVB (R1),RMSGNO
MOV (SP)+,R1
50000$:
CLC
50001$:
RTS PC

```

```

MOV R4,-(SP)
MOV SP,R4
SUB #2,SP

```

```

2336 ;TYPE AN ERROR MESSAGE AND EXIT WITH ERROR SET
2337
2338 IF INSIZ NE #10 THEN
2339 014264 026727 165670 000010
2340 014272 001406
2341 014274 ERRHRD 01,MSG01,AREA01 ;[3.2.20.3.4, 3.2.20.9.4]
2342 014274 104463
2343 014276 000001
2344 014300 002323
2345 014302 003760
2346 014304 RETURN ERROR
2347 014304 000261
2348 014306 000553
2349 014310
2350 014310 ELSE
2351 50002$:
2352 ;OTHERWISE, COMPUTE THE CRC AND CHECK IT AGAINST THAT SENT
2353
2354 014310 CALL CRC IN <INBUF,#6> OUT <TEMP> ;[3.2.20.3.1, 3.2.20.9.1]
2355 014310 162705 000002
2356 014314 010546
2357 014316 012745 000006
2358 014322 016745 165630
2359 014326 004767 002560
2360 014332 012605
2361 014334 012567 165554
2362 014340
2363 014340 016764 165612 177776
2364 014346 062764 000007 177776
2365 014354
2366 014354 117467 177776 165530
2367 014362
2368 014362 000367 165524
2369 014366
2370 014366 005364 177776
2371 014372
2372 014372 117467 177776 165512
2373
2374 ;IF CRC'S MATCH GO ON... OTHERWISE SKIP FOLLOWING CODE
2375
2376 014400 IF TEMP EQ SCRTCH THEN ;[3.2.20.3.2, 3.2.20.9.2]
2377 014400 026767 165510 165504
2378 014406 001076
2379
2380 ;COMPARE CONTROL MESSAGE TYPE WITH THAT PASSED AS AN ARGUMENT
2381
2382 014410 LET A(R4) := INBUF + #1
2383 014410 016764 165542 177776
2384 014416 005264 177776
2385 014422
2386 014422 117464 177776 177776
2387 014430
2388 014430 042764 177400 177776
2389
2390 ;IF NOT THE EXPECTED TYPE, PRINT ERROR MESSAGE AND EXIT WITH ERROR SET
2391

```

```

CMP INSIZ,#10
BEQ 50002$
TRAP T$ERCODE
.WORD 1
.WORD MSG01
.WORD AREA01

SEC
BR 50001$

SUB #1*2,R5
MOV R5,-(SP)
MOV #6,-(R5)
MOV INBUF,-(R5)
JSR PC,CRC
MOV (SP)+,R5
MOV (R5)+,TEMP

MOV INBUF,A(R4)
ADD #7,A(R4)

MOVB @A(R4),SCRTCH

SWAB SCRTCH

DEC A(R4)

MOVB @A(R4),SCRTCH

CMP TEMP,SCRTCH
BNE 50004$

MOV INBUF,A(R4)
INC A(R4)

MOVB @A(R4),A(R4)

BIC #HYBYTE,A(R4)

```

```

2392 014436                IF A(R4) NE B(R5) THEN                ;[3.2.20.3.3, 3.2.20.9.3]
2393 014436 026465 177776 000000                CMP      A(R4),B(R5)
2394 014444 001414                BEQ      50005$
2395 014446                LET A02A := A(R4)
2396 014446 016467 177776 167340                MOV      A(R4),A02A
2397 014454                LET A02B := B(R5)
2398 014454 016567 000000 167334                MOV      B(R5),A02B
2399 014462                ERRHRD 02,MSG02,AREA02                ;[3.2.20.3.4, 3.2.20.9.4]
2400 014462 104463                TRAP    T$ERCODE
2401 014464 000002                .WORD  2
2402 014466 002350                .WORD  MSG02
2403 014470 003762                .WORD  AREA02
2404 014472                RETURN ERROR
2405 014472 000261                SEC
2406 014474 000460                BR      50001$
2407
2408                ; OTHERWISE, CHECK TERMINAL ADDRESS
2409
2410                ELSE
2411 014476                50005$:
2412 014476                LET SCRTCH := #PTABLE + #5
2413 014476 012767 002200 165406                MOV      #PTABLE,SCRTCH
2414 014504 062767 000005 165400                ADD      #5,SCRTCH
2415 014512                LET SCRTCH :B= @SCRTCH
2416 014512 117767 165374 165372                MOV      @SCRTCH,SCRTCH
2417 014520                LET SCRTCH := SCRTCH CLR.BY #HYBYTE
2418 014520 042767 177400 165364                BIC      #HYBYTE,SCRTCH
2419 014526                LET TEMP := INBUF + #5
2420 014526 016767 165424 165360                MOV      INBUF,TEMP
2421 014534 062767 000005 165352                ADD      #5,TEMP
2422 014542                LET TEMP :B= @TEMP
2423 014542 117767 165346 165344                MOV      @TEMP,TEMP
2424 014550                LET TEMP := TEMP CLR.BY #HYBYTE
2425 014550 042767 177400 165336                BIC      #HYBYTE,TEMP
2426
2427                ; IF NOT THE EXPECTED TERMINAL ADDRESS, THEN REPORT THE ERROR AND
2428                ; EXIT WITH ERROR SET
2429
2430                IF TEMP NE SCRTCH THEN
2431 014556 026767 165332 165326                CMP      TEMP,SCRTCH
2432 014564 001406                BEQ      50007$
2433 014566                ERRHRD 03,MSG03,AREA03                ;[3.2.20.3.4, 3.2.20.9.4]
2434 014566 104463                TRAP    T$ERCODE
2435 014570 000003                .WORD  3
2436 014572 002437                .WORD  MSG03
2437 014574 004064                .WORD  AREA03
2438 014576                RETURN ERROR
2439 014576 000261                SEC
2440 014600 000416                BR      50001$
2441 014602                ENDIF
2442 014602                50007$:
2443 014602                ENDIF
2444 014602                50006$:
2445
2446                ; IF CRC WAS IN ERROR, SKIP ABOVE CODE, COME TO HERE, REPORT ERROR AND
2447                ; EXIT WITH ERROR SET

```

```

2448
2449 014602          ELSE
2450 014602 000414          50004$: BR      50010$
2451 014604
2452 014604          LET A04RC := SCRTCH
2453 014604 016767 165302 167336          MOV      SCRTCH,A04RC
2454 014612          LET A04EX := TEMP
2455 014612 016767 165276 167332          MOV      TEMP,A04EX
2456 014620          ERRHRD 04,MSG04,AREA04          ;[3.2.20.3.4, 3.2.20.9.4]
2457 014620 104463          TRAP     T$ERCODE
2458 014622 000004          .WORD    4
2459 014624 002503          .WORD    MSG04
2460 014626 004116          .WORD    AREA04
2461 014630          RETURN ERROR
2462 014630 000261          SEC
2463 014632 000401          BR      50001$
2464 014634          ENDIF
2465 014634          50010$:
2466 014634          ENDIF          50003$:
2467 014634
2468
2469 :NORMAL RETURN
2470 :
2471 014634          ENDRTN
2472 014634          50000$:
2473 014634 000241          CLC
2474 014636          50001$:
2475 014636 010406          MOV      R4,SP
2476 014640 012604          MOV      (SP)+,R4
2477 014642 000207          RTS      PC
2478
2479
2480
2481
2482
2483 :*****
2484 : THIS ROUTINE SETS UP AND CHECKS FOR COMPLETE TRANSMISSION +
2485 : RETURNS AN ERROR FLAG IF ANY ERROR OCCURS DURING TRANSMISSION +
2486 : SET 'ABORT' IF INITIALIZATION OF THE DEVICE FAILS +
2487 :*****
2488 014644          ROUTINE XMIT          ;[3.2.20.7 AND 3.2.20.1]
2489 014644          XMIT:
2490 014644          DECLARE LOCALS
2491 014644          LOCAL <B>
2492 014644          ALLOCATE
2493 014644 010446          MOV      R4,-(SP)
2494 014646 010604          MOV      SP,R4
2495 014650 162706 000002          SUB      #2,SP
2496
2497 :CLEAR ERROR COUNT, TRANSMIT DONE FLAG AND RECEIVE DONE FLAG
2498 :
2499 014654          LET B(R4) := #0
2500 014654 005064 177776          CLR      B(R4)
2501 014660          LET XDONE := #0
2502 014660 005067 000270          CLR      XDONE
2503 014664          LET RDONE : #0
  
```

```

2504 014664 005067 000266                                CLR      RDONE
2505
2506                :
2507                :                               START OF LOOP
2508 014670                REPEAT
2509 014670                                50002$:
2510
2511                :
2512                :                               TRY TO INIT DEVICE
2513 014670                :
2514 014670 004767 000264                CALL DEVINIT
2515
2516                :
2517                :                               IF ERROR OCCURRED....
2518 014674                :
2519 014674 103033                IF.ERROR THEN
2520 014676                :                               PUSH R1
2521 014676 010146                :                               PUSH R2
2522 014700                :
2523 014700 010246                :
2524                :
2525                :                               GET A COPY OF THE DEVICE REGISTERS
2526                :
2527 014702                :                               LET R1 := PTABLE
2528 014702 016701 165272                :                               MOV      PTABLE,R1
2529 014706                :                               LET R2 := #DZREG
2530 014706 012702 002254                :                               MOV      #DZREG,R2
2531 014712                :                               LET RCVCSR(R2) := RCVCSR(R1)
2532 014712 016162 000000 000000        :                               MOV      RCVCSR(R1),RCVCS
2533 014720                :                               LET RCVBUF(R2) := RCVBUF(R1)
2534 014720 016162 000002 000002        :                               MOV      RCVBUF(R1),RCVBU
2535 014726                :                               LET XMTCSR(R2) := XMTCSR(R1)
2536 014726 016162 000004 000004        :                               MOV      XMTCSR(R1),XMTCS
2537 014734                :                               LET XMTBUF(R2) := XMTBUF(R1)
2538 014734 016162 000006 000006        :                               MOV      XMTBUF(R1),XMTBU
2539 014742                :                               POP R2
2540 014742 012602                :                               MOV      (SP)+,R2
2541 014744                :                               POP R1
2542 014744 012601                :                               MOV      (SP)+,R1
2543
2544                :
2545                :                               REPORT THE ERROR AND UPDATE THE ERROR COUNT
2546 014746                :
2547 014746 104463                :                               ERRHRD 23,MSG23,AREA23
2548 014750 000027                :                               TRAP   T$ERCODE
2549 014752 003044                :                               .WORD  23
2550 014754 004262                :                               .WORD  MSG23
2551 014756                :                               .WORD  AREA23
2552 014756 005264 177776                :                               LET B(R4) :- B(R4) + #1
2553                :                               INC   B(R4)
2554                :
2555                :                               OTHERWISE, CLEAR THE ERROR COUNT
2556 014762                :
2557 014762 000402                :                               ELSE
2558 014764                :                               50003$:
2559 014764                :                               LET B(R4) := #0

```

```

2560 014764 005064 177776                                CLR      B(R4)
2561 014770                                ENDIF
2562 014770                                50004$:
2563                                     :
2564                                     : IF ERROR COUNT HAS EXCEEDED LIMITS, ABORT AND NEGATE THE COUNT
2565                                     :
2566 014770                                IF B(R4) EQ ILIMIT THEN
2567 014770 026467 177776 165126                    CMP      B(R4),ILIMIT
2568 014776 001010                                BNE     50005$
2569 015000                                ERRDF 00,MSG00,AREA00
2570 015000 104462                                TRAP    T$ERCODE
2571 015002 000000                                .WORD  0
2572 015004 002264                                .WORD  MSG00
2573 015006 003700                                .WORD  AREA00
2574 015010                                LET ABORT := ABORT + #1
2575 015010 005267 165102                                INC     ABORT
2576 015014                                LET B(R4) := NEGATE B(R4)
2577 015014 005464 177776                                NEG     B(R4)
2578 015020                                ENDIF
2579 015020                                50005$:
2580                                     :
2581                                     :           END OF LOOP
2582                                     : EXIT LOOP WHEN ERROR COUNT IS ZERO OR NEGATIVE (ABORT)
2583                                     :
2584 015020                                UNTIL B(R4) LE #0
2585 015020 005764 177776                                TST     B(R4)
2586 015024 003321                                BGT    50002$
2587                                     :
2588                                     : IF NOT ABORT THEN EXECUTE THE FOLLOWING CODE
2589                                     : OTHERWISE SKIP AND RETURN WITH ERROR SET
2590                                     :
2591 015026                                IF B(R4) EQ #0 THEN
2592 015026 005764 177776                                TST     B(R4)
2593 015032 001042                                BNE    50006$
2594                                     :
2595                                     : WAIT FOR TRANSMIT DONE FLAG TO SET
2596                                     :
2597 015034                                REPEAT
2598 015034                                BREAK
2599 015034                                50007$:
2600 015034 104022                                EMT    C$BRK
2601 015036                                UNTIL XDONE NE #0
2602 015036 005767 000112                                TST    XDONE
2603 015042 001774                                BEQ    50007$
2604                                     :
2605                                     : IF TRANSMIT FLAG IS NEGATIVE, AN ERROR OCCURRED.
2606                                     :
2607 015044                                IF XDONE LT #0 THEN
2608 015044 005767 000104                                TST    XDONE
2609 015050 002032                                BGE    50010$
2610 015052                                PUSH R1
2611 015052 010146                                MOV    R1,-(SP)
2612 015054                                PUSH R2
2613 015054 010246                                MOV    R2,-(SP)
2614                                     :
2615                                     : IF ERROR, GET COPY OF DEVICE REGISTERS, PRINT MESSAGE AND EXIT WITH ERROR SET

```

```
2616 ;OTHERWISE DO A NORMAL EXIT
2617 ;
2618 015056 LET R1 := PTABLE
2619 015056 016701 165116 MOV PTABLE,R1
2620 015062 LET R2 := #DZREG
2621 015062 012702 002254 MOV #DZREG,R2
2622 015066 LET RCVCSR(R2) := RCVCSR(R1)
2623 015066 016162 000000 000000 MOV RCVCSR(R1),RCVCS
2624 015074 LET RCVBUF(R2) := RCVBUF(R1)
2625 015074 016162 000002 000002 MOV RCVBUF(R1),RCVBU
2626 015102 LET XMTCSR(R2) := XMTCSR(R1)
2627 015102 016162 000004 000004 MOV XMTCSR(R1),XMTCS
2628 015110 LET XMTBUF(R2) := XMTBUF(R1)
2629 015110 016162 000006 000006 MOV XMTBUF(R1),XMTBU
2630 015116 POP R2
2631 015116 012602 MOV (SP)+,R2
2632 015120 POP R1
2633 015120 012601 MOV (SP)+,R1
2634 015122 ERRHRD 24,MSG24,AREA24
2635 015122 104463 TRAP T$ERCODE
2636 015124 000030 .WORD 24
2637 015126 003111 .WORD MSG24
2638 015130 004262 .WORD AREA24
2639 015132 RETURN FROR
2640 015132 000261 SEC
2641 015134 000404 BR 50001$
2642 015136 ENDIF
2643 015136 50010$:
2644 015136 ELSE
2645 015136 000402 BR 50011$
2646 015140 50006$:
2647 015140 RETURN ERROR
2648 015140 000261 SEC
2649 015142 000401 BR 50001$
2650 015144 ENDIF
2651 015144 50011$:
2652 015144 ENDRTN
2653 015144 50000$:
2654 015144 000241 CLC
2655 015146 50001$:
2656 015146 010406 MOV R4,SP
2657 015150 012604 MOV (SP)+,R4
2658 015152 000207 RTS
2659 015154 000000 PC
2660 015156 000000
2661
2662
2663
2664
2665
2666
2667 *****
2668 : ROUTINE TO INITIALIZE A DEVICE +
2669 : IF 'INIT' IS NON-ZERO, THE ROUTINE WILL EXECUTE AND SET IT +
2670 : RETURNS WITH AN ERROR IF CTS CANNOT BE SET +
2671 :
```



```

2672
2673
2674
2675
2676
2677 015160
2678 015160
2679 015160
2680 015160
2681 015160
2682 015160 010446
2683 015162 010604
2684 015164 162706 000002
2685
2686 015170
2687 015170 010146
2688 015172
2689 015172 010246
2690
2691
2692
2693 015174
2694 015174 016701 165000
2695
2696
2697
2698 015200
2699 015200 052761 000020 000000
2700 015206
2701 015206
2702 015206
2703 015206 032761 000020 000000
2704 015214 001374
2705
2706
2707
2708 015216
2709 015216 016767 000412 164666
2710 015224
2711 015224 000367 164662
2712 015230
2713 015230 056761 164656 000004
2714
2715
2716
2717 015236
2718 015236 005767 164746
2719 015242 001035
2720 015244
2721 015244 012700 000100
2722 015250 104027
2723 015252
2724 015252 036761 164634 000006
2725 015260 001026
2726 015262
2727 015262 012702 002254

```

RETURNS WITH ERROR AND 'ABORT' SET IF THE DEVICE DOES NOT EXIST

 ROUTINE DEVINIT ;[3.2.20.1.5 AND 3.2.20.7.2] DEVINIT:
 DECLARE LOCALS LOCAL <A>
 ALLOCATE
 MOV R4,-(SP)
 MOV SP,R4
 SUB #2,SP
 PUSH R1
 MOV R1,-(SP)
 PUSH R2
 MOV R2,-(SP)
 :GET DEVICE ADDRESS
 :
 LET R1 := PTABLE
 MOV PTABLE,R1
 :DO A MASTER CLEAR ON THE DZ AND WAIT FOR THE CLEAR BIT TO GO DOWN
 :
 LET RCVCSR(R1) := RCVCSR(R1) SET.BY #MASCLR
 BIS #MASCLR,RCVCSR(R
 REPEAT
 UNTIL #MASCLR NOTSETIN RCVCSR(R1) 50002\$:
 BIT #MASCLR,RCVCSR(R
 BNE 50002\$
 :GET THE LINE MASK AND SET THE RIGHT DATA TERMINAL READY BIT IN THE DZ
 :
 LET SCRTCH := \$LINE
 MOV \$LINE,SCRTCH
 LET SCRTCH := SWAP SCRTCH
 SWAB SCRTCH
 LET XMTCSR(R1) := XMTCSR(R1) SET.BY SCRTCH
 BIS SCRTCH,XMTCSR(R1)
 :WAIT 10MSEC AND SEE IF CARRIER IS UP
 :
 IF NMOD EQ #0 THEN
 TST NMOD
 BNE 50003\$
 WAITUS #100
 MOV #100,R0
 EMT C\$WTU
 IF SCRTCH NOTSETIN XMTBUF(R1) THEN
 BIT SCRTCH,XMTBUF(R1)
 BNE 50004\$
 LET R2 := #DZREG
 MOV #DZREG,R2

```
2728 015266          LET RCVCSR(R2) := RCVCSR(R1)
2729 015266 016162 000000 000000          MOV      RCVCSR(R1),RCVCS
2730 015274          LET RCVBUF(R2) := RCVBUF(R1)          MOV      RCVBUF(R1),RCVBU
2731 015274 016162 000002 000002          MOV      XMTCSR(R1),XMTCS
2732 015302          LET XMTCSR(R2) := XMTCSR(R1)          MOV      XMTBUF(R1),XMTBU
2733 015302 016162 000004 000004          MOV      XMTCSR(R1),XMTCS
2734 015310          LET XMTBUF(R2) := XMTBUF(R1)          MOV      XMTBUF(R1),XMTBU
2735 015310 016162 000006 000006          POP R2
2736 015316          POP R1
2737 015316 012602          MOV      (SP)+,R2
2738 015320          POP R1
2739 015320 012601          MOV      (SP)+,R1
2740 015322          ERRHRD 05,MSG05,AREA30
2741 015322 104463          TRAP    T$ERCODE
2742 015324 000005          .WORD  5
2743 015326 002375          .WORD  MSG05
2744 015330 004534          .WORD  AREA30
2745 015332          RETURN ERROR
2746 015332 000261          SEC
2747 015334 000534          BR      50001$
2748 015336          ENDIF
2749 015336          ENDIF          50004$:
2750 015336          ENDIF          50003$:
2751 015336
2752
2753 ;GET THE LINE NUMBER, STOP CODE, CHARACTER LENGTH AND BAUD RATE FROM
2754 ;THE PTABLE AND CLEAR EXTRANEIOUS BITS. SET THE RECEIVER ENABLE BIT
2755 ;AND LOAD THE DZ LINE CONTROL REGISTER.
2756 ;
2757 015336          LET R2 := #PTABLE + #6
2758 015336 012702 002200          MOV      #PTABLE,R2
2759 015342 062702 000006          ADD      #6,R2
2760 015346          LET R2 := (R2)
2761 015346 011202          MOV      (R2),R2
2762 015350          LET R2 := R2 CLR.BY #300
2763 015350 042702 000300          BIC      #300,R2
2764 015354          LET R2 := R2 SET.BY #10000
2765 015354 052702 010000          BIS      #10000,R2
2766 015360          LET RCVBUF(R1) :- R2
2767 015360 010261 000002          MOV      R2,RCVBUF(R1)
2768
2769 ;GET THE DEVICE PRIORITY AND SET UP THE INTERRUPT HANDLERS AND
2770 ;CLEAR THE NEW MESSAGE FLAG (NMSG)
2771 ;
2772 015364          LET DEVPRI := #PTABLE + #4
2773 015364 012767 002200 164620          MOV      #PTABLE,DEVPRI
2774 015372 062767 000004 164612          ADD      #4,DEVPRI
2775 015400          LET DEVPRI :B= @DEVPRI
2776 015400 117767 164606 164604          MOVB   @DEVPRI,DEVPRI
2777 015406          LET DEVPRI := DEVPRI CLR.BY #HYBYTE
2778 015406 042767 177400 164576          BIC      #HYBYTE,DEVPRI
2779 015414          LET A(R4) :- #PTABLE + #2
2780 015414 012764 002200 177776          MOV      #PTABLE,A(R4)
2781 015422 062764 000002 177776          ADD      #2,A(R4)
2782 015430          SETVEC @A(R4),#RINT,DEVPRI
2783 015430 016746 164556          MOV      DEVPRI,-(SP)
```

2784	015434	012746	016066			MOV	#RINT,-(SP)
2785	015440	017446	177776			MOV	@A(R4),-(SP)
2786	015444	012746	000003			MOV	#3,-(SP)
2787	015450	104037				EMT	C\$SVEC
2788	015452	062706	000010			ADD	#10,SP
2789	015456				LET NMSG := #0		
2790	015456	005067	000176			CLR	NMSG
2791	015462				LET A(R4) := @A(R4) + #4		
2792	015462	017464	177776	177776		MOV	@A(R4),A(R4)
2793	015470	062764	000004	177776		ADD	#4,A(R4)
2794	015476				SETVEC A(R4),#XINT,DEVPRI		
2795	015476	016746	164510			MOV	DEVPRI,-(SP)
2796	015502	012746	015720			MOV	#XINT,-(SP)
2797	015506	016446	177776			MOV	A(R4),-(SP)
2798	015512	012746	000003			MOV	#3,-(SP)
2799	015516	104037				EMT	C\$SVEC
2800	015520	062706	000010			ADD	#10,SP
2801							
2802					:GET COPIES OF THE DEVICE BUFFER REGISTER ADDRESSES		
2803							
2804	015524				LET XMTADD := R1 + #6		
2805	015524	010167	000106			MOV	R1,XMTADD
2806	015530	062767	000006	00C100		ADD	#6,XMTADD
2807	015536				LET RCVADD := R1 + #2		
2808	015536	010167	000076			MOV	R1,RCVADD
2809	015542	062767	000002	000070		ADD	#2,RCVADD
2810							
2811					:SET COUNTS FOR 4 SYNC'S AND FOUR FILL CHARACTERS		
2812							
2813	015550				LET SYNCNT := #4		
2814	015550	012767	000004	000066		MOV	#4,SYNCNT
2815	015556				LET FLLCNT := #4		
2816	015556	012767	000004	000062		MOV	#4,FLLCNT
2817							
2818					:USE LINE MASK TO ENABLE THE TRANSMITTER...ENABLE THE MASTER SCAN		
2819					:ENABLE TRANSMIT AND RECEIVE INTERRUPTS		
2820							
2821	015564				LET XMTCSR(R1) := XMTCSR(R1) SET.BY \$LINE		
2822	015564	056761	000044	000004		BIS	\$LINE,XMTCSR(R1)
2823	015572				LET RCVCSR(R1) := RCVCSR(R1) SET.BY #MSCNEN		
2824	015572	052761	000040	000000		BIS	#MSCNEN,RCVCSR(R1)
2825	015600				LET RCVCSR(R1) := RCVCSR(R1) SET.BY #TXINTE		
2826	015600	052761	040000	000000		BIS	#TXINTE,RCVCSR(R1)
2827	015606				LET RCVCSR(R1) := RCVCSR(R1) SET.BY #RXINTE		
2828	015606	052761	000100	000000		BIS	#RXINTE,RCVCSR(R1)
2829	015614				POP R2		
2830	015614	012602				MOV	(SP)+,R2
2831	015616				POP R1		
2832	015616	012601				MOV	(SP)+,R1
2833							
2834					:SET INIT COMPLETE FLAG		
2835							
2836	015620				LET INIT := INIT + #1		
2837	015620	005267	000024			INC	INIT
2838	015624				ENDRTN		
2839	015624						

50000\$:

```

2840 015624 000241
2841 015626
2842 015626 010406
2843 015630 012604
2844 015632 000207
2845 015634 000000
2846
2847
2848 015636 000000
2849 015640 000000
2850 015642 000000
2851 015644 000000
2852 015646 000000
2853 015650 000000
2854 015652 000000
2855 015654 000000
2856 015656 000000
2857 015660 000000
2858 015662 000000
2859 015664 000000
2860 015666 000000
2861
2862
2863
2864 015670 000000
2865 015672 000000
2866 015674 000000
2867 015676 000000
2868 015700 000000
2869 015702 000000
2870 015704 000000
2871 015706 000000
2872 015710 000000
2873 015712 000000
2874 015714 000000
2875 015716 000000

$LINE: 0
:
XMTADD: 0 ; TRANSMIT BUFFER REGISTER ADDRESS
RCVADD: 0 ; RECEIVE BUFFER REGISTER ADDRESS
TIMER: 0 ; TIMER FOR CLEAR TO SEND OPERATION
SYNCNT: 0 ; NUMBER OF SYNC CHARACTERS INSERTED BEFORE DDCMP MESSAGE
FLLCNT: 0 ; NUMBER OF FILL CHARACTERS AFTER MESSAGE
INIT: 0 ; INITIALIZE COMPLETE FLAG
FIL: 0 ; FILL CHARACTER
XBUFFER: 0 ; CURRENT TRANSMIT BUFFER POINTER
XSIZE: 0 ; CURRENT TRANSMIT BUFFER SIZE
NMSG: 0 ; NEW MESSAGE EXPECTED FLAG
RTEMP1: 0 ; CURRENT RECEIVE BUFFER POINTER
RTEMP2: 0 ; CURRENT RECEIVE BUFFER SIZE
CNTFLG: 0 ; CONTROL FLAG FOR INPUTTING DDCMP MESSAGES
:
RECEIVER QUEUES
:
RQUEUE: 0 ; QUEUE POINTER FOR BUFFER PROCESSING
RQUE: 0 ; QUEUE POINTER FOR NEXT AVAILABLE BUFFER
RQUET: 0 ; QUEUE OF BUFFER ADDRESSES AND SIZES
0
0
0
RQUEB: 0
RBQUE: 0 ; QUEUE POINTER FOR NEXT AVAILABLE BUFFER AREA
RBQUET: 0 ; QUEUE OF BUFFER AREAS
0
RBQUEB: 0
  
```

```

CLC
50001$:
MOV R4, SP
MOV (SP)+, R4
RTS PC
  
```

```
2876      :////////////////////TRANSMISSION INTERRUPT HANDLER////////////////////
2877      :
2878      :
2879 015720      :
2880      :
2881      :
2882      :
2883 015720      :
2884 015720 016700 164266      :
2885 015724 104041      :
2886      :
2887      :
2888      :
2889 015726      :
2890 015726 005767 177712      :
2891 015732 001047      :
2892      :
2893      :
2894      :
2895      :
2896 015734      :
2897 015734 010146      :
2898 015736      :
2899 015736 005767 177714      :
2900 015742 003413      :
2901      :
2902      :
2903      :
2904 015744      :
2905 015744 117701 177704      :
2906 015750      :
2907 015750 042701 177400      :
2908 015754      :
2909 015754 110177 177656      :
2910 015760      :
2911 015760 005267 177670      :
2912 015764      :
2913 015764 005367 177666      :
2914      :
2915      :
2916      :
2917 015770      :
2918 015770 000426      :
2919 015772      :
2920 015772      :
2921 015772 005767 177650      :
2922 015776 001406      :
2923      :
2924      :
2925      :
2926 016000      :
2927 016000 056777 177646 177630      :
2928 016006      :
2929 016006 005367 177634      :
2930 016012      :
2931 016012 000415      :
```

////////////////////TRANSMISSION INTERRUPT HANDLER////////////////////
BGNSRV XINT ;[3.2.20.1.7 AND 3.2.20.7.4]
SET PROCESSOR PRIORITY LEVEL
SETPRI DEVPRI
MOV DEVPRI,R0
EMT C\$SPRI
IF SYNC COUNT EQUAL TO ZERO...
IF SYNCNT EQ #0 THEN
TST SYNCNT
BNE 50005\$
CHECK BUFFER SIZE.
IF GREATER THAN ZERO, THEN....
PUSH R1
MOV R1,-(SP)
IF XSIZE GT #0 THEN
TST XSIZE
BLE 50006\$
SEND THE NEXT CHARACTER AND UPDATE POINTER AND COUNT
LET R1 :B= @XBUFFER
MOVB @XBUFFER,R1
LET R1 := R1 CLR.BY #HYBYTE
BIC #HYBYTE,R1
LET @XMTADD :B= R1
MOVB R1,@XMTADD
LET XBUFFER := XBUFFER + #1
INC XBUFFER
LET XSIZE := XSIZE - #1
DEC XSIZE
IF ALL CHARACTERS SENT, CHECK FILL COUNT.
ELSE
BR 50007\$
50006\$:
IF FLLCNT NE #0 THEN
TST FLLCNT
BEQ 50010\$
IF FILL COUNT ISN'T ZERO SEND A FILL CHAR. AND DECREMENT COUNT
LET @XMTADD := @XMTADD SET.BY FIL
BIS FIL,@XMTADD
LET FLLCNT := FLLCNT - #1
DEC FLLCNT
ELSE
BR 50011\$

```

2932 016014                                50010$:
2933
2934      :
2935      :   IF ALL TRANSMIT DONE, DISABLE TRANSMISSION AND SET DONE FLAG
2936      :
2937 016014 016701 177616                    LET R1 := XMTADD - #6
2938 016020 162701 000006                    MOV     XMTADD,R1
2939 016024                                SUB     #6,R1
2940 016024 005267 177124                    LET XDONE := XDONE + #1
2941 016030                                INC     XDONE
2942 016030 104021                                ABORTWAIT
2943 016032                                EMT     C$ABRT
2944 016032 046761 177576 000004            LET XMTCSR(R1) := XMTCSR(R1) CLR.BY $LINE
2945 016040                                BIC     $LINE,XMTCSR(R1)
2946 016040 042761 040000 000000            LET RCVCSR(R1) := RCVCSR(R1) CLR.BY #TXINTE
2947 016046                                BIC     #TXINTE,RCVCSR(R
2948 016046                                ENDIF
2949 016046                                ENDIF
2950 016046                                50011$:
2951 016046                                50007$:
2952 016046 012601                                POP R1
2953
2954      :
2955      :   ...OTHERWISE, SEND A SYNC AND DECREMENT THE SYNC COUNT
2956      :
2957 016050                                ELSE
2958 016052 000405                                BR      50012$
2959 016052                                50005$:
2960 016052 112777 000377 177556            LET @XMTADD :B= #SYNC
2961 016060                                MOV     #SYNC,@XMTADD
2962 016060 005367 177560                                LET SYNCNT := SYNCNT - #1
2963 016064                                DEC     SYNCNT
2964 016064                                ENDIF
2965
2966      :
2967      :   DO A SUPERVISOR-TYPE RT!
2968 016064                                ENDSRV
2969 016064                                L10027:
2970 016064 000002                                RTI
2971
2972
2973
2974
2975
2976
2977      :
2978      :   //////////////////////////////////////
2979 016066      :   RECEPTION INTERRUPT ROUTINE
2980      :   //////////////////////////////////////
2981      :   BGNSRV RINT ;[3.2.1.1.1, 3.2.20.2.1 AND 3.2.20.9.1]
2982      :
2983      :   SET PROCESSOR PRIORITY
2984 016066 016700 164120                    SETPRI DEVPRI
2985 016072 104041                                MOV     DEVPRI,RO
2986 016074                                EMT     C$SPRI
2987 016074 010146                                PUSH R1
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
  
```

```

2988
2989      : IF NEW MESSAGE FLAG IS RESET...
2990
2991      :
2992      : GET THE BYTE
2993
2994      016076      LET R1 := @RCVADD
2995      016076 017701 177536      MOV      @RCVADD,R1
2996
2997      : IF ERROR, REPORT IT AND EXIT IMMEDIATELY (FORCE RESTART)
2998
2999      016102      IF #70000 SETIN R1 THEN
3000      016102 032701 070000      BIT      #70000,R1
3001      016106 001407      BEQ      50013$
3002      016110      LET RDONE := R1
3003      016110 010167 177042      MOV      R1,RDONE
3004      016114      LET NMSG := #0
3005      016114 005067 177540      CLR      NMSG
3006      016120      POP R1
3007      016120 012601      MOV      (SP)+,R1
3008      016122 000167 000500      JMP REXIT
3009      016126      ENDIF
3010      016126
3011      016126      LET R1 := R1 CLR.BY #HYBYTE      50013$:
3012      016126 042701 177400      BIC      #HYBYTE,R1
3013      016132
3014      016132 005767 177522      IF NMSG EQ #0 THEN
3015      016136 001132      TST      NMSG
3016
3017      :
3018      : CHECK FOR SYNC/FILL CHARACTER
3019      : (LOOKING FOR FIRST NON-SYNC/NON-FILL)
3020
3021      016140      IFB R1 NE #SYNC ANDB R1 NE FIL THEN
3022      016144 120127 000377      CMPB    R1,#SYNC
3023      016146 001526      BEQ      50015$
3024      016152 001523      CMPB    R1,FIL
3025
3026      :
3027      : NOT FILL OR SYNC. CHECK FOR ENQ (DDCMP CONTROL MESSAGE).
3028
3029      016154      IFB R1 EQ #ENQ THEN
3030      016160 120127 000005      CMPB    R1,#ENQ
3031
3032      :
3033      : IF IT IS AN ENQ, SET EXPECTED BYTE COUNT TO 7, SET MESSAGE SIZE
3034      : TO 10 (OCTAL) AND UPDATE THE BUFFER QUEUES
3035
3036      016162 017767 177522 177472      LET RTEMP1 := @RBQUE
3037      016170      LET @RQUE := RTEMP1      MOV      @RBQUE,RTEMP1
3038      016170 016777 177466 177474      LET @RTEMP1 := R1      MOV      RTEMP1,@RQUE
3039      016176      LET @RTEMP1 := R1
3040      016176 110177 177460      MOVB   R1,@RTEMP1
3041      016202      LET RTEMP1 := RTEMP1 + #1
3042      016202 005267 177454      INC      RTEMP1
3043      016206      LET RBQUE := RBQUE + #2
    
```

3044	016206	062767	000002	177474				
3045	016214				IF RBQUE GT #RBQUEB THEN			ADD #2, RBQUE
3046	016214	026727	177470	015716				CMP RBQUE, #RBQUEB
3047	016222	003403						BLE 50017\$
3048	016224				LET RBQUE := #RBQUET			
3049	016224	012767	015712	177456				MOV #RBQUET, RBQUE
3050	016232				ENDIF			
3051	016232						50017\$:	
3052	016232				LET RTEMP2 := #7			
3053	016232	012767	000007	177424				MOV #7, RTEMP2
3054	016240				LET RQUE := RQUE + #2			
3055	016240	062767	000002	177424				ADD #2, RQUE
3056	016246				LET @RQUE := #10			
3057	016246	012777	000010	177416				MOV #10, @RQUE
3058	016254				LET RQUE := RQUE + #2			
3059	016254	062767	000002	177410				ADD #2, RQUE
3060	016262				IF RQUE GT #RQUEB THEN			
3061	016262	026727	177404	015706				CMP RQUE, #RQUEB
3062	016270	003403						BLE 50020\$
3063	016272				LET RQUE := #RQUET			
3064	016272	012767	015674	177372				MOV #RQUET, RQUE
3065	016300				ENDIF			
3066	016300						50020\$:	
3067								
3068					...			
3069					SET COUNTER FLAG TO -1 (MESSAGE SIZE COUNT COMPLETE)			
3070	016300				LET CNTFLG := #-1			
3071	016300	012767	177777	177360				MOV #-1, CNTFLG
3072	016306				ELSE			
3073	016306	000443						BR 50021\$
3074	016310						50016\$:	
3075					...			
3076					NOT FILL, SYNC OR ENQ. CHECK FOR SOH OR DLE (DDCMP DATA MESSAGE)			
3077					...			
3078	016310				IFB R1 EQ #SOH ORB R1 EQ #DLE THEN			
3079	016310	120127	000201					CMPB R1, #SOH
3080	016314	001403						BEQ 50022\$
3081	016316	120127	000220					CMPB R1, #DLE
3082	016322	001033						BNE 50023\$
3083	016324						50022\$:	
3084					...			
3085					MOVE QUEUE POINTER TO SIZE ENTRY AND SET COUNTER FLAG TO 1 (EXPECT			
3086					FIRST COUNT BYTE)			
3087					...			
3088	016324				LET RTEMP1 := @RBC E			
3089	016324	017767	177360	177330				MOV @RBQUE, RTEMP1
3090	016332				LET @RQUE := RTEMP1			
3091	016332	016777	177324	177332				MOV RTEMP1, @RQUE
3092	016340				LET @RTEMP1 :B= R1			
3093	016340	110177	177316					MOVB R1, @RTEMP1
3094	016344				LET RTEMP1 := RTEMP1 + #1			
3095	016344	005267	177312					INC RTEMP1
3096	016350				LET RBQUE := RBQUE + #2			
3097	016350	062767	000002	177332				ADD #2, RBQUE
3098	016356				IF RBQUE GT #RBQUEB THEN			
3099	016356	026727	177326	015716				CMP RBQUE, #RBQUEB


```

3100 016364 003403          *          BLE      50024$
3101 016366                LET RBQUE := #RBQUET          MOV      #RBQUET,RBQUE
3102 016366 012767 015712 177314          ENDIF
3103 016374                50024$:
3104 016374                LET RQUE := RQUE + #2          ADD      #2,RQUE
3105 016374                LET CNTFLG := #1          MOV      #1,CNTFLG
3106 016374 062767 000002 177270          ELSE
3107 016402                BR      50025$
3108 016402 012767 000001 177256          50023$:
3109 016410
3110 016410 000402
3111 016412
3112
3113          :
3114          :   BYTE WASN'T ANY OF THE LEGAL CHARACTERS... FORCE
3115          :   A RESTART BY FORCING THE NEW MESSAGE FLAG TO RESET ON EXIT.
3116 016412                LET NMSG := NMSG - #1          DEC      NMSG
3117 016412 005367 177242          ENDIF
3118 016416                50025$:
3119 016416                50021$:
3120 016416
3121 016416
3122          :
3123          :   SET THE NEW MESSAGE FLAG
3124          :   EXIT ROUTINE
3125          :
3126 016416                LET NMSG := NMSG + #1          INC      NMSG
3127 016416 005267 177236          ENDIF
3128 016422                50015$:
3129 016422
3130          :
3131          :   NEW MESSAGE FLAG IS SET... CONTINUE BUILDING MESSAGE
3132          :
3133 016422                ELSE
3134 016422 000500                BR      50026$
3135 016424                50014$:
3136
3137          :
3138          :   NOT A NEW MESSAGE...GET BYTE AND STORE IT
3139 016424                LET @RTEMP1 :B= R1          MOVB     R1,@RTEMP1
3140 016424 110177 177232
3141
3142          :
3143          :   IF COUNTER FLAG IS GREATER THAN ZERO, BYTE IS ALSO THE FIRST COUNT BYTE
3144 016430                IF CNTFLG GT #0 THEN
3145 016430 005767 177232                TST     CNTFLG
3146 016434 003413                BLE     50027$
3147
3148          :
3149          :   PUT THE BYTE IN RTEMP2 (EXPECTED COUNT), UPDATE THE BUFFER POINTER
3150          :   DECREMENT THE COUNTER FLAG (SET TO GET SECOND COUNT BYTE) AND
3151          :   GO DO AN RTI
3152 016436                LET RTEMP2 :B= @RTEMP1          MOVB     @RTEMP1,RTEMP2
3153 016436 117767 177220 177220                LET RTEMP2 := RTEMP2 CLR.BY #HYBYTE          BIC     #HYBYTE,RTEMP2
3154 016444
3155 016444 042767 177400 177212

```

3156	016452			LET RTEMP1 := RTEMP1 + #1		
3157	016452	005267	177204		INC	RTEMP1
3158	016456			LET CNTFLG := CNTFLG - #1		
3159	016456	005367	177204		DEC	CNTFLG
3160	016462			ELSE		
3161	016462	000460			BR	50030\$
3162	016464				50027\$:	
3163						
3164				COUNTER FLAG NOT GREATER THAN ZERO, IS IT ZERO?		
3165						
3166	016464			IF CNTFLG EQ #0 THEN		
3167	016464	005767	177176		TST	CNTFLG
3168	016470	001041			BNE	50031\$
3169						
3170				IF ZERO, BYTE IS ALSO SECOND BYTE OF COUNT (6 HIGH ORDER BITS)		
3171				GET IT, MASK IT AND PUT IT INTO RTEMP2 (EXPECTED COUNT)		
3172						
3173	016472			PUSH R1		
3174	016472	010146			MOV	R1,-(SP)
3175	016474			LET R1 := @RTEMP1		
3176	016474	117701	177162		MOVB	@RTEMP1,R1
3177	016500			LET RTEMP1 := RTEMP1 + #1		
3178	016500	005267	177156		INC	RTEMP1
3179	016504			LET R1 := SWAP R1		
3180	016504	000301			SWAB	R1
3181	016506			LET R1 := R1 CLR.BY #140377		
3182	016506	042701	140377		BIC	#140377,R1
3183	016512			LET RTEMP2 := RTEMP2 + R1		
3184	016512	060167	177146		ADD	R1,RTEMP2
3185	016516			POP R1		
3186	016516	012601			MOV	(SP)+,R1
3187						
3188				PUT THE COMPLETE SIZE IN THE QUEUE (EXPECTED COUNT +		
3189				12 (OCTAL) BYTES OF HEADER AND CRC)		
3190						
3191	016520			LET @RQUE := RTEMP2		
3192	016520	016777	177140 177144		MOV	RTEMP2,@RQUE
3193	016526			LET @RQUE := @RQUE + #12		
3194	016526	062777	000012 177136		ADD	#12,@RQUE
3195						
3196				ADJUST EXPECTED COUNT TO INCLUDE THE REMAINDER OF THE HEADER AND		
3197				THE DATA CRC		
3198						
3199	016534			LET RTEMP2 := RTEMP2 + #7		
3200	016534	062767	000007 177122		ADD	#7,RTEMP2
3201						
3202				UPDATE THE QUEUE AND DECREMENT THE COUNTER FLAG (COUNT COMPLETE)		
3203						
3204	016542			LET RQUE := RQUE + #2		
3205	016542	062767	000002 177122		ADD	#2,RQUE
3206	016550			IF RQUE GT #RQUEB THEN		
3207	016550	026727	177116 015706		CMP	RQUE,#RQUEB
3208	016556	003403			BLE	50032\$
3209	016560			LET RQUE := #RQUET		
3210	016560	012767	015674 177104		MOV	#RQUET,RQUE
3211	016566			ENDIF		

```

3212 016566                                50032$:
3213 016566                                LET CNTFLG := CNTFLG - #1
3214 016566 005367 177074                    DEC     CNTFLG
3215                                     :
3216                                     : GO DO AN RTI
3217                                     :
3218 016572                                ELSE
3219 016572 000414                            BR     50033$
3220 016574                                50031$:
3221                                     :
3222                                     : COUNTER FLAG IS LESS THAN ZERO... THIS IS A DATA BYTE
3223                                     : UPDATE POINTER AND COUNT
3224                                     :
3225 016574                                LET RTEMP1 := RTEMP1 + #1
3226 016574 005267 177062                    INC     RTEMP1
3227 016600                                LET RTEMP2 := RTEMP2 - #1
3228 016600 005367 177060                    DEC     RTEMP2
3229                                     :
3230                                     : IF COUNT HAS REACHED ZERO, END OF MESSAGE. SET DONE FLAG.
3231 016604                                IF RTEMP2 EQ #0 THEN
3232 016604 005767 177054                    TST     RTEMP2
3233 016610 001005                            BNE     50034$
3234 016612                                LET RDONE := RDONE + #1
3235 016612 005267 176340                    INC     RDONE
3236                                     :
3237                                     : ABORT ANY WAITS AND RESET NEW MESSAGE FLAG
3238                                     :
3239 016616                                ABORTWAIT
3240 016616 104021                            EMT     C$ABRT
3241 016620                                LET NMSG : #0
3242 016620 005067 177034                    CLR     NMSG
3243 016624                                ENDIF
3244 016624                                50034$:
3245 016624                                ENDIF
3246 016624                                50033$:
3247 016624                                ENDIF
3248 016624                                50030$:
3249 016624                                ENDIF
3250 016624                                50026$:
3251                                     :
3252                                     : DO RTI
3253                                     :
3254 016624                                POP R1
3255 016624 012601                            MOV     (SP)+,R1
3256 016626  REXIT: ENDSRV
3257 016626                                L10030:
3258 016626 000002                            RTI
3259
3260
3261
3262
3263 :+++++
3264 :
3265 : ROUTINE TO GET AN INPUT BUFFER FROM THE INPUT QUEUES
3266 :
3267 : WILL RETURN AN ERROR IF TERMINAL DOES NOT RESPOND

```

```

3268
3269
3270
3271
3272
3273
3274 016630
3275 016630
3276
3277
3278
3279 016630
3280 016630 005067 177006
3281
3282
3283
3284 016634
3285 016634
3286 016634
3287 016634 012700 000100
3288 016640 104027
3289 016642
3290 016642 005267 176774
3291
3292
3293
3294 016646
3295 016646 026767 176770 163244
3296 016654 003003
3297 016656 005767 176274
3298 016662 001764
3299 016664
3300
3301
3302
3303 016664
3304 016664 005067 176770
3305
3306
3307
3308 016670
3309 016670 026767 176746 163222
3310 016676 002044
3311
3312
3313
3314
3315 016700
3316 016700 005067 176736
3317 016704
3318 016704 005767 176246
3319 016710 003426
3320
3321
3322
3323 016712
  
```

```

:
: ARGUMENT (IF SET) INHIBITS ERROR PRINTOUT. THIS ALLOWS
: THE PROGRAM TO RETRY THE STRT-STACK SEQUENCE WITHOUT HAVING
: AN ERROR MESSAGE EACH TIME
:
:*****
: ROUTINE RCVE ;[3.2.1.1, 3.2.20.2 AND 3.2.20.8]
: RCVE:
:
: RESET TIMER
:
: LET TIMER := #0
:
: GO TO SUPERVISOR FOR AWHILE AND INCREMENT TIMER
:
: REPEAT
:   WAITUS #100 ;[3.2.1.1.4, 3.2.20.2.4, 3.2.20.8.4]
:   MOV #100,RO
:   EMT C$WTU
:   LET TIMER := TIMER + #1
:   INC TIMER
:
: LOOP UNTIL TIME LIMIT EXCEEDED OR RECEIVER DONE FLAG IS SET
:
: UNTIL TIMER GT TIMLIM OR RDONE NE #0
:   CMP TIMER,TIMLIM
:   BGT 50003$
:   TST RDONE
:   BEQ 50002$
:   50003$:
:
: RESET THE NEW MESSAGE FLAG
:
: LET NMSG := #0
:
: CLR NMSG
:
: IF THE TIME LIMIT WASN'T EXCEEDED...
:
: IF TIMER LT TIMLIM THEN
:   CMP TIMER,TIMLIM
:   BGE 50004$
:
: RESET THE TIMER
: CHECK FOR ERROR (RDONE LESS THAN ZERO)
:
: LET TIMER := #0
:
: CLR TIMER
:
: IF RDONE GT #0 THEN
:   TST RDONE
:   BLE 50005$
:
: NOT ERROR, GET BUFFER ADDRESS AND SIZE FROM QUEUE AND UPDATE QUEUE
:
: LET INBUF := @RQUEUE
  
```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 29-DEC-78 11:50 PAGE 68D 6
CZDZCB.P11 21-NOV-78 16:33
                                                                    SEQ 0068

3324 016712 017767 176752 163236
3325 016720
3326 016720 062767 000002 176742
3327 016726
3328 016726 017767 176736 163224
3329 016734
3330 016734 062767 000002 176726
3331 016742
3332 016742 026727 176722 015706
3333 016750 003403
3334 016752
3335 016752 012767 015674 176710
3336 016760
3337 016760
3338
3339
3340
3341 016760
3342 016760 005067 176172
3343 016764
3344 016764 000410
3345 016766
3346
3347
3348
3349 016766
3350 016766 104463
3351 016770 000032
3352 016772 003210
3353 016774 004454
3354 016776
3355 016776 005067 176154
3356 017002
3357 017002 000261
3358 017004 000440
3359 017006
3360 017006
3361 017006
3362 017006 000436
3363 017010
3364
3365
3366
3367 017010
3368 017010 005067 176626
3369 017014
3370 017014 010146
3371 017016
3372 017016 016701 176616
3373 017022 162701 000002
3374 017026
3375 017026 010246
3376 017030
3377 017030 012702 002254
3378 017034
3379 017034 016162 000000 000000

      MOV      @RQUEUE,INBUF
      LET RQUEUE := RQUEUE + #2
      ADD      #2,RQUEUE
      LET INSIZ := @RQUEUE
      MOV      @RQUEUE,INSIZ
      LET RQUEUE := RQUEUE + #2
      ADD      #2,RQUEUE
      IF RQUEUE GT #RQUEB THEN
      CMP      RQUEUE,#RQUEB
      BLE      50006$
      LET RQUEUE := #RQUET
      MOV      #RQUET,RQUEUE
      ENDIF
50006$:
:RESET DONE FLAG AND LEAVE
:
      LET RDONE : #0
      CLR      RDONE
      ELSE
      BR       50007$
50005$:
:ERROR DID OCCUR...RESET THE DONE FLAG, REPORT ERROR, RETURN WITH ERROR
:
      ERRHRD 26,MSG26,AREA26
      TRAP    T$ERCODE
      .WORD   26
      .WORD   MSG26
      .WORD   AREA26
      LET RDONE := #0
      CLR      RDONE
      RETURN  ERROR
      SEC
      BR       50001$
50007$:
      ELSE
      BR       50010$
50004$:
:TIME LIMIT EXCEEDED...RESET TIMER, GET REGISTER CONTENTS
:
      LET TIMER := #0
      CLR      TIMER
      PUSH R1
      MOV      R1,-(SP)
      LET R1 := RCVADD - #2
      MOV      RCVADD,R1
      SUB      #2,R1
      PUSH R2
      MOV      R2,-(SP)
      LET R2 := #DZREG
      MOV      #DZREG,R2
      LET RCV(CSR(R2)) := RCV(CSR(R1))
      MOV      RCV(CSR(R1)),RCVCS

```

```

3380 017042          LET RCVBUF(R2) := RCVBUF(R1)
3381 017042 016162 000002 000002          MOV      RCVBUF(R1),RCVBU
3382 017050          LET XMTCSR(R2) := XMTCSR(R1)
3383 017050 016162 000004 000004          MOV      XMTCSR(R1),XMTCS
3384 017056          LET XMTBUF(R2) := XMTBUF(R1)
3385 017056 016162 000006 000006          MOV      XMTBUF(R1),XMTBU
3386 017064          POP R2
3387 017064 012602          MOV      (SP)+,R2
3388 017066          POP R1
3389 017066 012601          MOV      (SP)+,R1
3390
3391          :PRINT ERROR MESSAGE
3392          :
3393 017070          ERRHRD 31,MSG31,AREA31
3394 017070 104463          TRAP     T$ERCODE
3395 017072 000037          .WORD   31
3396 017074 003301          .WORD   MSG31
3397 017076 004534          .WORD   AREA31
3398
3399          :RETURN WITH ERROR IN ANY CASE
3400          :
3401          RETURN ERROR
3402 017100 000261          SEC
3403 017102 000401          BR      50001$
3404 017104          ENDIF
3405 017104          ENDRTN
3406 017104          50010$:
3407 017104          50000$:
3408 017104 000241          CLC
3409 017106          50001$:
3410 017106 000207          RTS      PC
3411
3412
3413
3414
3415
3416
3417          :*****
3418          : DUMMY ROUTINE FOR COMPATIBILITY
3419          :*****
3420          :
3421 017110          ROUTINE INSERR <A>
3422 017110          INSERR:
3423 017110          50000$:
3424 017110          50001$:
3425 017110          RTS      PC
3426 017110 000207
3427
3428
3429
3430
3431
3432
3433          :*****
3434          : THIS ROUTINE COMPUTES CRC-16 AND REQUIRES THREE ARGUMENTS:
3435          :
  
```

```

3436      :          DATA ADDRESS, DATA SIZE (IN BYTES) AND A WORD LOCATION      +
3437      :          FOR THE COMPUTED CRC                                          +
3438      :          [3.2.1.2.1, 3.2.1.2.4, 3.2.20.1.2, 3.2.20.3.1,              +
3439      :          3.2.20.6.2, 3.2.20.6.4 AND 3.2.20.10.1]                       +
3440      :                                                                           +
3441      :          THIS IS A STANDARD ROUTINE FROM DEC'S NETWORK GROUP           +
3442      :                                                                           +
3443      :          +-----+
3444 017112  ROUTINE CRC <A,B,C>
3445 017112
3446 017112          CRC:
3447 017112  DECLARE LOCALS
3448 017112          LOCAL<D,E>
3449 017112  ALLOCATE
3450 017114  010446          MOV      R4,-(SP)
3451 017116  010604          MOV      SP,R4
3452 017122  162706  000004  SUB      #4,SP
3453 017122          PUSH R1
3454 017124          MOV      R1,-(SP)
3455 017124  010246          PUSH R2
3456 017126          MOV      R2,-(SP)
3457 017126  010346          PUSH R3
3458 017130          MOV      R3,-(SP)
3459 017130  016501  000000  LET R1 := A(R5)
3460 017134          MOV      A(R5),R1
3461 017134  016502  000002  LET R2 := B(R5)
3462 017140          MOV      B(R5),R2
3463 017140  005003          LET R3 := #0
3464 017142          CLR      R3
3465 017142          WHILE R2 GT #0 DO
3466 017142  005702          50002$: TST      R2
3467 017144  003500          BLE      50003$
3468 017146          LET D(R4) := B+(R1)+
3469 017146  112164  177776  MOVVB   (R1)+,D(R4)
3470 017152          LET D(R4) := D(R4) CLR.BY #HYBYTE
3471 017152  042764  177400  177776  BIC     #HYBYTE,D(R4)
3472 017160          LET R2 := R2 - #1
3473 017160  005302          DEC     R2
3474 017162          LET D(R4) := D(R4) XOR R3
3475 017162  010346          MOV      R3,-(SP)
3476 017164  046416  177776  BIC     D(R4),(SP)
3477 017170  040364  177776  BIC     R3,D(R4)
3478 017174  052664  177776  BIS     (SP)+,D(R4)
3479 017200          LET E(R4) := D(R4)
3480 017200  016464  177776  177774  MOV      D(R4),E(R4)
3481 017206          LET D(R4) := D(R4) CLR.BY #177760
3482 017206  042764  177760  177776  BIC     #177760,D(R4)
3483 017214          LET D(R4) := D(R4) SHIFT #1
3484 017214  006364  177776          ASL     D(R4)
3485 017220          LET D(R4) := D(R4) + #CRCTAB
3486 017220  062764  017366  177776  ADD     #CRCTAB,D(R4)
3487 017226          LET D(R4) := @D(R4)
3488 017226  017464  177776  177776  MOV      @D(R4),D(R4)
3489 017234          LET E(R4) := E(R4) CLR.BY #177417
3490 017234  042764  177417  177774  BIC     #177417,E(R4)
3491 017242          LET E(R4) := E(R4) SHIFT -3

```

3492	017242	006264	177774			ASR	E(R4)
3493	017246	006264	177774			ASR	E(R4)
3494	017252	006264	177774			ASR	E(R4)
3495	017256				LET E(R4) := E(R4) + #CRCTAB		
3496	017256	062764	017366	177774		ADD	#CRCTAB,E(R4)
3497	017264				LET E(R4) := E(R4) + #32.		
3498	017264	062764	000040	177774		ADD	#32.,E(R4)
3499	017272				LET E(R4) := @E(R4)		
3500	017272	017464	177774	177774		MOV	@E(R4),E(R4)
3501	017300				LET D(R4) := D(R4) XOR E(R4)		
3502	017300	016446	177774			MOV	E(R4),-(SP)
3503	017304	046416	177776			BIC	D(R4),(SP)
3504	017310	046464	177774	177776		BIC	E(R4),D(R4)
3505	017316	052664	177776			BIS	(SP)+,D(R4)
3506	017322				LET R3 := R3 CLR.BY #LOBYTE		
3507	017322	042703	000377			BIC	#LOBYTE,R3
3508	017326				LET R3 := SWAP R3		
3509	017326	000303				SWAB	R3
3510	017330				LET R3 := R3 XOR D(R4)		
3511	017330	016446	177776			MOV	D(R4),-(SP)
3512	017334	040316				BIC	R3,(SP)
3513	017336	046403	177776			BIC	D(R4),R3
3514	017342	052603				BIS	(SP)+,R3
3515	017344				ENDDO		
3516	017344	000676				BR	50002\$
3517	017346						
3518	017346				LET C(R5) := R3	50003\$:	
3519	017346	010365	000004			MOV	R3,C(R5)
3520	017352				POP R3		
3521	017352	012603				MOV	(SP)+,R3
3522	017354				POP R2		
3523	017354	012602				MOV	(SP)+,R2
3524	017356				POP R1		
3525	017356	012601				MOV	(SP)+,R1
3526	017360				ENDRTN		
3527	017360						
3528	017360					50000\$:	
3529	017360	010406				50001\$:	
3530	017362	012604				MOV	R4,SP
3531	017364	000207				MOV	(SP)+,R4
3532						RTS	PC
3533	017366	000000			CRCTAB: 000000		
3534	017370	140301			140301		
3535	017372	140601			140601		
3536	017374	000500			000500		
3537	017376	141401			141401		
3538	017400	001700			001700		
3539	017402	001200			001200		
3540	017404	141101			141101		
3541	017406	143001			143001		
3542	017410	003300			003300		
3543	017412	003600			003600		
3544	017414	143501			143501		
3545	017416	002400			002400		
3546	017420	142701			142701		
3547	017422	142201			142201		

3548	017424	002100	002100
3549	017426	000000	000000
3550	017430	146001	146001
3551	017432	154001	154001
3552	017434	012000	012000
3553	017436	170001	170001
3554	017440	036000	036000
3555	017442	024000	024000
3556	017444	162001	162001
3557	017446	120001	120001
3558	017450	066000	066000
3559	017452	074000	074000
3560	017454	132001	132001
3561	017456	050000	050000
3562	017460	116001	116001
3563	017462	104001	104001
3564	017464	042000	042000
3565	017466		ENDMOD


```

3622
3623      :LOOK FOR ACK
3624      :
3625      :      CALL TERIN OUT <MCOMP>
3626      017554 162705 000002      SUB      #1*2,R5
3627      017560 004767 174056      JSR      PC,TERIN
3628      017564 012567 000674      MOV      (R5)+,MCOMP
3629
3630      :IF ERROR...INSERT ERROR INTO TABLE (MANUFACTURING TEST ONLY)
3631      :      REPORT THE ERROR, REQUEST LOOP IF ENABLED (EXIT TEST
3632      :      IF NO LOOPING)
3633      :
3634      :      IF.ERROR THEN
3635      017570      103015      BCC      50005$
3636      017572      CALL INSERR IN <#0>
3637      017572 010546      MOV      R5,-(SP)
3638      017574 012745 000000      MOV      #0,-(R5)
3639      017600 004767 177304      JSR      PC,INSERR
3640      017604 012605      MOV      (SP)+,R5
3641      017606      ERRHRD 50,MSG50,AREA40
3642      017606 104463      TRAP    T$ERCODE
3643      017610 000062      .WORD  50
3644      017612 003467      .WORD  MSG50
3645      017614 004262      .WORD  AREA40
3646      017616      CKLOOP
3647      017616 104006      EMT     C$CLP1
3648      017620      EXIT TST
3649      017620 104032      EMT     C$EXIT
3650      017622 000342      .WORD  L10031-.
3651      017624      ENDIF
3652      017624      50005$:
3653
3654      :IF MESSAGE WAS NOT AN ACK...ERROR
3655      :      REPORT ERROR, LOOP IF ENABLED, EXIT TEST IF NOT
3656      :
3657      :      IF MCOMP NE #0 THEN
3658      017624 005767 000634      TST     MCOMP
3659      017630 001407      BEQ     50006$
3660      017632      ERRHRD 50,MSG50,AREA40
3661      017632 104463      TRAP    T$ERCODE
3662      017634 000062      .WORD  50
3663      017636 003467      .WORD  MSG50
3664      017640 004262      .WORD  AREA40
3665      017642      CKLOOP
3666      017642 104006      EMT     C$CLP1
3667      017644      EXIT TST
3668      017644 104032      EMT     C$EXIT
3669      017646 000316      .WORD  L10031-.
3670      017650      ENDIF
3671      017650      50006$:
3672
3673      :FORCE A RESTART ON THE LINK AND WAIT FOR SELFTEST TO COMPLETE
3674      :
3675      017650      LET INIT := #0
3676      017650 005067 175774      CLR     INIT
3677      017654      LET SCRATCH := #0

```

3678	017654	005067	162232			CLR	SCRATCH
3679	017660			REPEAT			
3680	017660				50007\$:		
3681	017660			WAITUS #100			
3682	017660	012700	000100			MOV	#100,R0
3683	017664	104027				EMT	C\$WTU
3684	017666			LET SCRATCH := SCRATCH + #1			
3685	017666	005267	162220			INC	SCRATCH
3686	017672			UNTIL SCRATCH GT STWAIT			
3687	017672	026767	162214	172012		CMP	SCRATCH,STWAIT
3688	017700	003767				BLE	50007\$
3689							
3690				SET UP MESSAGE ASKING FOR RESULTS AND SEND IT			
3691							
3692	017702			LET OUTMSG := MDMSG1 CLR.BY #100001			
3693	017702	016767	000570	162232		MOV	MDMSG1,OUTMSG
3694	017710	042767	100001	162224		BIC	#100001,OUTMSG
3695	017716			LET OUTMSG+2 := MDMSG1+2			
3696	017716	016767	000556	162220		MOV	MDMSG1+2,OUTMSG+
3697	017724			CALL TEROUT			
3698	017724	004767	172630			JSR	PC,TEROUT
3699							
3700				IF ERROR....INSERT ERROR IN TABLE (MANUF. TESTS ONLY), REPORT ERROR,			
3701				LOOP IF ENABLED, EXIT TEST IF NO LOOPING			
3702							
3703	017730			IF.ERROR THEN			
3704	017730	103015				BCC	50010\$
3705	017732			CALL INSERR IN <#0>			
3706	017732	010546				MOV	R5,-(SP)
3707	017734	012745	000000			MOV	#0,-(R5)
3708	017740	004767	177144			JSR	PC,INSERR
3709	017744	012605				MOV	(SP)+,R5
3710	017746			ERRHRD 50,MSG50,AREA40			
3711	017746	104463				TRAP	T\$ERCODE
3712	017750	000062				.WORD	50
3713	017752	003467				.WORD	MSG50
3714	017754	004262				.WORD	AREA40
3715	017756			CKLOOP			
3716	017756	104006				EMT	C\$CLP1
3717	017760			EXIT TST			
3718	017760	104032				EMT	C\$EXIT
3719	017762	000202				.WORD	L10031-
3720	017764			ENDIF			
3721	017764				50010\$:		
3722							
3723				ACK TFRMINAL AND GET ITS RESPONSE			
3724							
3725	017764			REPEAT			
3726	017764				50011\$:		
3727							
3728				GET THE RESPONSE			
3729							
3730	017764			CALL TERIN OUT <MCOMP>			
3731	017764	162705	000002			SUB	#1*2,R5
3732	017770	004767	173646			JSR	PC,TERIN
3733	017774	012567	000464			MOV	(R5)+,MCOMP

```

3734
3735      :
3736      :
3737      :
3738      :
3739      :
3740      :
3741      :
3742      :
3743      :
3744      :
3745      :
3746      :
3747      :
3748      :
3749      :
3750      :
3751      :
3752      :
3753      :
3754      :
3755      :
3756      :
3757      :
3758      :
3759      :
3760      :
3761      :
3762      :
3763      :
3764      :
3765      :
3766      :
3767      :
3768      :
3769      :
3770      :
3771      :
3772      :
3773      :
3774      :
3775      :
3776      :
3777      :
3778      :
3779      :
3780      :
3781      :
3782      :
3783      :
3784      :
3785      :
3786      :
3787      :
3788      :
3789      :

```

020000	103015			HANDLE ERROR IF ONE OCCURED.		
020002				IF.ERROR THEN		
020002	010546			CALL INSERR IN <#0>	BCC	50012\$
020004	012745	000000			MOV	R5,-(SP)
020010	004767	177074			MOV	#0,-(R5)
020014	012605				JSR	PC,INSERR
020016				ERRHRD 50,MSG50,APEA40	MOV	(SP)+,R5
020016	104463				TRAP	T\$ERCODE
020020	000062				.WORD	50
020022	003467				.WORD	MSG50
020024	004262				.WORD	AREA40
020026				CKLOOP		
020026	104006			EXIT TST	EMT	C\$CLP1
020030					EMT	C\$EXIT
020030	104032				.WORD	L10031-
020032	000132			ENDIF		
020034				IF MCOMP EQ #0 THEN	50012\$:	
020034	005767	000424			TST	MCOMP
020040	001026				BNE	50013\$
020042				CALL OUTREP		
020042	004767	000120			JSR	PC,OUTREP
				HANDLE ERROR IF ONE OCCURED WHILE ACK'ING		
				IF.ERROR THEN		
020046	103023			CALL INSERR IN <#0>	BCC	50014\$
020050					MOV	R5,-(SP)
020050	010546				MOV	#0,-(R5)
020052	012745	000000			JSR	PC,INSERR
020056	004767	177026			MOV	(SP)+,R5
020062	012605			CALL INSERR IN <#1>		
020064	010546				MOV	R5,-(SP)
020066	012745	000001			MOV	#1,-(R5)
020072	004767	177012			JSR	PC,INSERR
020076	012605				MOV	(SP)+,R5
020100				ERRHRD 50,MSG50,AREA40		
020100	104463				TRAP	T\$ERCODE
020102	000062				.WORD	50
020104	003467				.WORD	MSG50
020106	004262				.WORD	AREA40
020110				CKLOOP		
020110	104006			EXIT TST	EMT	C\$CLP1
020112					EMT	C\$EXIT
020112	104032				.WORD	L10031-
020114	009050			ENDIF		
020116				ENDIF	50014\$:	
020116					50013\$:	

```

3790
3791      : REPEAT THE LOOP UNTIL A MESSAGE IS RECEIVED
3792      :
3793      020116      UNTIL MCOMP NE #0
3794      020116      005767      000342
3795      020122      001720
3796
3797      : PROCESS THE MESSAGE RECEIVED
3798      :
3799      020124      CALL STINPT
3800      020124      004767      000146
3801
3802      : IF IN ERROR...INSERT ERROR INTO TABLE (MANUF.TESTS ONLY, REPORT
3803      : ERROR, LOOP IF ENABLED, EXIT TEST IF NOT
3804      :
3805      020130      IF.ERROR THEN
3806      020130      103015
3807      020132      CALL INSERR IN <#1>
3808      020132      010546
3809      020134      012745      000001
3810      020140      004767      176744
3811      020144      012605
3812      020146      ERRHRD 50,MSG50,AREA40
3813      020146      104463
3814      020150      000062
3815      020152      003467
3816      020154      004262
3817      020156      CKLOOP
3818      020156      104006
3819      020160      EXIT TST
3820      020160      104032
3821      020162      000002
3822      020164      ENDIF
3823      020164
3824
3825      : END OF TEST
3826      :
3827      020164      ENDTST
3828      020164
3829      020164      104001
3830
3831
3832
3833
3834
3835      : +-----+
3836      : | ROUTINE TO SEND AND ACK IN ORDER TO POLL A TERMINAL |
3837      : | +-----+ |
3838      : | ROUTINE OUTREP |
3839      : | +-----+ |
3840      020166
3841      020166
3842
3843      : INSERT MESSAGE NUMBER (LAST MESSAGE SENT) AND RESPONSE NUMBER
3844      : (LAST MESSAGE RECEIVED)....INSERT STATION ADDRESS
3845
  
```

TST MCOMP
BEQ 50011\$

JSR PC,STINPT

BCC 50015\$

MOV R5,-(SP)
MOV #1,-(R5)
JSR PC,INSERR
MOV (SP)+,R5

TRAP T\$ERCODE
.WORD 50
.WORD MSG50
.WORD AREA40

EMT C\$CLP1

EMT C\$EXIT
.WORD L10031-

50015\$:

L10031:

EMT C\$SETST

OUTREP:

```

3846 020166          LET MSGN :B= MSGNO
3847 020166 116767 162002 000316          MOVB  MSGNO,MSGN
3848 020174          LET REPN :B= RMSGNO
3849 020174 116767 161776 000307          MOVB  RMSGNO,REPN
3850 020202          LET REPA :B= STRTSA
3851 020202 116767 163467 000303          MOVB  STRTSA,REPA
3852
3853      : COMPUTE AND INSERT CRC
3854      :
3855 020210          CALL CRC IN <#REPM,#6> OUT <REPCRC>
3856 020210 162705 000002          SUB   #1*2,R5
3857 020214 010546          MOV   R5,-(SP)
3858 020216 012745 000006          MOV   #6,-(R5)
3859 020222 012745 020506          MOV   #REPM,-(R5)
3860 020226 004767 176660          JSR   PC,CRC
3861 020232 012605          MOV   (SP)+,R5
3862 020234 012567 000254          MOV   (R5)+,REPCRC
3863
3864      : SET UP BUFFER POINTER AND SIZE AND SEND THE MESSAGE
3865      :
3866 020240          LET XBUFFER := #REPM
3867 020240 012767 020506 175406          MOV   #REPM,XBUFFER
3868 020246          LET XSIZE := #10
3869 020246 012767 000010 175402          MOV   #10,XSIZE
3870 020254          CALL XMIT
3871 020254 004767 174364          JSR   PC,XMIT
3872
3873      : IF AN ERROR OCCURRED, RETURN WITH ERROR. OTHERWISE RETURN NORMALLY
3874      :
3875 020260          IF .ERROR THEN
3876 020260 103004          BCC   50002$
3877 020262          LET OUTERR := OUTERR + #1
3878 020262 005267 161642          INC   OUTERR
3879 020266          RETURN ERROR
3880 020266 000261          SEC
3881 020270 000401          BR   50001$
3882 020272          ENDIF
3883 020272          50002$:
3884 020272          ENDRTN
3885 020272          50000$:
3886 020272 000241          CLC
3887 020274          50001$:
3888 020274 000207          RTS   PC
3889
3890
3891
3892
3893
3894      : *****
3895      : ROUTINE TO PROCESS TERMINAL SELFTEST RESULTS
3896      : *****
3897
3898      : ROUTINE STINPT
3899 020276          STINPT:
3900 020276          PUSH R1
3901 020276
  
```

```

3902 020276 010146                                MOV    R1,-(SP)
3903 020300                                PUSH R2                                MOV    R2,-(SP)
3904 020300 010246                                PUSH R3                                MOV    R3,-(SP)
3905 020302 010346
3906 020302 010346
3907
3908      : SKIP RESPONSE HEADER AND CHECK FIRST DATA BYTE
3909      :
3910 020304                                LET R1 := INBUF + #4
3911 020304 016701 161646                        MOV    INBUF,R1
3912 020310 062701 000004                        ADD    #4,R1
3913
3914      : IF IT IS 'A', GOOD RESPONSE. OTHERWISE...
3915      :
3916 020314                                IFB (R1) NE #101 THEN
3917 020314 121127 000101                        CMPB   (R1),#101
3918 020320 001454                        BEQ    50002$
3919
3920      : GET THE BYTE AND CHECK TO SEE IF IT IS 'M' (MEMORY ERROR)
3921      :
3922 020322                                LET R3 :B= (R1)+
3923 020322 112103                                MOVB   (R1)+,R3
3924 020324                                IFB R3 EQ #115 THEN
3925 020324 120327 000115                        CMPB   R3,#115
3926 020330 001025                        BNE    50003$
3927
3928      : MEMORY ERROR...GET THE TYPE AND BANK NUMBER
3929      : INSERT INTO ERROR MESSAGE, REPORT THE ERROR AND
3930      : RETURN WITH ERROR
3931      : (NOTE: ERROR REPORTING PROTOCOL DELIMITERS ARE
3932      : IGNORED. THIS PROGRAM WILL NOT ACCEPT INPUT FROM
3933      : ANY OTHER INTELLIGENT TERMINAL PROPERLY.)
3934      :
3935 020332                                LET R1 := R1 + #2
3936 020332 062701 000002                        ADD    #2,R1
3937 020336                                LET R2 := #MTY
3938 020336 012702 003635                        MOV    #MTY,R2
3939 020342                                LET (R2)+ :B= (R1)+
3940 020342 112122                                MOVB   (R1)+,(R2)+
3941 020344                                LET (R2)+ :B= (R1)+
3942 020344 112122                                MOVB   (R1)+,(R2)+
3943 020346                                LET R2 := #BNO
3944 020346 012702 003651                        MOV    #BNO,R2
3945 020352                                LET R1 := R1 + #1
3946 020352 005201                                INC    R1
3947 020354                                LET R3 :B= (R1)
3948 020354 111103                                MOVB   (R1),R3
3949 020356                                LET R3 := R3 + #60
3950 020356 062703 000060                        ADD    #60,R3
3951 020362                                LET (R2) :B= R3
3952 020362 110312                                MOVB   R3,(R2)
3953 020364                                ERRHRD 71,MSG71
3954 020364 104423                                TRAP   T$ERCODE
3955 020366 000107                                .WORD 71
3956 020370 003611                                .WORD MSG71
3957 020372                                POP R3

```



```
3958 020372 012603                                MOV    (SP)+,R3
3959 020374                                POP R2
3960 020374 012602                                MOV    (SP)+,R2
3961 020376                                POP R1
3962 020376 012601                                MOV    (SP)+,R1
3963 020400                                RETURN ERROR
3964 020400 000261                                SEC
3965 020402 000427                                BR     50001$
3966 020404                                ELSE
3967 020404                                50003$:
3968
3969      :
3970      : NOT A MEMORY ERROR OR PASS...CHECK FOR 'P' (PROCESSOR ERROR)
3971      :
3971 020404                                IFB R3 EQ #120 THEN
3972 020404 120327 000120                                CMPB  R3,#120
3973 020410 001010                                BNE   50005$
3974
3975      :
3976      : PROCESSOR ERROR...REPORT IT AND EXIT WITH ERROR
3977      :
3977 020412                                ERRHRD 72,MSG72
3978 020412 104423                                TRAP  T$ERCODE
3979 020414 000110                                .WORD 72
3980 020416 003654                                .WORD MSG72
3981 020420                                POP R3
3982 020420 012603                                MOV    (SP)+,R3
3983 020422                                POP R2
3984 020422 012602                                MOV    (SP)+,R2
3985 020424                                POP R1
3986 020424 012601                                MOV    (SP)+,R1
3987 020426                                RETURN ERROR
3988 020426 000261                                SEC
3989 020430 000414                                BR     50001$
3990 020432                                ELSE
3991 020432                                50005$:
3992
3993      :
3994      : RESPONSE WAS NOT LEGAL...REPORT IT AND RETURN WITH ERROR
3995      :
3995 020432                                ERRHRD 70,MSG70
3996 020432 104423                                TRAP  T$ERCODE
3997 020434 000106                                .WORD 70
3998 020436 003556                                .WORD MSG70
3999 020440                                POP R3
4000 020440 012603                                MOV    (SP)+,R3
4001 020442                                POP R2
4002 020442 012602                                MOV    (SP)+,R2
4003 020444                                POP R1
4004 020444 012601                                MOV    (SP)+,R1
4005 020446                                RETJRN ERROR
4006 020446 000261                                SEC
4007 020450 000404                                BR     50001$
4008 020452                                ENDIF
4009 020452                                50006$:
4010 020452                                ENDIF
4011 020452                                50004$:
4012 020452                                ENDIF
4013 020452                                50002$:
```


4126	020640	012701	022014				
4127	020644	005301				MOV	#ELOOPB,R1
4128	020646			LET (R1) :B= #1		DEC	R1
4129	020646	112711	000001			MOVB	#1,(R1)
4130							
4131				:GET THE STATION ADDRESS AND INSERT IT IN THE HEADER			
4132							
4133	020652			LET R1 := #PTABLE + #5			
4134	020652	012701	002200			MOV	#PTABLE,R1
4135	020656	062701	000005			ADD	#5,R1
4136	020662			LET LOOSTA :B= (R1)			
4137	020662	111167	000523			MOVB	(R1),LOOSTA
4138	020666			POP R1			
4139	020666	012601				MOV	(SP)+,R1
4140							
4141				:INSERT LOOP DATA TYPE			
4142							
4143	020670			LET LOOPB :B= #OSOP			
4144	020670	112767	000030 000516			MOVB	#OSOP,LOOPB
4145							
4146				:SET MESSAGE NUMBER			
4147							
4148	020676			LET LOOPN :B= #1			
4149	020676	112767	000001 000504			MOVB	#1,LOOPN
4150							
4151				:COMPUTE AND INSERT HEADER CRC			
4152							
4153	020704			CALL CRC IN <#LOOPH,#6> OUT <LCRCH>			
4154	020704	162705	000002			SUB	#1*2,R5
4155	020710	010546				MOV	R5,-(SP)
4156	020712	012745	000006			MOV	#6,-(R5)
4157	020716	012745	021404			MOV	#LOOPH,-(R5)
4158	020722	004767	176164			JSR	PC,CRC
4159	020726	012605				MOV	(SP)+,R5
4160	020730	012567	000456			MOV	(R5)+,LCRCH
4161							
4162				:COMPUTE AND INSERT DATA CRC			
4163							
4164	020734			CALL CRC IN <#LOOPB,#256.> OUT <LCRCD>			
4165	020734	162705	000002			SUB	#1*2,R5
4166	020740	010546				MOV	R5,-(SP)
4167	020742	012745	000400			MOV	#256,-(R5)
4168	020746	012745	021414			MOV	#LOOPB,-(R5)
4169	020752	004767	176134			JSR	PC,CRC
4170	020756	012605				MOV	(SP)+,R5
4171	020760	012567	001030			MOV	(R5)+,LCRCD
4172							
4173				:SET UP BUFFER POINTER AND SIZE....SEND MESSAGE			
4174							
4175	020764			LET XBUFFER := #LOOPH			
4176	020764	012767	021404 174662			MOV	#LOOPH,XBUFFER
4177	020772			LET XSIZE := #266.			
4178	020772	012767	000412 174656			MOV	#266.,XSIZE
4179	021000			CALL XMIT			
4180	021000	004767	173640			JSR	PC,XMIT
4181							

```

4182      ;IF ERROR OCCURRED...INSERT THE ERROR INTO TABLE (MANUF. TESTS ONLY),
4183      ;REPORT THE ERROR, LOOP IF ENABLED
4184      :
4185      IF.ERROR THEN
4186      021004      103012      BCC      50013$
4187      021006      CALL INSERR IN <#0>
4188      021006      010546      MOV      R5,-(SP)
4189      021010      012745      000000      MOV      #0,-(R5)
4190      021014      004767      176070      JSR      PC,INSERR
4191      021020      012605      MOV      (SP)+,R5
4192      021022      ERRHRD 51,MSG51,AREA40
4193      021022      104463      TRAP    T$ERCODE
4194      021024      000063      .WORD  51
4195      021026      003515      .WORD  MSG51
4196      021030      004262      .WORD  AREA40
4197      021032      ENDIF
4198      021032      50013$:
4199      021032      CKLOOP
4200      021032      104006      EMT     C$CLP1
4201      :
4202      ;SEND ACK'S TO TERMINAL UNTIL IT RESPONDS
4203      :
4204      021034      REPEAT
4205      021034      50014$:
4206      :
4207      ;GET RESPONSE
4208      :
4209      021034      CALL TERIN OUT <LCOMP>
4210      021034      162705      000002      JSR      #1*2,R5
4211      021040      004767      172576      JSR      PC,TERIN
4212      021044      012567      000326      MOV      (R5)+,LCOMP
4213      :
4214      ;IF ERROR OCCURS...INSERT IN TABLE (MAN. ONLY), REPORT IT, LOOP IF
4215      ;ENABLED, EXIT TEST IF NOT
4216      :
4217      IF.ERROR THEN
4218      021050      103012      BCC      50015$
4219      021052      CALL INSERR IN <#0>
4220      021052      010546      MOV      R5,-(SP)
4221      021054      012745      000000      MOV      #0,-(R5)
4222      021060      004767      176024      JSR      PC,INSERR
4223      021064      012605      MOV      (SP)+,R5
4224      021066      ERRHRD 51,MSG51,AREA40
4225      021066      104463      TRAP    T$ERCODE
4226      021070      000063      .WORD  51
4227      021072      003515      .WORD  MSG51
4228      021074      004262      .WORD  AREA40
4229      021076      ENDIF
4230      021076      50015$:
4231      021076      CKLOOP
4232      021076      104006      EMT     C$CLP1
4233      021100      IF LCOMP EQ #0 THEN
4234      021100      005767      000272      TST     LCOMP
4235      021104      001026      BNE     50016$
4236      021106      CALL C$TREP
4237      021106      004767      177054      JSR     PC,OUTREP

```

```

4238
4239
4240
4241
4242 021112
4243 021112 103023
4244 021114
4245 021114 010546
4246 021116 012745 000000
4247 021122 004767 175762
4248 021126 012605
4249 021130
4250 021130 010546
4251 021132 012745 000002
4252 021136 004767 175746
4253 021142 012605
4254 021144
4255 021144 104463
4256 021146 000063
4257 021150 003515
4258 021152 004262
4259 021154
4260 021154 104006
4261 021156
4262 021156 104032
4263 021160 000074
4264 021162
4265 021162
4266 021162
4267 021162
4268
4269
4270
4271 021162
4272 021162 005767 000210
4273 021166 001722
4274
4275
4276
4277 021170
4278 021170 004767 000062
4279 021174
4280 021174 104004
4281
4282
4283
4284 021176
4285 021176 005067 174446
4286 021202
4287 021202 004767 171606
4288
4289
4290
4291
4292 021206
4293 021206 103020

```

```

: IF AN ERROR OCCURS...INSERT IN TABLE (MAN. ONLY), REPORT IT, LOOP
: IF ENABLED - EXIT TEST IF NOT
:
      IF.ERROR THEN
          CALL INSERR IN <#0>
          MOV R5,-(SP)
          MOV #0,-(R5)
          JSR PC,INSERR
          MOV (SP)+,R5
          CALL INSERR IN <#2>
          MOV R5,-(SP)
          MOV #2,-(R5)
          JSR PC,INSERR
          MOV (SP)+,R5
          FRRHRD 51,MSG51,AREA40
          TRAP T$ERCODE
          .WORD 51
          .WORD MSG51
          .WORD AREA40
          CKLOOP
          EXIT TST
          EMT C$CLP1
          EMT C$EXIT
          .WORD L10032-
      ENDIF
50017$:
50016$:
: REPEAT LOOP UNTIL TERMINAL SENDS LOOPBACK DATA
:
      UNTIL LCOMP NE #C
          TST LCOMP
          BEQ 50014$
: PROCESS THE LOOPBACK DATA
:
      CALL PRGLOP
          JSR PC,PROLOP
          BGNSEG
          EMT C$BSEG
: REINIT THE LINK
:
      LET INIT :- #0
          CLR INIT
          CALL XSTRT
          JSR PC,XSTRT
: IF ERROR OCCURS, INSERT ERROR INTO TABLE (MANUF. ONLY), REPORT ERROR,
: LOOP ON REINIT IF ERROR LOOPING ENABLED...EXIT TEST IF NO LOOPING
:
      IF.ERROR THEN
          BCC 50020$

```

```

4294 021210          CALL INSERR IN <#0>
4295 021210 010546
4296 021212 012745 000000
4297 021216 004767 175666
4298 021222 012605
4299 021224          CALL INSERR IN <#2>
4300 021224 010546
4301 021226 012745 000002
4302 021232 004767 175652
4303 021236 012605
4304 021240          ERRHRD 51,MSG51,AREA61
4305 021240 104463
4306 021242 000063
4307 021244 003515
4308 021246 005052
4309 021250          ENDIF
4310 021250          CKLOOP
4311 021250
4312 021250 104006
4313 021252          ENDSEGE
4314 021252
4315 021252 104005
4316
4317          :END OF TEST
4318          :
4319 021254          ENDTST
4320 021254
4321 021254 104001
4322
4323
4324
4325
4326
4327          :
4328          :*****
4329          :ROUTINE TO PROCESS LOOPBACK DATA
4330          :*****
4331          :
4332 021256          ROUTINE PROLOP
4333 021256
4334 021256          PUSH R1
4335 021256 010146          MOV R1,-(SP)
4336 021260          PUSH R2
4337 021260 010246          MOV R2,-(SP)
4338 021262          PUSH R3
4339 021262 010346          MOV R3,-(SP)
4340
4341          :R1 = POINTER TO EXPECTED DATA
4342          :R2 = COUNT OF DATA EXPECTED
4343          :R3 = POINTER TO DATA FROM TERMINAL
4344          :
4345 021264          LET R1 := #LOOPB
4346 021264 012701 021414          MOV #LOOPB,R1
4347 021270          LET R2 := #128.
4348 021270 012702 000200          MOV #128.,R2
4349 021274          LET R3 := INBUF
  
```

```

4350 021274 016703 160656                                MOV     INBUF,R3
4351
4352                :CHECK DATA BYTE-BY-BYTE...
4353                :
4354 021300                REPEAT
4355 021300                                50002$:
4356                :
4357                :IF ERROR...INSERT ERROR INTO TABLE (MANUF. ONLY), REPORT ERROR
4358                :(INCLUDING RECEIVED/EXPECTED BYTES), RETURN WITH ERROR
4359                :
4360 021300                IFB (R1) NE (R3) THEN
4361 021300 121113                                CMPB   (R1),(R3)
4362 021302 001423                                BEQ    50003$
4363 021304                LET A52A := (R1)
4364 021304 011167 163770                                MOV    (R1),A52A
4365 021310                LET A52B := (R3)
4366 021310 011367 163766                                MOV    (R3),A52B
4367 021314                CALL INSERR IN <#2>
4368 021314 010546                                MOV    R5,-(SP)
4369 021316 012745 000002                                MOV    #2,-(R5)
4370 021322 004767 175562                                JSR    PC,INSERR
4371 021326 012605                                MOV    (SP)+,R5
4372 021330                ERRSOF 52,MSG52,AREA52
4373 021330 104464                                TRAP   T$ERCODE
4374 021332 000064                                .WORD 52
4375 021334 003543                                .WORD MSG52
4376 021336 005202                                .WORD AREA52
4377 021340                POP R3
4378 021340 012603                                MOV    (SP)+,R3
4379 021342                POP R2
4380 021342 012602                                MOV    (SP)+,R2
4381 021344                POP R1
4382 021344 012601                                MOV    (SP)+,R1
4383 021346                RETURN ERROR
4384 021346 000261                                SEC
4385 021350 000411                                BR    50001$
4386 021352                ENDF
4387 021352                                50003$:
4388                :
4389                :UPDATE COUNT AND POINTERS
4390                :
4391 021352                LET R1 := R1 + #1
4392 021352 005201                                INC    R1
4393 021354                LET R2 := R2 - #1
4394 021354 005302                                DEC    R2
4395 021356                LET R3 := R3 + #1
4396 021356 005203                                INC    R3
4397                :
4398                :REPEAT UNTIL ALL DATA HAS BEEN CHECKED
4399                :
4400 021360                UNTIL R2 EQ #0
4401 021360 005702                                TST   R2
4402 021362 001346                                BNE   50002$
4403 021364                POP R3
4404 021364 012603                                MOV    (SP)+,R3
4405 021366                POP R2

```



```
4406 021366 012602
4407 021370          POP R1
4408 021370 012601
4409 021372          ENDRTN
4410 021372
4411 021372 000241
4412 021374
4413 021374 000207
4414
4415
4416          : LOOPBACK MESSAGE AREA
4417          :
4418 021376 000000
4419 021400 121405
4420 021402 000410
4421 021404 220      000      201 LOOPH: .BYTE DLE,0,201,0
4422 021407 000
4423 021410 000
4424 021411 000
4425 021412 000000
4426 021414 000200
4427 022014
4428 022014 000000
4429
4430 022016          ENDMOD
```

50000\$: CLC
50001\$: RTS PC

MOV (SP)+,R2
MOV (SP)+,R1

```
4431 022016          BGNMOD PTCODE
4432                :
4433                :
4434                :
4435 022016          BGNHRD
4436 022016 000044          .WORD L10033-L$HARD/2
4437 022020          GPRMA  DEVICE,0,0,0,177776,YES
4438 022020 000031          .WORD  T$CODE
4439 022022 022130          .WORD  DEVICE
4440 022024 000000          .WORD  T$LLOLIM
4441 022026 177776          .WORD  T$HILIM
4442 022030          GPRMA  VECTOR,2,0,0,400,YES
4443 022030 001031          .WORD  T$CODE
4444 022032 022145          .WORD  VECTOR
4445 022034 000000          .WORD  T$LLOLIM
4446 022036 000400          .WORD  T$HILIM
4447 022040          GPRMD  PRIOR,4,0,340,0,7,YES
4448 022040 002032          .WORD  T$CODE
4449 022042 022161          .WORD  PRIOR
4450 022044 000340          .WORD  340
4451 022046 000000          .WORD  T$LLOLIM
4452 022050 000007          .WORD  T$HILIM
4453 022052          GPRMD  STATION,4,0,177400,0,377,YES
4454 022052 002032          .WORD  T$CODE
4455 022054 022203          .WORD  STATION
4456 022056 177400          .WORD  177400
4457 022060 000000          .WORD  T$LLOLIM
4458 022062 000377          .WORD  T$HILIM
4459 022064          GPRMD  LINE,6,0,7,0,7,YES
4460 022064 003032          .WORD  T$CODE
4461 022066 022226          .WORD  LINE
4462 022070 000007          .WORD  7
4463 022072 000000          .WORD  T$LLOLIM
4464 022074 000007          .WORD  T$HILIM
4465 022076          GPRMD  BAUD,6,0,7400,0,17,YES
4466 022076 003032          .WORD  T$CODE
4467 022100 022244          .WORD  BAUD
4468 022102 007400          .WORD  7400
4469 022104 000000          .WORD  T$LLOLIM
4470 022106 000017          .WORD  T$HILIM
4471 022110          GPRMD  FLAGZ,6,0,70,0,7,YES
4472 022110 003032          .WORD  T$CODE
4473 022112 022260          .WORD  FLAGZ
4474 022114 000070          .WORD  70
4475 022116 000000          .WORD  T$LLOLIM
4476 022120 000007          .WORD  T$HILIM
4477 022122          GPRML  NMODEM,10,177777,YES
4478 022122 004130          .WORD  T$CODE
4479 022124 022324          .WORD  NMODEM
4480 022126 177777          .WORD  177777
4481 022130          ENDHRD
4482
4483 022130          .EVEN
4484
4485 022130 055104 040440 042104  DEVICE: .ASCIZ /DZ ADDRESS: /
4486 022136 042522 051523 020072
```

L10033:

```

4487 022144 000
4488 022145 104 020132 042526 VECTOR: .ASCIZ /DZ VECTOR: /
4489 022152 052103 051117 020072
4490 022160 000
4491 022161 104 020132 052502 PRIOR: .ASCIZ /DZ BUS PRIORITY: /
4492 022166 020123 051120 047511
4493 022174 044522 054524 020072
4494 022202 000
4495 022203 124 051105 044515 STATION:.ASCIZ /TERMINAL ADDRESS: /
4496 022210 040516 020114 042101
4497 022216 051104 051505 035123
4498 022224 000040
4499 022226 044514 042516 047040 LINE: .ASCIZ /LINE NUMBER: /
4500 022234 046525 042502 035122
4501 022242 000040
4502 022244 040502 042125 051040 BAUD: .ASCIZ 7BAUD RATE: /
4503 022252 052101 035105 000040

```

```

:BAUD RATE SELECTION:
:      TYPE      TO GET  BAUD RATE

```

```

:BAUD RATE SELECTION:
:      TYPE      TO GET  THIS RATE
:0
:1
:2
:3
:4
:5
:6
:7
:10
:11
:12
:13
:14
:15
:16
:17

```

```

4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527 022260 052123 050117 041440 FLAGZ: .ASCIZ @STOP CODE/CHAR. LENGTH (SEE DOC.): @
4528 022266 042117 027505 044103
4529 022274 051101 020056 042514
4530 022302 043516 044124 024040
4531 022310 042523 020105 047504
4532 022316 027103 035051 000040

```

```

:ALLOWABLE RESPONSES:
:0 = 1 STOP BIT - 5 BIT CHAR.
:1 = .. - 6 ..
:2 = .. - 7 ..
:3 = .. - 8 ..
:4 = 2 STOP BIT - 5 ..
:5 = .. - 6 ..
:6 = .. - 7 ..
:7 = .. - 8 ..

```

```

4533
4534
4535
4536
4537
4538
4539
4540
4541
4542

```

```

4543
4544 022324 051525 047111 020107 ;MODEM: .ASCIZ /USING A NULL MODEM? /
4545 022332 020101 052516 046114
4546 022340 046440 042117 046505
4547 022346 020077 000
4548 022352 .EVEN
4549
4550
4551
4552
4553 022352 BGNSFT
4554 022352 000015 .WORD L10034-L$SOFT/2
4555 022354 GPRMD ALIMIT,0,0,177777,0,77777,YES
4556 022354 000032 .WORD T$CODE
4557 022356 022406 .WORD ALIMIT
4558 022360 177777 .WORD 177777
4559 022362 000000 .WORD T$LOLIM
4560 022364 077777 .WORD T$HILIM
4561 022366 GPRML PATTRN,2,177777,YES
4562 022366 001130 .WORD T$CODE
4563 022370 022455 .WORD PATTRN
4564 022372 177777 .WORD 177777
4565 022374 GPRMD SWAIT,4,0,177777,0,77777,YES
4566 022374 002032 .WORD T$CODE
4567 022376 022520 .WORD SWAIT
4568 022400 177777 .WORD 177777
4569 022402 000000 .WORD T$LOLIM
4570 022404 077777 .WORD T$HILIM
4571 022406 ENDSFT
4572 .EVEN
4573 022406 L10034:
4574 022406 047510 020127 040515 ALIMIT: .ASCIZ /HOW MANY INITIATE TRIES BEFORE ABORT? /
4575 022414 054516 044440 044516
4576 022422 044524 052101 020105
4577 022430 051124 042511 020123
4578 022436 042502 047506 042522
4579 022444 040440 047502 052122
4580 022452 020077 000
4581 022455 125 042523 040440 PATTRN: .ASCIZ /USE ALL ZERO'S FOR LOOPBACK DATA? /
4582 022462 046114 055040 051105
4583 022470 023517 020123 047506
4584 022476 020122 047514 050117
4585 022504 040502 045503 042040
4586 022512 052101 037501 000040
4587 022520 040527 052111 043040 SWAIT: .ASCIZ @WAIT FACTOR FOR SELFTST (20000 = 11/70 - 4000 - 11/10)?@
4588 022526 041501 047524 020122
4589 022534 047506 020122 042523
4590 022542 043114 042524 052123
4591 022550 024040 030062 030060
4592 022556 020060 020075 030461
4593 022564 033457 020060 020055
4594 022572 030064 030060 036440
4595 022600 030440 027461 030061
4596 022606 037451 000
4597 022612 .EVEN
4598 022612 ENDMOD
  
```

4599 022612
4600
4601 022612
4602

LASTAD

.EVEN

L\$LAST::
.REQ DOCTOR.P11[400,1066]

4603
 4604 053406 000000
 053410 000000
 053412 000000
 053414 000000
 053420
 000200

.SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
 .WORD 0 ;SPACE FOR USER POOL POINTER
 .WORD 0 ;SIZE
 .WORD 0 ;CHECKSUM (NOT CURRENTLY USED)
 .WORD 0 ;SIZE OF H.W. PTAB. ALLOCATION
 END.SUPV-.+2
 .END 200

L\$CO	002032	G	699#		
L\$DEPO	002011	G	690#		
L\$DESC	002102	G	720#		
L\$DEVP	002064	G	713#		
L\$DISP	011716	G	702	1320#	
L\$DR	002250	G	715	816#	
L\$DRCT	002070	G	715#		
L\$DRS	002072	G	716#		
L\$DRST	002254	G	716	818#	
L\$DTP	002040	G	702#		
L\$DU	011726	G	718	1339#	
L\$DUT	002076	G	718#		
L\$DVTY	002234	G	713	811#	
L\$EF	002056	G	710#		
L\$EFLG	002034	G	700#		
L\$EXP1	002042	G	703#		
L\$EXP2	002044	G	704#		
L\$EXP3	002046	G	705#		
L\$HARD	022020	G	693	4436	4437#
L\$HPCP	002016	G	693#		
L\$HPTP	002022	G	695#		
L\$HW	011672	G	695	1299	1300#
L\$IICP	002104	G	721#		
L\$INIT	011730	G	721	1347#	
L\$LADP	002026	G	697#		
L\$LAST	022612	G	697	4601#	
L\$MREV	002050	G	706#		
L\$NAME	002000	G	681#		
L\$REPP	002066	G	714#		
L\$REV	002010	G	689#		
L\$RPT	011722	G	714	1329#	
L\$SOFT	022354	G	694	4554	4555#
L\$SPC	002062	G	712#		
L\$SPCP	002020	G	694#		
L\$SPTP	002024	G	696#		
L\$STA	002030	G	698#		
L\$SW	011706	G	696	1308	1309#
L\$TIML	002014	G	692#		
L\$TIMU	002054	G	709#		
L\$TIM1	002052	G	708#		
L\$TSTI	002100	G	719#		
L\$UNIT	002012	G	691#	1388	1459
L.CLK.	027210		4604#		
L10000	003724		992#		
L10001	003760		1003#		
L10002	004012		1016#		
L10003	004114		1039#		
L10004	004146		1054#		
L10005	004250		1076#		
L10006	004256		1082#		
L10007	004260		1086#		
L10010	004352		1116#		
L10011	004500		1141#		
L10012	004664		1187#		
L10013	005014		1212#		
L10014	005142		1246#		

PATRN	022455	4563	4581#													
PRINTC	043416	4604#														
PRINTF	046724	4604#														
PRIOR	022161	4449	4491#													
PRNST	043306	4604#														
PROBUF	013350	1733	1981#													
PROLOP	021256	4278	4333#													
PRO.CM	027556	4604#*														
PTABLE	002200	793#	1477	1493	1533	1806	2031	2413	2528	2619	2694	2758	2773	2780		
		4134														
PTAB.S	023076	4604#*														
PTCODE	022016	4432#														
PTEND	002212	798#	1487													
PUTCHR	042122	4604#														
PWR.FA	053246	4604#														
PWR.FL	022702	4604#*														
PWR.MS	053374	4604#														
PWR.SA	053370	4604#*														
PWR.UP	053372	4604#*														
P.CLK.	027216	4604#														
RBQUE	015710	1426*	1428	2872#	3036	3044*	3046	3049*	3089	3097*	3099	3102*				
RBQUEB	015716	1438	2875#	3046	3099											
RBQUET	015712	1426	2873#	3049	3102											
RCVADD	015640	2808*	2809*	2849#	2995	3372										
RCVBUF=	000002	746#	1164	2534*	2625*	2731*	2767*	3381*								
RCVCSR=	000000	745#	1157	1537*	2532*	2623*	2699*	2703	2729*	2824*	2826*	2828*	2946*	3379*		
RCVE	016630	1845	2125	3275#												
RDONE	015156	1134	2504*	2660#	3003*	3235*	3297	3318	3342*	3355*						
READ.P	050306	4604#														
REGBAC	052776	4604#														
REGSAV	052762	4604#														
RENABL=	010000	756#														
REP =	000003	736#														
REPA	020513	3851*	4048#													
REPCRC	020514	3862*	4049#													
REPM	020506	3859	3867	4045#												
REPN	020511	3849*	4046#													
REQN.P	022666	4604#*														
REQN.T	027560	4604#*														
REXIT	016626	3008	3256#													
RE.SET	024350	4604#														
RINT	016066	2784	2980#													
RMSGNO	002176	792#	1688*	2019	2175*	2302*	3849									
RQUE	015672	1422*	1424	2865#	3038*	3055*	3057*	3059*	3061*	3064*	3091*	3106*	3192*	3194*		
		3205*	3207	3210*												
RQUEB	015706	2871#	3061	3207	3332											
RQUET	015674	1422	2866#	3064	3210	3335										
RQUEUE	015670	1424*	2864#	3324	3326*	3328	3330*	3332	3335*							
RSTACK	052236	4604#														
RTEMP1	015662	2858#	3036*	3038	3040*	3042*	3089*	3091	3093*	3095*	3140*	3153	3157*	3176		
		3178*	3226*													
RTEMP2	015664	2859#	3053*	3153*	3155*	3184*	3192	3200*	3228*	3232						
RXINTE=	000100	753#	2828													
R5STCK	002110	763#	1369*													
SAVEDO-	024546	4604#														
SCRATCH	002112	764#	1031	1430*	1434	1436*	1450*	1481	1485*	1588*	1589	1591*	1593	1644*		

		1645	1647*	1649	1697*	1698*	1700	1929*	1930*	1941	2039*	2040*	2050	2063*
		2073	2366*	2368*	2372*	2377	2413*	2414*	2416*	2418*	2431	2453	2709*	2711*
		2713	2724	3678*	3685*	3687	4107*	4112*	4118					
SEARCH	044244	4604#												
SEGSTA	023112 G	4604#*												
SET.MA	027770	4604#												
SHIFT	053074 G	4604#												
SOH =	000201	733#	1996	3079										
SPEC.U	027504	4604#*												
SPV.SE	000400	4604#												
SSTACK	011670	1291#	1367											
STACK =	000007	739#	1864											
STARTC	051764 G	4604#												
STATIO	022203	4455	4495#											
STIME	002122	768#												
STINPT	020276	3800	3900#											
STRCHR	042776	4604#												
STRT =	000006	738#	973											
STRTB	003670	972#	1801											
STRTC	003676	978#	1821*											
STRISA	003675	977#	1809*	2231	3851									
STRT.T	027562	4604#*												
STWAIT	011712	1311#	3687											
ST.SET	024414	4604#												
SUNIT.	027566	4604#*												
SUPERV	025450	4604#												
SUPFLA	023056 G	4604#*												
SUPV.T	023230 G	4604#*												
SUP.PR	024166	4604#												
SVCGBL=	000000	673#	680	681	689	690	691	692	693	694	695	696	697	698
		699	700	701	702	703	704	705	706	708	709	710	712	713
		714	715	716	717	718	719	720	721	722	725	811	816	818
		984	1002	1007	1029	1043	1067	1081	1085	1089	1133	1151	1205	1221
		1255	1277	1282	1295	1300	1309	1320	1324	1329	1334	1339	1347	1527
		1553	2880	2980	3567	4432	4437	4555	4601#	4602				
SVCHAN	032104	4604#												
SVCINS	000001	573#	674#	681	682	683	684	685	686	687	688	689	690	691
		692	693	694	695	696	697	698	699	700	701	702	703	704
		705	706	707	708	709	710	711	712	713	714	715	716	717
		718	719	720	721	722	811	813	815	816	817	818	985	986
		987	988	989	990	993	1004	1008	1009	1010	1011	1012	1013	1014
		1017	1031	1032	1033	1034	1035	1036	1037	1040	1046	1047	1048	1049
		1050	1051	1052	1055	1068	1069	1070	1071	1072	1073	1074	1077	1083
		1087	1103	1104	1105	1106	1107	1108	1110	1111	1112	1113	1114	1117
		1134	1135	1136	1137	1138	1139	1142	1157	1158	1159	1160	1161	1162
		1164	1165	1166	1167	1168	1169	1171	1172	1173	1174	1175	1176	1178
		1179	1180	1181	1182	1183	1188	1206	1207	1208	1209	1210	1213	1222
		1223	1224	1225	1226	1239	1240	1241	1242	1243	1244	1247	1256	1257
		1258	1259	1260	1261	1262	1265	1279	1284	1299	1308	1319	1320	1321
		1331	1336	1341	1356	1357	1381	1382	1402	1403	1448	1449	1450	1510
		1550	1745	1746	1747	1833	1834	1835	1852	1853	1854	1876	1877	1878
		2132	2133	2134	2135	2214	2215	2216	2217	2234	2235	2236	2237	2282
		2283	2284	2285	2342	2343	2344	2345	2400	2401	2402	2403	2434	2435
		2436	2437	2457	2458	2459	2460	2547	2548	2549	2550	2570	2571	2572
		2573	2600	2635	2636	2637	2638	2721	2722	2741	2742	2743	2744	2783
		2784	2785	2786	2787	2788	2795	2796	2797	2798	2799	2800	2884	2885

TYPLIN 042332	4604#																	
TYPNUM 041714	4604#																	
TYPSTR 042352	4604#																	
TYP.ER 036160	4604#																	
TY.UNI 031172	4604#																	
TSARGC= 000003	681#	682#	683#	684#	685#	686#	985#	990	1008#	1014	1031#	1037	1046#					
	1052	1068#	1074	1103#	1108	1110#	1114	1134#	1139	1157#	1162	1164#	1169					
	1171#	1176	1178#	1183	1206#	1210	1222#	1226	1239#	1244	1256#	1262						
TSCODE= 002032	4438#	4443#	4448#	4454#	4460#	4466#	4472#	4478#	4556#	4562#	4566#							
TSERCO= 000064	1745#	1833#	1852#	1876#	2132#	2214#	2234#	2282#	2342#	2400#	2434#	2457#	2547#					
	2570#	2635#	2741#	3350#	3394#	3611#	3642#	3661#	3711#	3745#	3777#	3813#	3954#					
	3978#	3996#	4085#	4193#	4225#	4255#	4305#	4373#										
TSERRN= 000064	673#	1746#	1834#	1853#	1877#	2133#	2215#	2235#	2283#	2343#	2401#	2435#	2458#					
	2548#	2571#	2636#	2742#	3351#	3395#	3612#	3643#	3662#	3712#	3746#	3778#	3814#					
	3955#	3979#	3997#	4086#	4194#	4226#	4256#	4306#	4374#									
TSFXCP= 000000	4438#	4442	4443#	4447	4448#	4453	4454#	4459	4460#	4465	4466#	4471	4472#					
	4477	4556#	4561	4566#	4571													
TSFLAG= 000040	3618#	3649#	3668#	3718#	3752#	3784#	3820#	4092#	4262#									
TSHILI= 077777	4438#	4441	4443#	4446	4448#	4452	4454#	4458	4460#	4464	4466#	4470	4472#					
	4476	4556#	4560	4566#	4570													
TSLOLI= 000000	4438#	4440	4443#	4445	4448#	4451	4454#	4457	4460#	4463	4466#	4469	4472#					
	4475	4556#	4559	4566#	4569													
TSLSYM= 010000	673#	993	1004	1017	1040	1055	1077	1083	1087	1117	1142	1188	1213					
	1247	1265	1279	1284	1307	1314	1331	1336	1341	1510	1550	2970	3258					
	3829	4321	4484	4574														
TSNEST= 177777	673#	680#	724#	725#	984#	992#	1002#	1003#	1007#	1016#	1029#	1039#	1043#					
	1054#	1067#	1076#	1081#	1082#	1085#	1086#	1089#	1116#	1133#	1141#	1151#	1187#					
	1205#	1212#	1221#	1246#	1255#	1264#	1277#	1278#	1282#	1283#	1294#	1295#	1299#					
	1306#	1308#	1313#	1323#	1324#	1329#	1330#	1334#	1335#	1339#	1340#	1347#	1509#					
	1527#	1549#	1552#	1553#	2880#	2969#	2980#	3257#	3566#	3567#	3577#	3828#	4057#					
	4280#	4314#	4320#	4431#	4432#	4436#	4482#	4554#	4572#	4599#								
TSNSKO= 000000	680#	724	725#	1294	1295#	1323	1324#	1552	1553#	3566	3567#	4431	4432#					
	4599																	
TSNSK1= 000005	984#	992	1002#	1003	1007#	1016	1029#	1039	1043#	1054	1067#	1076	1081#					
	1082	1085#	1086	1089#	1116	1133#	1141	1151#	1187	1205#	1212	1221#	1246					
	1255#	1264	1277#	1278	1282#	1283	1299#	1306	1308#	1313	1329#	1330	1334#					
	1335	1339#	1340	1347#	1509	1527#	1549	2880#	2969	2980#	3257	3577#	3828					
	4057#	4320	4436#	4482	4554#	4572												
TSNSK2= 000003	4280#	4314																
TSSAVL= 177777	673#																	
TSSEGL= 177777	673#	4280#	4314#	4316														
TSSEKO= 010000	4280#	4314																
TSSEUN= 000000	673#	3577#	4057#															
TSSTAGL= 177777	673#																	
TSSTAGN= 010035	673#	984#	1002#	1007#	1029#	1043#	1067#	1081#	1085#	1089#	1133#	1151#	1205#					
	1221#	1255#	1277#	1282#	1299#	1308#	1329#	1334#	1339#	1347#	1527#	2880#	2980#					
	3577#	4057#	4436#	4554#														
TSSTEMP 000000	724#	992#	1003#	1016#	1039#	1054#	1076#	1082#	1086#	1116#	1141#	1187#	1212#					
	1246#	1264#	1278#	1283#	1294#	1306#	1313#	1320#	1321#	1322#	1323#	1330#	1335#					
	1340#	1509#	1549#	1552#	2969#	3257#	3566#	3618#	3619	3649#	3650	3668#	3669					
	3718#	3719	3752#	3753	3784#	3785	3820#	3821	3828#	4092#	4093	4262#	4263					
	4314#	4320#	4431#	4438#	4443#	4448#	4454#	4460#	4466#	4472#	4478#	4482#	4556#					
	4562#	4566#	4572#	4599#														
TSSTEST= 000002	673#	3577#	4057#															
TSSTSTM= 177777	673#	989	993	1004	1013	1017	1036	1040	1051	1055	1073	1077	1083					
	1087	1107	1113	1117	1138	1142	1161	1168	1175	1182	1188	1209	1213					

	3487	3488#	3489	3490#	3491	3492#	3495	3496#	3497	3498#	3499	3500#	3501
	3502#	3506	3507#	3508	3509#	3510	3511#	3515	3519#	3520	3581#	3582	3586#
	3587#	3588	3589#	3590	3594#	3606#	3626#	3637#	3676#	3677	3678#	3679	3685#
	3686	3693#	3694#	3695	3696#	3697	3698#	3706#	3731#	3740#	3760#	3767#	3772#
	3800#	3808#	3847#	3848	3849#	3850	3851#	3852	3856#	3867#	3868	3869#	3870
	3871#	3878#	3879	3880#	3882#	3911#	3912#	3913	3923#	3924	3936#	3937	3938#
	3939	3940#	3941	3942#	3943	3944#	3945	3946#	3947	3948#	3949	3950#	3951
	3952#	3953	3964#	3966#	3988#	3990#	4006#	4008#	4061#	4062	4063#	4064	4065#
	4066	4070#	4071	4072#	4080#	4102#	4103	4107#	4108	4112#	4113	4118#	4119
	4126#	4127#	4128	4129#	4130	4134#	4135#	4136	4137#	4138	4144#	4145	4149#
	4150	4154#	4165#	4176#	4177	4178#	4179	4180#	4188#	4210#	4220#	4237#	4245#
	4250#	4278#	4285#	4286	4287#	4295#	4300#	4346#	4347	4348#	4349	4350#	4351
	4364#	4365	4366#	4367	4368#	4384#	4386#	4392#	4393	4394#	4395	4396#	4397
SF SAND- 000310	674#	1096	1232	1390	1461	1595	1608	1651	1676	1723	1943	2153	2213
	2233	2281	2341	2379	2395	2433	2569	2594	2610	2720	2726	2892	2901
	2923	3002	3016	3023	3031	3048	3063	3081	3101	3147	3169	3209	3234
	3311	3320	3334	3468	3660	3759	3919	3927	3974	4106	4236	4363	
SF SBAD= 000401	674#	1092	1096	1097	1101	1155	1228	1232	1233	1237	1367	1369	1386
	1390	1391	1407	1415	1417	1422	1424	1426	1428	1430	1434	1436	1446
	1457	1461	1462	1466	1477	1481	1483	1485	1493	1496	1498	1500	1502
	1533	1535	1537	1544	1582	1588	1595	1596	1598	1608	1609	1611	1619
	1627	1629	1631	1636	1638	1644	1651	1652	1676	1686	1688	1695	1697
	1700	1703	1705	1725	1799	1801	1806	1809	1864	1918	1924	1929	1935
	1945	1949	1951	1953	1955	1957	1990	1996	2001	2003	2005	2007	2012
	2014	2019	2024	2026	2031	2034	2039	2042	2058	2063	2065	2081	2084
	2086	2088	2093	2096	2153	2154	2156	2172	2175	2198	2201	2203	2205
	2213	2233	2249	2251	2264	2267	2269	2271	2273	2281	2297	2299	2302
	2341	2363	2366	2368	2370	2372	2379	2383	2386	2388	2395	2396	2398
	2413	2416	2418	2420	2423	2425	2433	2453	2455	2500	2502	2504	2528
	2530	2532	2534	2536	2538	2552	2560	2569	2575	2577	2594	2610	2619
	2621	2623	2625	2627	2629	2694	2699	2709	2711	2713	2720	2726	2727
	2729	2731	2733	2735	2758	2761	2763	2765	2767	2773	2776	2778	2780
	2790	2792	2805	2808	2814	2816	2822	2824	2826	2828	2837	2892	2901
	2905	2907	2909	2911	2913	2923	2927	2929	2937	2940	2944	2946	2960
	2962	2995	3002	3003	3005	3012	3016	3025	3031	3036	3038	3040	3042
	3044	3048	3049	3053	3055	3057	3059	3063	3064	3071	3084	3089	3091
	3093	3095	3097	3101	3102	3106	3108	3117	3127	3140	3147	3153	3155
	3157	3159	3169	3176	3178	3180	3182	3184	3192	3194	3200	3205	3209
	3210	3214	3226	3228	3234	3235	3242	3280	3290	3304	3311	3316	3320
	3324	3326	3328	3330	3334	3335	3342	3355	3368	3372	3377	3379	3381
	3383	3385	3459	3461	3463	3468	3469	3471	3473	3475	3480	3482	3484
	3486	3488	3490	3492	3496	3498	3500	3502	3507	3509	3511	3519	3581
	3586	3589	3660	3676	3678	3685	3693	3696	3759	3847	3849	3851	3867
	3869	3878	3911	3919	3923	3927	3936	3938	3940	3942	3944	3946	3948
	3950	3952	3974	4061	4063	4065	4070	4102	4106	4107	4112	4118	4126
	4129	4134	4137	4144	4149	4176	4178	4236	4285	4346	4348	4350	4363
	4364	4365	4392	4394	4396								
SF SBLA= 000170	674#												
SF SCAS= 000150	674#												
SF \$DEC= 000220	674#												
SF \$DO - 000340	674#	1725	1945	3468									
SF \$FAL 000405	674#												
SF \$GOO- 000400	674#	1092	1093	1094	1097	1098	1101	1102	1155	1156	1228	1229	1230
	1233	1234	1237	1238	1362	1367	1368	1369	1370	1384	1386	1387	1388
	1391	1392	1405	1407	1408	1415	1416	1417	1418	1422	1423	1424	1425
	1426	1427	1428	1429	1430	1431	1434	1435	1436	1437	1446	1447	1455

1457	1458	1459	1462	1463	1466	1467	1477	1478	1481	1482	1483	1484
1485	1486	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
1533	1534	1535	1536	1537	1538	1544	1545	1582	1583	1588	1589	1592
1593	1596	1597	1598	1599	1606	1609	1610	1611	1612	1619	1620	1627
1628	1629	1630	1631	1632	1636	1637	1638	1639	1644	1645	1648	1649
1652	1653	1674	1677	1679	1686	1687	1688	1689	1695	1696	1697	1698
1699	1700	1701	1703	1704	1705	1706	1710	1720	1721	1729	1738	1743
1757	1799	1800	1801	1802	1806	1807	1808	1809	1810	1815	1826	1831
1845	1850	1864	1865	1866	1874	1918	1919	1920	1924	1925	1929	1930
1931	1935	1936	1940	1941	1949	1950	1951	1952	1953	1954	1955	1956
1957	1958	1990	1991	1992	1996	1997	2001	2002	2003	2004	2005	2006
2007	2008	2012	2013	2014	2015	2019	2020	2024	2025	2026	2027	2031
2032	2033	2034	2035	2039	2040	2041	2042	2043	2047	2058	2059	2063
2064	2065	2066	2070	2081	2082	2083	2084	2085	2086	2087	2088	2089
2093	2094	2095	2096	2097	2098	2125	2130	2151	2154	2155	2156	2157
2158	2163	2172	2173	2174	2175	2176	2187	2198	2199	2200	2201	2202
2203	2204	2205	2206	2211	2231	2249	2250	2251	2252	2253	2264	2265
2266	2267	2268	2269	2270	2271	2272	2273	2274	2279	2297	2298	2299
2300	2301	2302	2303	2339	2355	2363	2364	2365	2366	2367	2368	2369
2370	2371	2372	2373	2377	2383	2384	2385	2386	2387	2388	2389	2393
2396	2397	2398	2399	2413	2414	2415	2416	2417	2418	2419	2420	2421
2422	2423	2424	2425	2426	2431	2453	2454	2455	2456	2500	2501	2502
2503	2504	2505	2514	2519	2528	2529	2530	2531	2532	2533	2534	2535
2536	2537	2538	2539	2552	2553	2560	2561	2567	2575	2576	2577	2578
2592	2608	2619	2620	2621	2622	2623	2624	2625	2626	2627	2628	2629
2630	2694	2695	2699	2700	2709	2710	2711	2712	2713	2714	2718	2724
2727	2728	2729	2730	2731	2732	2733	2734	2735	2736	2758	2759	2760
2761	2762	2763	2764	2765	2766	2767	2768	2773	2774	2775	2776	2777
2778	2779	2780	2781	2782	2790	2791	2792	2793	2794	2805	2806	2807
2808	2809	2810	2814	2815	2816	2817	2822	2823	2824	2825	2826	2827
2828	2829	2837	2838	2890	2899	2905	2906	2907	2908	2909	2910	2911
2912	2913	2914	2921	2927	2928	2929	2930	2937	2938	2939	2940	2941
2944	2945	2946	2947	2960	2961	2962	2963	2995	2996	3000	3003	3004
3005	3006	3012	3013	3014	3021	3029	3036	3037	3038	3039	3040	3041
3042	3043	3044	3045	3046	3049	3050	3053	3054	3055	3056	3057	3058
3059	3060	3061	3064	3065	3071	3072	3079	3089	3090	3091	3092	3093
3094	3095	3096	3097	3098	3099	3102	3103	3106	3107	3108	3109	3117
3118	3127	3128	3140	3141	3145	3153	3154	3155	3156	3157	3158	3159
3160	3167	3176	3177	3178	3179	3180	3181	3182	3183	3184	3185	3192
3193	3194	3195	3200	3201	3205	3206	3207	3210	3211	3214	3215	3226
3227	3228	3229	3232	3235	3236	3242	3243	3280	3281	3290	3291	3304
3305	3309	3316	3317	3318	3324	3325	3326	3327	3328	3329	3330	3331
3332	3335	3336	3342	3343	3355	3356	3368	3369	3372	3373	3374	3377
3378	3379	3380	3381	3382	3383	3384	3385	3386	3459	3460	3461	3462
3463	3464	3465	3466	3469	3470	3471	3472	3473	3474	3475	3479	3480
3481	3482	3483	3484	3485	3486	3487	3488	3489	3490	3491	3492	3495
3496	3497	3498	3499	3500	3501	3502	3506	3507	3508	3509	3510	3511
3515	3519	3520	3581	3582	3586	3587	3588	3589	3590	3594	3599	3606
3626	3635	3637	3658	3676	3677	3678	3679	3685	3686	3693	3694	3695
3696	3697	3698	3704	3706	3731	3738	3740	3757	3760	3765	3767	3772
3800	3806	3808	3847	3848	3849	3850	3851	3852	3856	3867	3868	3869
3870	3871	3876	3878	3879	3911	3912	3913	3917	3923	3924	3925	3936
3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949
3950	3951	3952	3953	3972	4061	4062	4063	4064	4065	4066	4070	4071
4072	4078	4080	4102	4103	4104	4107	4108	4112	4113	4118	4119	4126
4127	4128	4129	4130	4134	4135	4136	4137	4138	4144	4145	4149	4150

\$F\$IF = 000110

\$F\$INC= 000210
\$F\$LOO= 000200
\$F\$NAM= 000160
\$F\$NO - 000403

4154	4165	4176	4177	4178	4179	4180	4186	4188	4210	4218	4220	4234
4237	4243	4245	4250	4278	4285	4286	4287	4293	4295	4300	4346	4347
4348	4349	4350	4351	4361	4364	4365	4366	4367	4368	4392	4393	4394
4395	4396	4397										
674#	1094	1099	1230	1235	1362	1371	1384	1388	1393	1395	1397	1405
1409	1455	1459	1464	1468	1593	1600	1606	1616	1621	1649	1654	1674
1679	1684	1690	1721	1743	1752	1831	1840	1850	1859	1874	1883	1941
2130	2145	2151	2163	2170	2181	2211	2224	2231	2244	2279	2292	2339
2350	2377	2393	2411	2431	2442	2444	2450	2465	2467	2519	2557	2562
2567	2579	2592	2608	2643	2645	2651	2718	2724	2749	2751	2890	2899
2918	2921	2931	2948	2950	2957	2964	3000	3010	3014	3021	3023	3029
3046	3051	3061	3066	3073	3079	3081	3099	3104	3110	3119	3121	3129
3134	3145	3161	3167	3207	3212	3219	3232	3244	3246	3248	3250	3309
3318	3332	3337	3344	3360	3362	3405	3599	3621	3635	3652	3658	3671
3704	3721	3738	3755	3757	3765	3787	3789	3806	3823	3876	3883	3917
3925	3967	3972	3991	4009	4011	4013	4078	4095	4104	4109	4114	4186
4198	4218	4230	4234	4243	4265	4267	4293	4310	4361	4387		
674#												
674#												
674#												
674#	1092	1093	1094	1097	1098	1101	1102	1155	1156	1228	1229	1230
1233	1234	1237	1238	1367	1368	1369	1370	1386	1387	1388	1391	1392
1407	1408	1415	1416	1417	1418	1422	1423	1424	1425	1426	1427	1428
1429	1430	1431	1434	1435	1436	1437	1446	1447	1457	1458	1459	1462
1463	1466	1467	1477	1478	1481	1482	1483	1484	1485	1486	1493	1495
1496	1497	1498	1499	1500	1501	1502	1503	1533	1534	1535	1536	1537
1538	1544	1545	1582	1583	1588	1592	1593	1596	1597	1598	1599	1606
1609	1610	1611	1612	1619	1620	1627	1628	1629	1630	1631	1632	1636
1637	1638	1639	1644	1648	1649	1652	1653	1674	1683	1686	1687	1688
1689	1695	1696	1697	1699	1700	1701	1703	1704	1705	1706	1721	1723
1751	1799	1800	1801	1802	1806	1808	1809	1810	1839	1858	1864	1865
1882	1918	1920	1924	1925	1929	1931	1935	1936	1941	1943	1949	1950
1951	1952	1953	1954	1955	1956	1957	1958	1990	1992	1996	1997	2001
2002	2003	2004	2005	2006	2007	2008	2012	2013	2014	2015	2019	2020
2024	2025	2026	2027	2031	2033	2034	2035	2039	2041	2042	2043	2058
2059	2063	2064	2065	2066	2081	2083	2084	2085	2086	2087	2088	2089
2093	2095	2096	2098	2144	2151	2154	2155	2156	2157	2169	2172	2174
2175	2176	2180	2198	2200	2201	2202	2203	2204	2205	2206	2211	2223
2243	2249	2250	2251	2252	2264	2266	2267	2268	2269	2270	2271	2272
2273	2274	2279	2291	2297	2298	2299	2301	2302	2303	2339	2349	2363
2365	2366	2367	2368	2369	2370	2371	2372	2373	2377	2383	2385	2386
2387	2388	2389	2393	2396	2397	2398	2399	2407	2413	2415	2416	2417
2418	2419	2420	2422	2423	2424	2425	2426	2431	2441	2453	2454	2455
2456	2464	2500	2501	2502	2503	2504	2505	2528	2529	2530	2531	2532
2533	2534	2535	2536	2537	2538	2539	2552	2553	2560	2561	2567	2575
2576	2577	2578	2592	2608	2619	2620	2621	2622	2623	2624	2625	2626
2627	2628	2629	2630	2642	2650	2694	2695	2699	2700	2709	2710	2711
2712	2713	2714	2718	2724	2727	2728	2729	2730	2731	2732	2733	2734
2735	2736	2748	2758	2760	2761	2762	2763	2764	2765	2766	2767	2768
2773	2775	2776	2777	2778	2779	2780	2782	2790	2791	2792	2794	2805
2807	2808	2810	2814	2815	2816	2817	2822	2823	2824	2825	2826	2827
2828	2829	2837	2838	2890	2899	2905	2906	2907	2908	2909	2910	2911
2912	2913	2914	2921	2927	2928	2929	2930	2937	2939	2940	2941	2944
2945	2946	2947	2960	2961	2962	2963	2995	2996	3000	3003	3004	3005
3006	3012	3013	3014	3036	3037	3038	3039	3040	3041	3042	3043	3044
3045	3046	3049	3050	3053	3054	3055	3056	3057	3058	3059	3060	3061

	3064	3065	3071	3072	3089	3090	3091	3092	3093	3094	3095	3096	3097
	3098	3099	3102	3103	3106	3107	3108	3109	3117	3118	3127	3128	3140
	3141	3145	3153	3154	3155	3156	3157	3158	3159	3160	3167	3176	3177
	3178	3179	3180	3181	3182	3183	3184	3185	3192	3193	3194	3195	3200
	3201	3205	3206	3207	3210	3211	3214	3215	3226	3227	3228	3229	3232
	3235	3236	3242	3243	3280	3281	3290	3291	3304	3305	3309	3316	3317
	3318	3324	3325	3326	3327	3328	3329	3330	3331	3332	3335	3336	3342
	3343	3355	3356	3359	3368	3369	3372	3374	3377	3378	3379	3380	3381
	3382	3383	3384	3385	3386	3404	3459	3460	3461	3462	3463	3464	3466
	3469	3470	3471	3472	3473	3474	3475	3479	3480	3481	3482	3483	3484
	3485	3486	3487	3488	3489	3490	3491	3492	3495	3496	3497	3498	3499
	3500	3501	3502	3506	3507	3508	3509	3510	3511	3515	3519	3520	3581
	3582	3586	3588	3589	3590	3658	3676	3677	3678	3679	3685	3686	3693
	3695	3696	3697	3757	3847	3848	3849	3850	3851	3852	3867	3868	3869
	3870	3878	3879	3882	3911	3913	3923	3924	3936	3937	3938	3939	3940
	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951	3952	3953
	3966	3990	4008	4061	4062	4063	4064	4065	4066	4070	4071	4102	4103
	4104	4107	4108	4112	4113	4118	4119	4126	4128	4129	4130	4134	4136
	4137	4138	4144	4145	4149	4150	4176	4177	4178	4179	4234	4285	4286
	4346	4347	4348	4349	4350	4351	4364	4365	4366	4367	4386	4392	4393
	4394	4395	4396	4397									
\$FSOR - 000320	674#	1096	1232	1390	1461	1595	1608	1651	1676	1723	1725	1943	1945
	2153	2213	2233	2281	2341	2379	2395	2433	2569	2594	2610	2720	2726
	2892	2901	2923	3002	3016	3023	3025	3031	3048	3063	3081	3101	3147
	3169	3209	3234	3311	3320	3334	3468	3660	3759	3919	3927	3974	4106
	4236	4363											
\$FSRTI - 000350	674#												
\$FSRTN= 000300	674#	1571	1656	1673	1771	1787	1888	1908	1965	1982	2103	2116	2306
	2328	2472	2490	2653	2679	2839	3276	3407	3423	3424	3446	3527	3842
	3885	3901	4021	4334	4410								
\$FSSEL = 000140	674#												
\$FS THE = 000330	674#	1096	1232	1390	1461	1595	1608	1651	1676	2153	2213	2233	2281
	2341	2379	2395	2433	2569	2594	2610	2720	2726	2892	2901	2923	3002
	3016	3025	3031	3048	3063	3084	3101	3147	3169	3209	3234	3311	3320
	3334	3660	3759	3919	3927	3974	4106	4236	4363				
\$FSTRU= 000404	674#												
\$FSUNT- 000130	674#	1432	1438	1444	1470	1479	1487	2509	2585	2598	2602	2701	2703
	3285	3295	3680	3687	3726	3794	4116	4120	4205	4272	4355	4401	
\$FSWHI- 000120	674#	1720	1721	1723	1765	1940	1941	1943	1959	3023	3081	3465	3466
	3516												
\$FSYES- 000402	674#	1093	1094	1098	1099	1102	1156	1229	1230	1234	1235	1238	1362
	1368	1370	1371	1384	1387	1388	1392	1393	1395	1397	1405	1408	1409
	1416	1418	1423	1425	1427	1429	1431	1435	1437	1447	1455	1458	1459
	1463	1464	1467	1468	1478	1482	1484	1486	1495	1497	1499	1501	1503
	1534	1536	1538	1545	1583	1592	1593	1597	1599	1600	1601	1610	1612
	1616	1620	1621	1628	1630	1632	1637	1639	1648	1649	1651	1654	1674
	1679	1684	1687	1689	1690	1696	1699	1701	1704	1706	1721	1723	1743
	1752	1800	1802	1808	1810	1831	1840	1850	1859	1865	1874	1883	1920
	1925	1931	1936	1941	1943	1950	1952	1954	1956	1958	1992	1997	2002
	2004	2006	2008	2013	2015	2020	2025	2027	2033	2035	2041	2043	2059
	2064	2066	2083	2085	2087	2089	2095	2098	2130	2145	2151	2155	2157
	2163	2170	2174	2176	2181	2200	2202	2204	2206	2211	2224	2231	2244
	2250	2252	2266	2268	2270	2272	2274	2279	2292	2298	2301	2303	2339
	2350	2365	2367	2369	2371	2373	2377	2385	2387	2389	2393	2397	2399
	2411	2415	2417	2419	2422	2424	2426	2431	2442	2444	2450	2454	2456
	2465	2467	2501	2503	2505	2519	2529	2531	2533	2535	2537	2539	2553

2557	2561	2562	2567	2576	2578	2579	2592	2608	2620	2622	2624	2626	
2628	2630	2643	2645	2651	2695	2700	2710	2712	2714	2718	2724	2728	
2730	2732	2734	2736	2749	2751	2760	2762	2764	2766	2768	2775	2777	
2779	2782	2791	2794	2807	2810	2815	2817	2823	2825	2827	2829	2838	
2890	2899	2906	2908	2910	2912	2914	2918	2921	2928	2930	2931	2939	
2941	2945	2947	2948	2950	2957	2961	2963	2964	2996	3000	3004	3006	
3010	3013	3014	3021	3023	3029	3037	3039	3041	3043	3045	3046	3050	
3051	3054	3056	3058	3060	3061	3065	3066	3072	3073	3079	3081	3090	
3092	3094	3096	3098	3099	3103	3104	3107	3109	3110	3118	3119	3121	
3128	3129	3134	3141	3145	3154	3156	3158	3160	3161	3167	3177	3179	
3181	3183	3185	3193	3195	3201	3206	3207	3211	3212	3215	3219	3227	
3229	3232	3236	3243	3244	3246	3248	3250	3281	3291	3305	3309	3317	
3318	3325	3327	3329	3331	3332	3336	3337	3343	3344	3356	3360	3362	
3369	3374	3378	3380	3382	3384	3386	3405	3460	3462	3464	3466	3470	
3472	3474	3479	3481	3483	3485	3487	3489	3491	3495	3497	3499	3501	
3506	3508	3510	3515	3520	3582	3588	3590	3599	3621	3635	3652	3658	
3671	3677	3679	3686	3695	3697	3704	3721	3738	3755	3757	3765	3787	
3789	3806	3823	3848	3850	3852	3868	3870	3876	3879	3883	3913	3917	
3924	3925	3937	3939	3941	3943	3945	3947	3949	3951	3953	3967	3972	
3991	4009	4011	4013	4062	4064	4066	4071	4078	4095	4103	4104	4108	
4109	4113	4114	4119	4128	4130	4136	4138	4145	4150	4177	4179	4186	
4198	4218	4230	4234	4243	4265	4267	4286	4293	4310	4347	4349	4351	
4361	4365	4367	4387	4393	4395	4397							
\$IFLEV= 177777	674#	1094#	1099#	1230#	1235#	1362#	1384#	1388#	1393#	1395#	1397#	1405#	1409#
	1455#	1459#	1464#	1468#	1593#	1600#	1606#	1621#	1649#	1654#	1674#	1679#	1684#
	1690#	1743#	1752#	1831#	1840#	1850#	1859#	1874#	1883#	2130#	2145#	2151#	2163#
	2170#	2181#	2211#	2224#	2231#	2244#	2279#	2292#	2339#	2377#	2393#	2431#	2442#
	2444#	2465#	2467#	2519#	2562#	2567#	2579#	2592#	2608#	2643#	2651#	2718#	2724#
	2749#	2751#	2890#	2899#	2921#	2948#	2950#	2964#	3000#	3010#	3014#	3021#	3029#
	3046#	3051#	3061#	3066#	3079#	3099#	3104#	3119#	3121#	3129#	3145#	3167#	3207#
	3212#	3232#	3244#	3246#	3248#	3250#	3309#	3318#	3332#	3337#	3360#	3405#	3599#
	3621#	3635#	3652#	3658#	3671#	3704#	3721#	3738#	3755#	3757#	3765#	3787#	3789#
	3806#	3823#	3876#	3883#	3917#	3925#	3972#	4009#	4011#	4013#	4078#	4095#	4104#
\$ISK0 000001	4114#	4186#	4198#	4218#	4230#	4234#	4243#	4265#	4267#	4293#	4310#	4361#	4387#
	1094#	1099	1230#	1235	1362#	1397	1405#	1409	1455#	1468	1593#	1600	1606#
	1621	1649#	1654	1674#	1690	1743#	1752	1831#	1840	1850#	1859	1874#	1883
	2130#	2145	2151#	2181	2211#	2224	2231#	2244	2279#	2292	2339#	2467	2519#
	2562	2567#	2579	2592#	2651	2718#	2751	2890#	2964	3000#	3010	3014#	3250
	3309#	3405	3599#	3621	3635#	3652	3658#	3671	3704#	3721	3738#	3755	3757#
	3789	3806#	3823	3876#	3883	3917#	4013	4078#	4095	4104#	4114	4186#	4198
\$ISK1 000001	4218#	4230	4234#	4267	4293#	4310	4361#	4387					
	1384#	1395	1459#	1464	1679#	1684	2163#	2170	2377#	2465	2608#	2643	2724#
	2749	2899#	2950	3021#	3129	3145#	3248	3318#	3360	3765#	3787	3925#	4011
	4243#	4265											
\$ISK2 = 000001	1388#	1393	2393#	2444	2921#	2948	3029#	3121	3167#	3246	3332#	3337	3972#
	4009												
\$ISK3 = 000001	2431#	2442	3046#	3051	3061#	3066	3079#	3119	3207#	3212	3232#	3244	
\$ISK4 = 000001	3099#	3104											
\$LINE 015634	1502*	2709	2822	2845#	2944								
\$LOCTA= 177777	674#	1095	1099	1231	1235	1362	1371	1372	1384	1389	1393	1395	1397
	1405	1409	1432	1439	1444	1455	1460	1464	1468	1471	1479	1488	1570
	1594	1600	1607	1616	1617	1621	1650	1654	1656	1657	1672	1675	1679
	1682	1684	1690	1720	1722	1724	1743	1750	1752	1765	1766	1771	1773
	1786	1831	1838	1840	1850	1857	1859	1874	1881	1883	1888	1890	1907
	1940	1942	1944	1959	1960	1965	1966	1981	2103	2104	2115	2130	2143
	2145	2152	2163	2168	2170	2179	2181	2212	2222	2224	2232	2242	2244

2280	2290	2292	2306	2308	2327	2340	2348	2350	2378	2394	2406	2411
2432	2440	2442	2444	2450	2451	2463	2465	2467	2472	2474	2489	2509
2519	2557	2558	2562	2568	2579	2586	2593	2598	2603	2609	2641	2643
2645	2646	2649	2651	2653	2655	2678	2701	2704	2719	2725	2747	2749
2751	2839	2841	2891	2900	2918	2919	2922	2931	2932	2948	2950	2957
2958	2964	3001	3010	3015	3022	3024	3030	3047	3051	3062	3066	3073
3074	3080	3082	3083	3100	3104	3110	3111	3119	3121	3129	3134	3135
3146	3161	3162	3168	3208	3212	3219	3220	3233	3244	3246	3248	3250
3275	3285	3296	3298	3299	3310	3319	3333	3337	3344	3345	3358	3360
3362	3363	3403	3405	3407	3409	3422	3424	3425	3445	3465	3467	3516
3517	3527	3528	3599	3621	3635	3652	3659	3671	3680	3688	3704	3721
3726	3738	3755	3758	3765	3787	3789	3795	3806	3823	3841	3876	3881
3883	3885	3887	3900	3918	3926	3965	3967	3973	3989	3991	4007	4009
4011	4013	4021	4023	4078	4095	4105	4109	4110	4114	4116	4121	4186
4198	4205	4218	4230	4235	4243	4265	4267	4273	4293	4310	4333	4355
4362	4385	4387	4402	4410	4412							
674#	1092	1094	1095	1097	1101	1153	1155	1185	1228	1230	1231	1233
1237	1351	1362	1367	1369	1371	1384	1386	1388	1389	1391	1405	1407
1415	1417	1422	1424	1426	1428	1430	1434	1436	1438	1439	1446	1455
1457	1459	1460	1462	1466	1470	1471	1477	1481	1483	1485	1487	1488
1493	1494	1496	1498	1500	1502	1507	1531	1533	1535	1537	1539	1544
1575	1576	1577	1582	1588	1589	1590	1591	1593	1594	1596	1598	1606
1607	1609	1611	1616	1619	1627	1629	1631	1636	1638	1644	1645	1646
1647	1649	1650	1652	1658	1659	1660	1674	1675	1677	1679	1681	1682
1686	1688	1695	1697	1698	1700	1703	1705	1710	1711	1712	1713	1721
1722	1723	1724	1729	1730	1731	1732	1733	1734	1735	1736	1738	1743
1749	1750	1757	1758	1759	1760	1765	1772	1774	1790	1791	1792	1799
1801	1806	1807	1809	1815	1816	1817	1818	1819	1820	1821	1826	1831
1837	1838	1845	1850	1856	1857	1864	1866	1867	1868	1869	1874	1880
1881	1889	1891	1892	1893	1911	1912	1913	1918	1919	1924	1929	1930
1935	1941	1942	1943	1944	1949	1951	1953	1955	1957	1959	1967	1968
1969	1983	1985	1990	1991	1996	2001	2003	2005	2007	2012	2014	2019
2024	2026	2031	2032	2034	2039	2040	2042	2047	2048	2049	2050	2051
2052	2053	2058	2063	2065	2070	2071	2072	2073	2074	2075	2076	2081
2082	2084	2086	2088	2093	2094	2096	2097	2099	2101	2105	2120	2125
2130	2140	2142	2143	2151	2152	2154	2156	2158	2159	2160	2161	2163
2165	2167	2168	2172	2173	2175	2177	2179	2187	2188	2189	2190	2191
2192	2193	2198	2199	2201	2203	2205	2211	2212	2219	2221	2222	2231
2232	2239	2241	2242	2249	2251	2253	2254	2255	2256	2257	2258	2259
2264	2265	2267	2269	2271	2273	2279	2280	2287	2289	2290	2297	2299
2300	2302	2304	2307	2309	2331	2332	2333	2339	2340	2347	2348	2355
2356	2357	2358	2359	2360	2361	2363	2364	2366	2368	2370	2372	2377
2378	2383	2384	2386	2388	2393	2394	2396	2398	2405	2406	2413	2414
2416	2418	2420	2421	2423	2425	2431	2432	2439	2440	2450	2453	2455
2462	2463	2473	2475	2476	2477	2493	2494	2495	2500	2502	2504	2514
2519	2521	2523	2528	2530	2532	2534	2536	2538	2540	2542	2552	2557
2560	2567	2568	2575	2577	2585	2586	2592	2593	2602	2603	2608	2609
2611	2613	2619	2621	2623	2625	2627	2629	2631	2633	2640	2641	2645
2648	2649	2654	2656	2657	2658	2682	2683	2684	2687	2689	2694	2699
2703	2704	2709	2711	2713	2718	2719	2724	2725	2727	2729	2731	2733
2735	2737	2739	2746	2747	2758	2759	2761	2763	2765	2767	2773	2774
2776	2778	2780	2781	2790	2792	2793	2805	2806	2808	2809	2814	2816
2822	2824	2826	2828	2830	2832	2837	2840	2842	2843	2844	2890	2891
2897	2899	2900	2905	2907	2909	2911	2913	2918	2921	2922	2927	2929
2931	2937	2938	2940	2944	2946	2952	2957	2960	2962	2987	2995	3000
3001	3003	3005	3007	3012	3014	3015	3021	3022	3023	3024	3029	3030

SLSTIN= 000001

3036	3038	3040	3042	3044	3046	3047	3049	3053	3055	3057	3059	3061
3062	3064	3071	3073	3079	3080	3081	3082	3089	3091	3093	3095	3097
3099	3100	3102	3106	3108	3110	3117	3127	3134	3140	3145	3146	3153
3155	3157	3159	3161	3167	3168	3174	3176	3178	3180	3182	3184	3186
3192	3194	3200	3205	3207	3208	3210	3214	3219	3226	3228	3232	3233
3235	3242	3255	3280	3290	3295	3296	3297	3298	3304	3309	3310	3316
3318	3319	3324	3326	3328	3330	3332	3333	3335	3342	3344	3355	3357
3358	3362	3368	3370	3372	3373	3375	3377	3379	3381	3383	3385	3387
3389	3402	3403	3408	3410	3426	3449	3450	3451	3453	3455	3457	3459
3461	3463	3466	3467	3469	3471	3473	3475	3476	3477	3478	3480	3482
3484	3486	3488	3490	3492	3493	3494	3496	3498	3500	3502	3503	3504
3505	3507	3509	3511	3512	3513	3514	3516	3519	3521	3523	3525	3529
3530	3531	3581	3586	3587	3589	3594	3599	3606	3607	3608	3609	3626
3627	3628	3635	3637	3638	3639	3640	3658	3659	3676	3678	3685	3687
3688	3693	3694	3696	3698	3704	3706	3707	3708	3709	3731	3732	3733
3738	3740	3741	3742	3743	3757	3758	3760	3765	3767	3768	3769	3770
3772	3773	3774	3775	3794	3795	3800	3806	3808	3809	3810	3811	3847
3849	3851	3856	3857	3858	3859	3860	3861	3862	3867	3869	3871	3876
3878	3880	3881	3886	3888	3902	3904	3906	3911	3912	3917	3918	3923
3925	3926	3936	3938	3940	3942	3944	3946	3948	3950	3952	3958	3960
3962	3964	3965	3972	3973	3982	3984	3986	3988	3989	4000	4002	4004
4006	4007	4015	4017	4019	4022	4024	4061	4063	4065	4070	4072	4078
4080	4081	4082	4083	4100	4102	4104	4105	4107	4109	4112	4118	4120
4121	4126	4127	4129	4134	4135	4137	4139	4144	4149	4154	4155	4156
4157	4158	4159	4160	4165	4166	4167	4168	4169	4170	4171	4176	4178
4180	4186	4188	4189	4190	4191	4210	4211	4212	4218	4220	4221	4222
4223	4234	4235	4237	4243	4245	4246	4247	4248	4250	4251	4252	4253
4272	4273	4278	4285	4287	4293	4295	4296	4297	4298	4300	4301	4302
4303	4335	4337	4339	4346	4348	4350	4361	4362	4364	4366	4368	4369
4370	4371	4378	4380	4382	4384	4385	4392	4394	4396	4401	4402	4404
4406	4408	4411	4413									
674#	1099	1235	1372	1393	1395	1397	1409	1432	1444	1464	1468	1479
1570	1600	1617	1621	1654	1656	1657	1672	1684	1690	1720	1752	1766
1771	1773	1786	1840	1859	1883	1888	1890	1907	1940	1960	1965	1966
1981	2103	2104	2115	2145	2170	2181	2224	2244	2292	2306	2308	2327
2350	2411	2442	2444	2451	2465	2467	2472	2474	2489	2509	2558	2562
2579	2598	2643	2646	2651	2653	2655	2678	2701	2749	2751	2839	2841
2919	2932	2948	2950	2958	2964	3010	3051	3066	3074	3083	3104	3111
3119	3121	3129	3135	3162	3212	3220	3244	3246	3248	3250	3275	3285
3299	3337	3345	3360	3363	3405	3407	3409	3422	3424	3425	3445	3465
3517	3527	3528	3621	3652	3671	3680	3721	3726	3755	3787	3789	3823
3841	3883	3885	3887	3900	3967	3991	4009	4011	4013	4021	4023	4095
4110	4114	4116	4198	4205	4230	4265	4267	4310	4333	4355	4387	4410
4412												
674#	1094#	1099#	1230#	1235#	1362#	1371	1384#	1388#	1393#	1395#	1397#	1405#
1409#	1432#	1438#	1444#	1455#	1459#	1464#	1468#	1470#	1479#	1487#	1571#	1593#
1600#	1606#	1616	1621#	1649#	1654#	1656#	1673#	1674#	1679#	1684#	1690#	1720#
1743#	1752#	1765#	1771#	1787#	1831#	1840#	1850#	1859#	1874#	1883#	1888#	1908#
1940#	1959#	1965#	1982#	2103#	2116#	2130#	2145#	2151#	2163#	2170#	2181#	2211#
2224#	2231#	2244#	2279#	2292#	2306#	2328#	2339#	2350	2377#	2393#	2411	2431#
2442#	2444#	2450	2465#	2467#	2472#	2490#	2509#	2519#	2557	2562#	2567#	2579#
2585#	2592#	2598#	2602#	2608#	2643#	2645	2651#	2653#	2679#	2701#	2703#	2718#
2724#	2749#	2751#	2839#	2890#	2899#	2918	2921#	2931	2948#	2950#	2957	2964#
3000#	3010#	3014#	3021#	3029#	3046#	3051#	3061#	3066#	3073	3079#	3099#	3104#
3110	3119#	3121#	3129#	3134	3145#	3161	3167#	3207#	3212#	3219	3232#	3244#
3246#	3248#	3250#	3276#	3285#	3295#	3309#	3318#	3332#	3337#	3344	3360#	3362

SLSTA= 000001

SNESTL= 177777

	3405#	3407#	3423#	3424#	3446#	3465#	3516#	3527#	3599#	3621#	3635#	3652#	3658#
	3671#	3680#	3687#	3704#	3721#	3726#	3738#	3755#	3757#	3765#	3787#	3789#	3794#
	3806#	3823#	3842#	3876#	3883#	3885#	3901#	3917#	3925#	3967	3972#	3991	4009#
	4011#	4013#	4021#	4078#	4095#	4104#	4109	4114#	4116#	4120#	4186#	4198#	4205#
	4218#	4230#	4234#	4243#	4265#	4267#	4272#	4293#	4310#	4334#	4355#	4361#	4387#
	4401#	4410#											
\$NSK0 - 000300	1094#	1099	1230#	1235	1362#	1371	1397	1405#	1409	1432#	1438	1444#	1470
	1479#	1487	1571#	1656	1673#	1771	1787#	1888	1908#	1965	1982#	2103	2116#
	2306	2328#	2472	2490#	2653	2679#	2839	2890#	2957	2964	3000#	3010	3014#
	3134	3250	3276#	3407	3423#	3424	3446#	3527	3599#	3621	3635#	3652	3658#
	3671	3680#	3687	3704#	3721	3726#	3794	3806#	3823	3842#	3885	3901#	4021
	4078#	4095	4104#	4109	4114	4116#	4120	4186#	4198	4205#	4272	4293#	4310
	4334#	4410											
\$NSK1 = 000130	1384#	1395	1455#	1468	1593#	1600	1606#	1616	1621	1649#	1654	1674#	1690
	1720#	1765	1831#	1840	1850#	1859	1874#	1883	1940#	1959	2130#	2145	2151#
	2181	2211#	2224	2231#	2244	2279#	2292	2339#	2350	2467	2509#	2585	2592#
	2645	2651	2701#	2703	2718#	2751	2899#	2918	2950	3021#	3129	3145#	3161
	3248	3285#	3295	3309#	3362	3405	3465#	3516	3738#	3755	3757#	3789	3876#
	3883	3917#	4013	4218#	4230	4234#	4267	4355#	4401				
\$NSK2 = 000110	1388#	1393	1459#	1464	1679#	1684	1743#	1752	2163#	2170	2377#	2450	2465
	2519#	2557	2562	2567#	2579	2598#	2602	2608#	2643	2724#	2749	2921#	2931
	2948	3029#	3073	3121	3167#	3219	3246	3318#	3344	3360	3765#	3787	3925#
	3967	4011	4243#	4265	4361#	4387							
\$NSK3 = 000110	2393#	2411	2444	3046#	3051	3061#	3066	3079#	3110	3119	3207#	3212	3232#
	3244	3332#	3337	3972#	3991	4009							
\$NSK4 = 000110	2431#	2442	3099#	3104									
\$SAVLE= 177777	674#	1765#	1766#	1959#	1960#	3516#	3517#						
\$SAV2 053140 G	4604#												
\$SAV3 053154 G	4604#												
\$SAV4 053172 G	4604#												
\$SAV5 053212 G	4604#												
\$SSK0 = 050003	1765#	1766	1959#	1960	3516#	3517							
\$TAGLE= 177777	674#	1096#	1099#	1232#	1235#	1363#	1372#	1373#	1385#	1390#	1393#	1395#	1397#
	1406#	1409#	1433#	1438#	1445#	1456#	1461#	1464#	1468#	1470#	1480#	1487#	1595#
	1600#	1608#	1617#	1618#	1621#	1651#	1654#	1676#	1680#	1684#	1690#	1721#	1725#
	1744#	1752#	1765#	1832#	1840#	1851#	1859#	1875#	1883#	1941#	1945#	1959#	2131#
	2145#	2153#	2164#	2170#	2181#	2213#	2224#	2233#	2244#	2281#	2292#	2341#	2350#
	2351#	2379#	2395#	2411#	2412#	2433#	2442#	2444#	2451#	2452#	2465#	2467#	2510#
	2520#	2558#	2559#	2562#	2569#	2579#	2585#	2594#	2599#	2602#	2610#	2643#	2646#
	2647#	2651#	2702#	2703#	2720#	2726#	2749#	2751#	2892#	2901#	2919#	2920#	2923#
	2932#	2933#	2948#	2950#	2958#	2959#	2964#	3002#	3010#	3016#	3025#	3031#	3048#
	3051#	3063#	3066#	3074#	3075#	3084#	3101#	3104#	3111#	3112#	3119#	3121#	3129#
	3135#	3136#	3147#	3162#	3163#	3169#	3209#	3212#	3220#	3221#	3234#	3244#	3246#
	3248#	3250#	3286#	3295#	3311#	3320#	3334#	3337#	3345#	3346#	3360#	3363#	3364#
	3405#	3466#	3468#	3516#	3600#	3621#	3636#	3652#	3660#	3671#	3681#	3687#	3705#
	3721#	3727#	3739#	3755#	3759#	3766#	3787#	3789#	3794#	3807#	3823#	3877#	3883#
	3919#	3927#	3967#	3968#	3974#	3991#	3992#	4009#	4011#	4013#	4079#	4095#	4106#
	4110#	4111#	4114#	4117#	4120#	4187#	4198#	4206#	4219#	4230#	4236#	4244#	4265#
	4267#	4272#	4294#	4310#	4356#	4363#	4387#	4401#					
\$TAGNU- 050004	674#	1095	1096#	1231	1232#	1362	1363#	1371	1373#	1384	1385#	1389	1390#
	1405	1406#	1432	1433#	1444	1445#	1455	1456#	1460	1461#	1479	1480#	1570#
	1571#	1594	1595#	1607	1608#	1616	1618#	1650	1651#	1672#	1673#	1675	1676#
	1679	1680#	1720	1721#	1722	1724	1725#	1743	1744#	1786#	1787#	1831	1832#
	1850	1851#	1874	1875#	1907#	1908#	1940	1941#	1942	1944	1945#	1981#	1982#
	2115#	2116#	2130	2131#	2152	2153#	2163	2164#	2212	2213#	2232	2233#	2280
	2281#	2327#	2328#	2340	2341#	2351#	2378	2379#	2394	2395#	2412#	2432	2433#

STEMP 000300

2450	2452#	2489#	2490#	2509	2510#	2519	2520#	2557	2559#	2568	2569#	2593
2594#	2598	2599#	2609	2610#	2645	2647#	2678#	2679#	2701	2702#	2719	2720#
2725	2726#	2891	2892#	2900	2901#	2918	2920#	2922	2923#	2931	2933#	2957
2959#	3001	3002#	3015	3016#	3022	3024	3025#	3030	3031#	3047	3048#	3062
3063#	3073	3075#	3080	3082	3083	3084#	3100	3101#	3110	3112#	3134	3136#
3146	3147#	3161	3163#	3168	3169#	3208	3209#	3219	3221#	3233	3234#	3275#
3276#	3285	3286#	3296	3299	3300#	3310	3311#	3319	3320#	3333	3334#	3344
3346#	3362	3364#	3422#	3423#	3445#	3446#	3465	3466#	3467	3468#	3599	3600#
3635	3636#	3659	3660#	3680	3681#	3704	3705#	3726	3727#	3738	3739#	3758
3759#	3765	3766#	3806	3807#	3841#	3842#	3876	3877#	3900#	3901#	3918	3919#
3926	3927#	3968#	3973	3974#	3992#	4078	4079#	4105	4106#	4109	4111#	4116
4117#	4186	4187#	4205	4206#	4218	4219#	4235	4236#	4243	4244#	4293	4294#
4333#	4334#	4355	4356#	4362	4363#							
1092#	1093#	1097#	1098#	1099#	1101#	1102#	1155#	1156#	1228#	1229#	1233#	1234#
1235#	1237#	1238#	1367#	1368#	1369#	1370#	1371#	1372#	1386#	1387#	1391#	1392#
1393#	1395#	1397#	1407#	1408#	1409#	1415#	1416#	1417#	1418#	1422#	1423#	1424#
1425#	1426#	1427#	1428#	1429#	1430#	1431#	1434#	1435#	1436#	1437#	1438#	1439
1446#	1447#	1457#	1458#	1462#	1463#	1464#	1466#	1467#	1468#	1470#	1471	1477#
1478#	1481#	1482#	1483#	1484#	1485#	1486#	1487#	1488	1493#	1495#	1496#	1497#
1498#	1499#	1500#	1501#	1502#	1503#	1533#	1534#	1535#	1536#	1537#	1538#	1544#
1545#	1582#	1583#	1588#	1592#	1596#	1597#	1598#	1599#	1600#	1609#	1610#	1611#
1612#	1616#	1617#	1619#	1620#	1621#	1627#	1628#	1629#	1630#	1631#	1632#	1636#
1637#	1638#	1639#	1644#	1648#	1652#	1653#	1654#	1656#	1684#	1686#	1687#	1688#
1689#	1690#	1695#	1696#	1697#	1699#	1700#	1701#	1703#	1704#	1705#	1706#	1752#
1765#	1766#	1771#	1799#	1800#	1801#	1802#	1806#	1808#	1809#	1810#	1840#	1859#
1864#	1865#	1883#	1888#	1918#	1920#	1924#	1925#	1929#	1931#	1935#	1936#	1949#
1950#	1951#	1952#	1953#	1954#	1955#	1956#	1957#	1958#	1959#	1960#	1965#	1990#
1992#	1996#	1997#	2001#	2002#	2003#	2004#	2005#	2006#	2007#	2008#	2012#	2013#
2014#	2015#	2019#	2020#	2024#	2025#	2026#	2027#	2031#	2033#	2034#	2035#	2039#
2041#	2042#	2043#	2058#	2059#	2063#	2064#	2065#	2066#	2081#	2083#	2084#	2085#
2086#	2087#	2088#	2089#	2093#	2095#	2096#	2098#	2103#	2145#	2154#	2155#	2156#
2157#	2170#	2172#	2174#	2175#	2176#	2181#	2198#	2200#	2201#	2202#	2203#	2204#
2205#	2206#	2224#	2244#	2249#	2250#	2251#	2252#	2264#	2266#	2267#	2268#	2269#
2270#	2271#	2272#	2273#	2274#	2292#	2297#	2298#	2299#	2301#	2302#	2303#	2306#
2350#	2363#	2365#	2366#	2367#	2368#	2369#	2370#	2371#	2372#	2373#	2383#	2385#
2386#	2387#	2388#	2389#	2396#	2397#	2398#	2399#	2411#	2413#	2415#	2416#	2417#
2418#	2419#	2420#	2422#	2423#	2424#	2425#	2426#	2442#	2444#	2450#	2451#	2453#
2454#	2455#	2456#	2465#	2467#	2472#	2500#	2501#	2502#	2503#	2504#	2505#	2528#
2529#	2530#	2531#	2532#	2533#	2534#	2535#	2536#	2537#	2538#	2539#	2552#	2553#
2557#	2558#	2560#	2561#	2562#	2575#	2576#	2577#	2578#	2579#	2585#	2586	2602#
2603	2619#	2620#	2621#	2622#	2623#	2624#	2625#	2626#	2627#	2628#	2629#	2630#
2643#	2645#	2646#	2651#	2653#	2694#	2695#	2699#	2700#	2703#	2704	2709#	2710#
2711#	2712#	2713#	2714#	2727#	2728#	2729#	2730#	2731#	2732#	2733#	2734#	2735#
2736#	2749#	2751#	2758#	2760#	2761#	2762#	2763#	2764#	2765#	2766#	2767#	2768#
2773#	2775#	2776#	2777#	2778#	2779#	2780#	2782#	2790#	2791#	2792#	2794#	2805#
2807#	2808#	2810#	2814#	2815#	2816#	2817#	2822#	2823#	2824#	2825#	2826#	2827#
2828#	2829#	2837#	2838#	2839#	2905#	2906#	2907#	2908#	2909#	2910#	2911#	2912#
2913#	2914#	2918#	2919#	2927#	2928#	2929#	2930#	2931#	2932#	2937#	2939#	2940#
2941#	2944#	2945#	2946#	2947#	2948#	2950#	2957#	2958#	2960#	2961#	2962#	2963#
2964#	2995#	2996#	3003#	3004#	3005#	3006#	3010#	3012#	3013#	3036#	3037#	3038#
3039#	3040#	3041#	3042#	3043#	3044#	3045#	3049#	3050#	3051#	3053#	3054#	3055#
3056#	3057#	3058#	3059#	3060#	3064#	3065#	3066#	3071#	3072#	3073#	3074#	3089#
3090#	3091#	3092#	3093#	3094#	3095#	3096#	3097#	3098#	3102#	3103#	3104#	3106#
3107#	3108#	3109#	3110#	3111#	3117#	3118#	3119#	3121#	3127#	3128#	3129#	3134#
3135#	3140#	3141#	3153#	3154#	3155#	3156#	3157#	3158#	3159#	3160#	3161#	3162#
3176#	3177#	3178#	3179#	3180#	3181#	3182#	3183#	3184#	3185#	3192#	3193#	3194#

3195#	3200#	3201#	3205#	3206#	3210#	3211#	3212#	3214#	3215#	3219#	3220#	3226#
3227#	3228#	3229#	3235#	3236#	3242#	3243#	3244#	3246#	3248#	3250#	3280#	3281#
3290#	3291#	3295#	3298	3304#	3305#	3316#	3317#	3324#	3325#	3326#	3327#	3328#
3329#	3330#	3331#	3335#	3336#	3337#	3342#	3343#	3344#	3345#	3355#	3356#	3360#
3362#	3363#	3368#	3369#	3372#	3374#	3377#	3378#	3379#	3380#	3381#	3382#	3383#
3384#	3385#	3386#	3405#	3407#	3424#	3459#	3460#	3461#	3462#	3463#	3464#	3469#
3470#	3471#	3472#	3473#	3474#	3475#	3479#	3480#	3481#	3482#	3483#	3484#	3485#
3486#	3487#	3488#	3489#	3490#	3491#	3492#	3495#	3496#	3497#	3498#	3499#	3500#
3501#	3502#	3506#	3507#	3508#	3509#	3510#	3511#	3515#	3516#	3517#	3519#	3520#
3527#	3581#	3582#	3586#	3588#	3589#	3590#	3621#	3652#	3671#	3676#	3677#	3678#
3679#	3685#	3686#	3687#	3688	3693#	3695#	3696#	3697#	3721#	3755#	3787#	3789#
3794#	3795	3823#	3847#	3848#	3849#	3850#	3851#	3852#	3867#	3868#	3869#	3870#
3878#	3879#	3883#	3885#	3911#	3913#	3923#	3924#	3936#	3937#	3938#	3939#	3940#
3941#	3942#	3943#	3944#	3945#	3946#	3947#	3948#	3949#	3950#	3951#	3952#	3953#
3967#	3991#	4009#	4011#	4013#	4021#	4061#	4062#	4063#	4064#	4065#	4066#	4070#
4071#	4095#	4102#	4103#	4107#	4108#	4109#	4110#	4112#	4113#	4114#	4118#	4119#
4120#	4121	4126#	4128#	4129#	4130#	4134#	4136#	4137#	4138#	4144#	4145#	4149#
4150#	4176#	4177#	4178#	4179#	4198#	4230#	4265#	4267#	4272#	4273	4285#	4286#
4310#	4346#	4347#	4348#	4349#	4350#	4351#	4364#	4365#	4366#	4367#	4387#	4392#
4393#	4394#	4395#	4396#	4397#	4401#	4402	4410#					
\$TSK0 - 050002	1096#	1099	1232#	1235	1363#	1372	1373#	1397	1406#	1409	1433#	1438
	1470	1480#	1487	1595#	1600	1608#	1617	1618#	1621	1651#	1654	1676#
	1721#	1765	1832#	1840	1851#	1859	1875#	1883	1941#	1959	2131#	2145
	2181	2213#	2224	2233#	2244	2281#	2292	2341#	2350	2351#	2467	2510#
	2594#	2646	2647#	2651	2702#	2703	2720#	2751	2892#	2958	2959#	2964
	3010	3016#	3135	3136#	3250	3286#	3295	3311#	3363	3364#	3405	3466#
	3600#	3621	3636#	3652	3660#	3671	3681#	3687	3705#	3721	3727#	3794
	3823	3877#	3883	3919#	4013	4079#	4095	4106#	4110	4111#	4114	4117#
	4187#	4198	4206#	4272	4294#	4310	4356#	4401				
\$TSK1 = 050003	1385#	1395	1456#	1468	1680#	1684	1725#	1765	1945#	1959	2164#	2170
	2451	2452#	2465	2520#	2558	2559#	2562	2569#	2579	2599#	2602	2610#
	2726#	2749	2901#	2919	2920#	2950	3025#	3129	3147#	3162	3163#	3248
	3345	3346#	3360	3468#	3516	3739#	3755	3759#	3789	3927#	3967	3968#
	4219#	4230	4236#	4267	4363#	4387						
\$TSK2 050017	1390#	1393	1461#	1464	1744#	1752	2395#	2411	2412#	2444	2923#	2932
	2948	3031#	3074	3075#	3121	3169#	3220	3221#	3246	3334#	3337	3766#
	3974#	3991	3992#	4009	4244#	4265						
\$TSK3 = 050034	2433#	2442	3048#	3051	3063#	3066	3084#	3111	3112#	3119	3209#	3212
	3244											3234#
\$TSK4 - 050024	3101#	3104										
\$SARGC= 000000	674#	1571#	1673#	1787#	1908#	1982#	2116#	2328#	2490#	2679#	3276#	3423#
	3842#	3901#	4334#									3446#
\$SBYTE- 000402	674#	1094#	1230#	1388#	1459#	1593#	1606#	1649#	1674#	1721#	1723#	1941#
	2151#	2211#	2231#	2279#	2339#	2377#	2393#	2431#	2567#	2592#	2608#	2718#
	2890#	2899#	2921#	3000#	3014#	3021#	3023#	3029#	3046#	3061#	3079#	3081#
	3145#	3167#	3207#	3232#	3309#	3318#	3332#	3466#	3658#	3757#	3917#	3925#
	4104#	4234#	4361#									3972#
\$SCASE= 000000	674#											
\$SDST - 000003	674#	1589#	1645#	3475#	3502#	3511#						
\$SELOC- 000402	674#	1094#	1099#	1230#	1235#	1362#	1371	1384#	1388#	1393#	1395#	1397#
	1409#	1455#	1459#	1464#	1468#	1593#	1600#	1606#	1616	1621#	1649#	1654#
	1679#	1683#	1684#	1690#	1743#	1751#	1752#	1831#	1839#	1840#	1850#	1858#
	1874#	1882#	1883#	2130#	2144#	2145#	2151#	2163#	2169#	2170#	2180#	2181#
	2223#	2224#	2231#	2243#	2244#	2279#	2291#	2292#	2339#	2349#	2350	2377#
	2407#	2411	2431#	2441#	2442#	2444#	2450	2464#	2465#	2467#	2519#	2557
	2567#	2579#	2592#	2608#	2642#	2643#	2645	2650#	2651#	2718#	2724#	2748#

SSERFL= 000000
SSFLAG= 000001

SSFROM= 000000

SSLOC = 021362

SSLOCN- 000000

SSREG - 177777

SSRETU= 000001

SSRTNI 050000

2751#	2890#	2899#	2918	2921#	2931	2948#	2950#	2957	2964#	3000#	3010#	3014#
3021#	3029#	3046#	3051#	3061#	3066#	3073	3079#	3099#	3104#	3110	3119#	3121#
3129#	3134	3145#	3161	3167#	3207#	3212#	3219	3232#	3244#	3246#	3248#	3250#
3309#	3318#	3332#	3337#	3344	3359#	3360#	3362	3404#	3405#	3599#	3621#	3635#
3652#	3658#	3671#	3704#	3721#	3738#	3755#	3757#	3765#	3787#	3789#	3806#	3823#
3876#	3882#	3883#	3917#	3925#	3966#	3967	3972#	3990#	3991	4008#	4009#	4011#
4013#	4078#	4095#	4104#	4109	4114#	4186#	4198#	4218#	4230#	4234#	4243#	4265#
4267#	4293#	4310#	4361#	4386#	4387#							
674#												
674#	1094#	1096#	1099#	1230#	1232#	1235#	1362#	1384#	1388#	1390#	1393#	1395#
1397#	1405#	1409#	1455#	1459#	1461#	1464#	1468#	1593#	1595#	1600#	1606#	1608#
1621#	1649#	1651#	1654#	1674#	1676#	1679#	1684#	1690#	1720#	1721#	1723#	1725#
1743#	1752#	1831#	1840#	1850#	1859#	1874#	1883#	1940#	1941#	1943#	1945#	2130#
2145#	2151#	2153#	2163#	2170#	2181#	2211#	2213#	2224#	2231#	2233#	2244#	2279#
2281#	2292#	2339#	2341#	2377#	2379#	2393#	2395#	2431#	2433#	2442#	2444#	2465#
2467#	2519#	2562#	2567#	2569#	2579#	2592#	2594#	2608#	2610#	2643#	2651#	2718#
2720#	2724#	2726#	2749#	2751#	2890#	2892#	2899#	2901#	2921#	2923#	2948#	2950#
2964#	3000#	3002#	3010#	3014#	3016#	3021#	3023#	3025#	3029#	3031#	3046#	3048#
3051#	3061#	3063#	3066#	3079#	3081#	3084#	3099#	3101#	3104#	3119#	3121#	3129#
3145#	3147#	3167#	3169#	3207#	3209#	3212#	3232#	3234#	3244#	3246#	3248#	3250#
3309#	3311#	3318#	3320#	3332#	3334#	3337#	3360#	3405#	3465#	3466#	3468#	3599#
3621#	3635#	3652#	3658#	3660#	3671#	3704#	3721#	3738#	3755#	3757#	3759#	3765#
3787#	3789#	3806#	3823#	3876#	3883#	3917#	3919#	3925#	3927#	3974#	4009#	
4011#	4013#	4078#	4095#	4104#	4106#	4114#	4186#	4198#	4218#	4230#	4234#	4236#
4243#	4265#	4267#	4293#	4310#	4361#	4363#	4387#					
674#	1677#	1710#	1729#	1738#	1757#	1815#	1826#	1845#	1866#	2047#	2070#	2125#
2158#	2187#	2253#	2355#	2514#	3594#	3606#	3626#	3637#	3698#	3706#	3731#	3740#
3760#	3767#	3772#	3800#	3808#	3856#	3871#	4072#	4080#	4154#	4165#	4180#	4188#
4210#	4220#	4237#	4245#	4250#	4278#	4287#	4295#	4300#	4368#			
674#	1095#	1096	1231#	1232	1362#	1363	1384#	1385	1389#	1390	1405#	1406
1439#	1440	1455#	1456	1460#	1461	1471#	1472	1488#	1489	1594#	1595	1607#
1608	1650#	1651	1675#	1676	1679#	1680	1722#	1723	1724#	1725	1743#	1744
1831#	1832	1850#	1851	1874#	1875	1942#	1943	1944#	1945	2130#	2131	2152#
2153	2163#	2164	2212#	2213	2232#	2233	2280#	2281	2340#	2341	2378#	2379
2394#	2395	2432#	2433	2519#	2520	2568#	2569	2586#	2587	2593#	2594	2603#
2604	2609#	2610	2704#	2705	2719#	2720	2725#	2726	2891#	2892	2900#	2901
2922#	2923	3001#	3002	3015#	3016	3022#	3023	3024#	3025	3030#	3031	3047#
3048	3062#	3063	3082#	3083	3100#	3101	3146#	3147	3168#	3169	3208#	3209
3233#	3234	3298#	3299	3310#	3311	3319#	3320	3333#	3334	3467#	3468	3599#
3600	3635#	3636	3659#	3660	3688#	3689	3704#	3705	3738#	3739	3758#	3759
3765#	3766	3795#	3796	3806#	3807	3876#	3877	3918#	3919	3926#	3927	3973#
3974	4078#	4079	4105#	4106	4121#	4122	4186#	4187	4218#	4219	4235#	4236
4243#	4244	4273#	4274	4293#	4294	4362#	4363	4402#	4403			
674#	1571#	1573#	1574#	1577	1658	1673#	1774	1787#	1788#	1789#	1792	1891
1908#	1909#	1910#	1913	1967	1982#	2105	2116#	2309	2328#	2329#	2330#	2333
2475	2490#	2491#	2492#	2495	2656	2679#	2680#	2681#	2684	2842	3276#	3410
3423#	3426	3446#	3447#	3448#	3451	3529	3842#	3888	3901#	4024	4334#	4413
674#	1573#	1658	1788#	1891	1909#	1967	2329#	2475	2491#	2656	2680#	2842
3447#	3529											
674#	1571#	1657	1673#	1681#	1749#	1772	1787#	1837#	1856#	1880#	1889	1908#
1966	1982#	2104	2116#	2142#	2167#	2221#	2241#	2289#	2307	2328#	2347#	2405#
2439#	2462#	2473	2490#	2640#	2648#	2654	2679#	2746#	2840	3276#	3357#	3402#
3408	3423#	3425	3446#	3528	3842#	3880#	3886	3901#	3964#	3988#	4006#	4022
4334#	4384#	4411										
674#	1571#	1656	1673#	1771	1787#	1888	1908#	1965	1982#	2103	2116#	2179
2306	2328#	2472	2490#	2653	2679#	2839	3276#	3407	3423#	3424	3446#	3527

ABORTW	2941	3239													
ALLOCA	1574	1789	1910	2330	2492	2681	3448								
BGNAU	1333														
BGNCLN	1526														
BGNDU	1338														
BGNHRD	4435														
BGNHW	1298														
BGNINI	1346														
BGNMOD	679	724	1294	1323	1552	3566	4431								
BGNMSG	983	1001	1006	1028	1042	1066	1080	1084	1088	1132	1150	1204	1220	1254	1276
	1281														
BGNRPT	1328														
BGNSEG	4279														
BGNSFT	4553														
BGNSRV	2879	2979													
BGNSW	1307														
BGNTST	3576	4056													
BREAK	2599														
CALL	1676	1709	1728	1737	1756	1814	1825	1844	1865	2046	2069	2124	2157	2186	2252
	2354	2513	3593	3605	3625	3636	3697	3705	3730	3739	3759	3766	3771	3799	3807
	3855	3870	4071	4079	4153	4164	4179	4187	4209	4219	4236	4244	4249	4277	4286
	4294	4299	4367												
CKLOOP	3615	3646	3665	3715	3749	3781	3817	4089	4199	4231	4259	4311			
DEALLO	1656#	1658	1771#	1888#	1891	1965#	1967	2103#	2306#	2472#	2475	2653#	2656	2839#	2842
	3407#	3424#	3527#	3529	3885#	4021#	4410#								
DECLAR	1572	1787	1908	2328	2490	2679	3446								
DEVREG	814														
DEVTYP	810														
DISPAT	1318														
ELSE	1370	1615	2349	2410	2449	2556	2644	2917	2930	2956	3072	3109	3133	3160	3218
	3343	3361	3966	3990	4108										
ENDAU	1334														
ENDCLN	1548														
ENDDO	1764	1958	3515												
ENDDU	1339														
ENDHRD	4481														
ENDHW	1305														
ENDIF	1098	1234	1392	1394	1396	1408	1463	1467	1599	1620	1653	1683	1689	1751	1839
	1858	1882	2144	2169	2180	2223	2243	2291	2441	2443	2464	2466	2561	2578	2642
	2650	2748	2750	2947	2949	2963	3009	3050	3065	3103	3118	3120	3128	3211	3243
	3245	3247	3249	3336	3359	3404	3620	3651	3670	3720	3754	3786	3788	3822	3882
	4008	4010	4012	4094	4113	4197	4229	4264	4266	4309	4386				
ENDINI	1508														
ENDMOD	723	1293	1322	1551	3565	4430	4598								
ENDMSG	991	1002	1015	1038	1053	1075	1081	1085	1115	1140	1186	1211	1245	1263	1277
	1282														
ENDRPT	1329														
ENDRTN	1655	1770	1887	1964	2102	2305	2471	2652	2838	3406	3423	3526	3884	4020	4409
ENDSEG	4313														
ENDSFT	4571														
ENDSRV	2968	3256													
ENDSW	1312														
ENDTST	3827	4319													
EQUALS	728														
ERRDF	2569														
ERRHRD	1744	1832	1851	1875	2131	2213	2233	2281	2341	2399	2433	2456	2546	2634	2740

ASSEMBLY ROUTINES		MACY11 30A(1052)		29-DEC-78 11:50		PAGE 125		G 10		CROSS REFERENCE TABLE -- MACRO NAMES						SEQ 0123
DOCTOR.P11 17-FEB-78 09:08																
	3349	3393	3610	3641	3660	3710	3744	3776	3812	3953	3977	3995	4084	4192	4224	
	4254	4304														
ERRSOF	4372															
EXIT	3617	3648	3667	3717	3751	3783	3819	4091	4261							
GPHARD	1447															
GPRMA	4437	4442														
GPRMD	4447	4453	4459	4465	4471	4555	4565									
GPRML	4477	4561														
HEADER	680															
IF	1093	1229	1387	1458	1592	1605	1648	1673	2150	2210	2278	2338	2376	2392	2430	
	2566	2591	2607	2717	2723	2889	2898	2920	2999	3013	3045	3060	3098	3144	3166	
	3206	3231	3308	3317	3331	3657	3756	4103	4233							
IFB	2230	3020	3028	3078	3916	3924	3971	4360								
IF.ERR	1361	1404	1678	1742	1830	1849	1873	2129	2162	2518	3598	3634	3703	3737	3764	
	3805	3875	4077	4185	4217	4242	4292									
IF.NO.	1383	1454														
LASTAD	4599															
LET	1091	1096	1100	1154	1227	1232	1236	1366	1368	1385	1390	1406	1414	1416	1421	
	1423	1425	1427	1429	1433	1435	1445	1456	1461	1465	1476	1480	1482	1484	1492	
	1495	1497	1499	1501	1532	1534	1536	1543	1581	1587	1595	1597	1608	1610	1618	
	1626	1628	1630	1635	1637	1643	1651	1681#	1685	1687	1694	1696	1699	1702	1704	
	1749#	1798	1800	1805	1808	1837#	1856#	1863	1880#	1917	1923	1928	1934	1948	1950	
	1952	1954	1956	1989	1995	2000	2002	2004	2006	2011	2013	2018	2023	2025	2030	
	2033	2038	2041	2057	2062	2064	2080	2083	2085	2087	2092	2095	2142#	2153	2155	
	2167#	2171	2174	2179#	2197	2200	2202	2204	2221#	2241#	2248	2250	2263	2266	2268	
	2270	2272	2289#	2296	2298	2301	2347#	2362	2365	2367	2369	2371	2382	2385	2387	
	2395	2397	2405#	2412	2415	2417	2419	2422	2424	2439#	2452	2454	2462#	2499	2501	
	2503	2527	2529	2531	2533	2535	2537	2551	2559	2574	2576	2618	2620	2622	2624	
	2626	2628	2640#	2648#	2693	2698	2708	2710	2712	2726	2728	2730	2732	2734	2746#	
	2757	2760	2762	2764	2766	2772	2775	2777	2779	2789	2791	2804	2807	2813	2815	
	2821	2823	2825	2827	2836	2904	2906	2908	2910	2912	2926	2928	2936	2939	2943	
	2945	2959	2961	2994	3002	3004	3011	3035	3037	3039	3041	3043	3048	3052	3054	
	3056	3058	3063	3070	3088	3090	3092	3094	3096	3101	3105	3107	3116	3126	3139	
	3152	3154	3156	3158	3175	3177	3179	3181	3183	3191	3193	3199	3204	3209	3213	
	3225	3227	3234	3241	3279	3289	3303	3315	3323	3325	3327	3329	3334	3341	3354	
	3357#	3367	3371	3376	3378	3380	3382	3384	3402#	3458	3460	3462	3468	3470	3472	
	3474	3479	3481	3483	3485	3487	3489	3491	3495	3497	3499	3501	3506	3508	3510	
	3518	3580	3585	3588	3675	3677	3684	3692	3695	3846	3848	3850	3866	3868	3877	
	3880#	3910	3922	3935	3937	3939	3941	3943	3945	3947	3949	3951	3964#	3988#	4006#	
	4060	4062	4064	4069	4101	4106	4111	4117	4125	4128	4133	4136	4143	4148	4175	
	4177	4284	4345	4347	4349	4363	4365	4384#	4391	4393	4395					
LOCAL	1573	1788	1909	2329	2491	2680	3447									
MSBYTE	681#	686	687	688												
MSCHEC	3618#	3649#	3668#	3718#	3752#	3784#	3820#	4092#	4262#							
MSCKID	703#															
MSCOUN	985#	1008#	1031#	1046#	1068#	1103#	1110#	1134#	1157#	1164#	1171#	1178#	1206#	1222#	1239#	
	1256#															
MSDATA	681#	689	690	691	692	693	694	695	696	697	698	699	700	701	702	
	703#	704	705	706	708	709	710	712	713	714	715	716	717	718	719	
	720	721	722	811#												
MSDECR	724#	992#	1003#	1016#	1039#	1054#	1076#	1082#	1086#	1116#	1141#	1187#	1212#	1246#	1264#	
	1278#	1283#	1294#	1306#	1313#	1323#	1330#	1335#	1340#	1509#	1549#	1552#	2969#	3257#	3566#	
	3828#	4314#	4320#	4431#	4482#	4572#	4599#									
MSDEFA	4438#	4443#	4448#	4454#	4460#	4466#	4472#	4478#	4556#	4562#	4566#					
MSENDE	724#	992#	1003#	1016#	1039#	1054#	1076#	1082#	1086#	1116#	1141#	1187#	1212#	1246#	1264#	
	1278#	1283#	1294#	1306#	1313#	1323#	1330#	1335#	1340#	1509#	1549#	1552#	2969#	3257#	3566#	

MSGNTA	992#	1003#	1016#	1039#	1054#	1076#	1082#	1086#	1116#	1141#	1187#	1212#	1246#	1264#	1278#
	1283#	1306#	1313#	1330#	1335#	1340#	1509#	1549#	2969#	3257#	3828#	4320#	4482#	4483#	4572#
MSGNTE	4573#	4057#													
MSHAPT	681#														
MSHAP	681#	703													
MSINCR	680#	725#	984#	989#	993#	1002#	1004#	1007#	1013#	1017#	1029#	1036#	1040#	1043#	1051#
	1055#	1067#	1073#	1077#	1081#	1083#	1085#	1087#	1089#	1107#	1113#	1117#	1133#	1138#	1142#
	1151#	1161#	1168#	1175#	1182#	1188#	1205#	1209#	1213#	1221#	1225#	1243#	1247#	1255#	1261#
	1265#	1277#	1279#	1282#	1284#	1295#	1299#	1308#	1324#	1329#	1331#	1334#	1336#	1339#	1341#
	1347#	1357#	1382#	1403#	1449#	1510#	1527#	1550#	1553#	1745#	1833#	1852#	1876#	2132#	2214#
	2234#	2282#	2342#	2400#	2434#	2457#	2547#	2570#	2600#	2635#	2722#	2741#	2787#	2799#	2880#
	2885#	2942#	2980#	2985#	3240#	3288#	3350#	3394#	3567#	3577#	3611#	3616#	3618#	3642#	3647#
	3649#	3661#	3666#	3668#	3683#	3711#	3716#	3718#	3745#	3750#	3752#	3777#	3782#	3784#	3813#
	3818#	3820#	3829#	3954#	3978#	3996#	4057#	4085#	4090#	4092#	4193#	4200#	4225#	4232#	4255#
	4260#	4262#	4280#	4305#	4312#	4315#	4321#	4373#	4432#	4436#	4554#				
MSLDRO	1356#	1381#	1402#	1448#	2721#	2884#	2984#	3287#	3682#						
MSMASK	815#	816													
MSMCHI	673#														
MSMCLO	673#														
MSMSK1	815#														
MSPOP	724#	992#	1003#	1016#	1039#	1054#	1076#	1082#	1086#	1116#	1141#	1187#	1212#	1246#	1264#
	1278#	1283#	1294#	1306#	1313#	1323#	1330#	1335#	1340#	1509#	1549#	1552#	2969#	3257#	3566#
	3828#	4314#	4320#	4431#	4482#	4572#	4599#								
MSPRIN	985#	1008#	1031#	1046#	1068#	1103#	1110#	1134#	1157#	1164#	1171#	1178#	1206#	1222#	1239#
	1256#														
MSPUSH	680#	725#	984#	1002#	1007#	1029#	1043#	1067#	1081#	1085#	1089#	1133#	1151#	1205#	1221#
	1255#	1277#	1282#	1295#	1299#	1308#	1324#	1329#	1334#	1339#	1347#	1527#	1553#	2880#	2980#
	3567#	3577#	4057#	4280#	4432#	4436#	4554#								
MSPUT	985#	1008#	1031#	1046#	1068#	1103#	1110#	1134#	1157#	1164#	1171#	1178#	1206#	1222#	1239#
	1256#	2783#	2795#												
MSPUT1	985#	986	987	1008#	1009	1010	1011	1031#	1032	1033	1034	1046#	1047	1048	1049
	1068#	1069	1070	1071	1103#	1104	1105	1110#	1111	1134#	1135	1136	1157#	1158	1159
	1164#	1165	1166	1171#	1172	1173	1178#	1179	1180	1206#	1207	1222#	1223	1239#	1240
	1241	1256#	1257	1258	1259	2783#	2784	2785	2786	2795#	2796	2797	2798		
MSRADI	4438#	4443#	4448#	4454#	4460#	4466#	4472#	4478#	4556#	4562#	4566#				
MSRNRO	1448#	1450													
MSSETS	680#	725#	984#	1002#	1007#	1029#	1043#	1067#	1081#	1085#	1089#	1133#	1151#	1205#	1221#
	1255#	1277#	1282#	1295#	1299#	1308#	1324#	1329#	1334#	1339#	1347#	1527#	1553#	2880#	2980#
	3567#	3577#	4057#	4280#	4432#	4436#	4554#								
MS SVC	985#	989	992#	993	1003#	1004	1008#	1013	1016#	1017	1031#	1036	1039#	1040	1046#
	1051	1054#	1055	1068#	1073	1076#	1077	1082#	1083	1086#	1087	1103#	1107	1110#	1113
	1116#	1117	1134#	1138	1141#	1142	1157#	1161	1164#	1168	1171#	1175	1178#	1182	1187#
	1188	1206#	1209	1212#	1213	1222#	1225	1239#	1243	1246#	1247	1256#	1261	1264#	1265
	1278#	1279	1283#	1284	1330#	1331	1335#	1336	1340#	1341	1356#	1357	1381#	1382	1402#
	1403	1448#	1449	1509#	1510	1549#	1550	2600#	2721#	2722	2783#	2787	2795#	2799	2884#
	2885	2942#	2984#	2985	3240#	3287#	3288	3616#	3618#	3647#	3649#	3666#	3668#	3682#	3683
	3716#	3718#	3750#	3752#	3782#	3784#	3818#	3820#	3828#	3829	4090#	4092#	4200#	4232#	4260#
	4262#	4280#	4312#	4314#	4315	4320#	4321								
MS TLAB	989#	993#	1004#	1013#	1017#	1036#	1040#	1051#	1055#	1073#	1077#	1083#	1087#	1107#	1113#
	1117#	1138#	1142#	1161#	1168#	1175#	1182#	1188#	1209#	1213#	1225#	1243#	1247#	1261#	1265#
	1279#	1284#	1331#	1336#	1341#	1357#	1382#	1403#	1449#	1510#	1550#	1745#	1833#	1852#	1876#
	2132#	2214#	2234#	2282#	2342#	2400#	2434#	2457#	2547#	2570#	2600#	2635#	2722#	2741#	2787#
	2799#	2885#	2942#	2985#	3240#	3288#	3350#	3394#	3611#	3616#	3618#	3642#	3647#	3649#	3661#
	3666#	3668#	3683#	3711#	3716#	3718#	3745#	3750#	3752#	3777#	3782#	3784#	3813#	3818#	3820#
	3829#	3954#	3978#	3996#	4085#	4090#	4092#	4193#	4200#	4225#	4232#	4255#	4260#	4262#	4280#

MSTST.	4305#	4312#	4315#	4321#	4373#													
	989#	993#	1004#	1013#	1017#	1036#	1040#	1051#	1055#	1073#	1077#	1083#	1087#	1107#	1113#			
	1117#	1138#	1142#	1161#	1168#	1175#	1182#	1188#	1209#	1213#	1225#	1243#	1247#	1261#	1265#			
	1279#	1284#	1331#	1336#	1341#	1357#	1382#	1403#	1449#	1510#	1550#	1745#	1833#	1852#	1876#			
	2132#	2214#	2234#	2282#	2342#	2400#	2434#	2457#	2547#	2570#	2600#	2635#	2722#	2741#	2787#			
	2799#	2885#	2942#	2985#	3240#	3288#	3350#	3394#	3611#	3616#	3618#	3642#	3647#	3649#	3661#			
	3666#	3668#	3683#	3711#	3716#	3718#	3745#	3750#	3752#	3777#	3782#	3784#	3813#	3818#	3820#			
	3829#	3954#	3978#	3996#	4085#	4090#	4092#	4193#	4200#	4225#	4232#	4255#	4260#	4262#	4280#			
	4305#	4312#	4315#	4321#	4373#													
	MSWORD	703#	711	815#	816#	817	1319#	1320	1321	1745#	1746	1747	1833#	1834	1835	1852#		
		1853	1854	1876#	1877	1878	2132#	2133	2134	2135	2214#	2215	2216	2217	2234#	2235		
		2236	2237	2282#	2283	2284	2285	2342#	2343	2344	2345	2400#	2401	2402	2403	2434#		
		2435	2436	2437	2457#	2458	2459	2460	2547#	2548	2549	2550	2570#	2571	2572	2573		
		2635#	2636	2637	2638	2741#	2742	2743	2744	3350#	3351	3352	3353	3394#	3395	3396		
		3397	3611#	3612	3613	3614	3618#	3642#	3643	3644	3645	3649#	3661#	3662	3663	3664		
	3668#	3711#	3712	3713	3714	3718#	3745#	3746	3747	3748	3752#	3777#	3778	3779	3780			
	3784#	3813#	3814	3815	3816	3820#	3954#	3955	3956	3978#	3979	3980	3996#	3997	3998			
	4085#	4086	4087	4088	4092#	4193#	4194	4195	4196	4225#	4226	4227	4228	4255#	4256			
	4257	4258	4262#	4305#	4306	4307	4308	4373#	4374	4375	4376	4438#	4443#	4448#	4454#			
	4460#	4466#	4472#	4478#	4556#	4562#	4566#											
POINTE POP	678																	
	1184	1186	1506	1508	1538	1540	2098	2100	2102	2139	2141	2164	2166	2176	2178			
	2218	2220	2238	2240	2286	2288	2303	2305	2539	2541	2543	2630	2632	2634	2736			
	2738	2740	2829	2831	2833	2951	2953	3006	3008	3185	3187	3254	3256	3386	3388			
	3390	3520	3522	3524	3526	3957	3959	3961	3963	3981	3983	3985	3987	3999	4001			
	4003	4005	4014	4016	4018	4020	4138	4140	4377	4379	4381	4383	4403	4405	4407			
	4409																	
PRINTX	984	1007	1030	1045	1067	1102	1109	1133	1156	1163	1170	1177	1205	1221	1238			
	1255																	
PUSH	1152	1154	1350	1352	1530	1532	1982	1984	1986	2119	2121	2520	2522	2524	2610			
	2612	2614	2686	2688	2690	2896	2898	2986	2988	3173	3175	3369	3371	3374	3376			
	3452	3454	3456	3458	3901	3903	3905	3907	4099	4101	4334	4336	4338	4340				
READEF	1355	1380	1401															
REPEAT	1431	1443	1478	2508	2597	2700	3284	3679	3725	4115	4204	4354						
RETURN	1680	1748	1836	1855	1879	2141	2166	2178	2220	2240	2288	2346	2404	2438	2461			
	2639	2647	2745	3356	3401	3879	3963	3987	4005	4383								
ROUTIN	1569	1671	1785	1906	1980	2114	2326	2488	2677	3274	3421	3444	3840	3899	4332			
SETPRI	2883	2983																
SETVEC	2782	2794																
STRUCT	673#																	
SVC	673#																	
UNTIL	1437	1469	1486	2584	2601	2702	3294	3686	3793	4119	4271	4400						
WAITUS	2720	3286	3681															
WHILE	1719	1939	3464															
XFER	3618#	3649#	3668#	3718#	3752#	3784#	3820#	4092#	4262#									
SADDON	1094#	1096#	1230#	1232#	1362#	1363#	1371#	1373#	1384#	1385#	1388#	1390#	1405#	1406#	1432#			
	1433#	1444#	1445#	1455#	1456#	1459#	1461#	1479#	1480#	1570#	1571#	1593#	1595#	1606#	1608#			
	1616#	1618#	1649#	1651#	1672#	1673#	1674#	1676#	1679#	1680#	1720#	1721#	1723#	1725#	1743#			
	1744#	1765#	1786#	1787#	1831#	1832#	1850#	1851#	1874#	1875#	1907#	1908#	1940#	1941#	1943#			
	1945#	1959#	1981#	1982#	2115#	2116#	2130#	2131#	2151#	2153#	2163#	2164#	2211#	2213#	2231#			
	2233#	2279#	2281#	2327#	2328#	2339#	2341#	2350#	2351#	2377#	2379#	2393#	2395#	2411#	2412#			
	2431#	2433#	2450#	2452#	2489#	2490#	2509#	2510#	2519#	2520#	2557#	2559#	2567#	2569#	2592#			
	2594#	2598#	2599#	2608#	2610#	2645#	2647#	2678#	2679#	2701#	2702#	2718#	2720#	2724#	2726#			
	2890#	2892#	2899#	2901#	2918#	2920#	2921#	2923#	2931#	2933#	2957#	2959#	3000#	3002#	3014#			
	3016#	3021#	3023#	3025#	3029#	3031#	3046#	3048#	3061#	3063#	3073#	3075#	3079#	3081#	3084#			
	3099#	3101#	3110#	3112#	3134#	3136#	3145#	3147#	3161#	3163#	3167#	3169#	3207#	3209#	3219#			

	3000#	3014#	3021#	3023	3029#	3046#	3061#	3079#	3081	3099#	3145#	3167#	3207#	3232#	3309#
	3318#	3332#	3465#	3466	3658#	3757#	3917#	3925#	3972#	4104#	4234#	4361#			
\$COMPA	1094#	1230#	1362#	1384#	1388#	1405#	1455#	1459#	1593#	1606#	1649#	1674#	1679#	1721#	1743#
	1831#	1850#	1874#	1941#	2130#	2151#	2163#	2211#	2231#	2279#	2339#	2377#	2393#	2431#	2519#
	2567#	2592#	2608#	2718#	2724#	2890#	2899#	2921#	3000#	3014#	3021#	3029#	3046#	3061#	3079#
	3099#	3145#	3167#	3207#	3232#	3309#	3318#	3332#	3466#	3599#	3635#	3658#	3704#	3738#	3757#
\$COUNT	3765#	3806#	3876#	3917#	3925#	3972#	4078#	4104#	4186#	4218#	4234#	4243#	4293#	4361#	
	1677#	1710#	1729#	1738#	1757#	1815#	1826#	1845#	1866#	2047#	2070#	2125#	2158#	2187#	2253#
	2355#	2514#	3594#	3606#	3626#	3637#	3698#	3706#	3731#	3740#	3760#	3767#	3772#	3800#	3808#
	3856#	3871#	4072#	4080#	4154#	4165#	4180#	4188#	4210#	4220#	4237#	4245#	4250#	4278#	4287#
	4295#	4300#	4368#												
\$DO	1094#	1230#	1388#	1459#	1593#	1606#	1649#	1674#	1721#	1723#	1941#	1943#	2151#	2211#	2231#
	2279#	2339#	2377#	2393#	2431#	2567#	2592#	2608#	2718#	2724#	2890#	2899#	2921#	3000#	3014#
	3021#	3023#	3029#	3046#	3061#	3079#	3081#	3099#	3145#	3167#	3207#	3232#	3309#	3318#	3332#
	3466#	3658#	3757#	3917#	3925#	3972#	4104#	4234#	4361#						
\$ELSE	1371#	1616#	2350#	2411#	2450#	2557#	2645#	2918#	2931#	2957#	3073#	3110#	3134#	3161#	3219#
	3344#	3362#	3967#	3991#	4109#										
\$ERRMS	1092#	1094#	1095#	1097#	1099#	1101#	1155#	1228#	1230#	1231#	1233#	1235#	1237#	1362#	1367#
	1369#	1371#	1372#	1384#	1386#	1388#	1389#	1391#	1393#	1395#	1397#	1405#	1407#	1409#	1415#
	1417#	1422#	1424#	1426#	1428#	1430#	1434#	1436#	1438#	1439#	1446#	1455#	1457#	1459#	1460#
	1462#	1464#	1466#	1468#	1470#	1471#	1477#	1481#	1483#	1485#	1487#	1488#	1493#	1494#	1496#
	1498#	1500#	1502#	1533#	1535#	1537#	1544#	1573#	1575#	1582#	1588#	1589#	1593#	1594#	1596#
	1598#	1600#	1606#	1607#	1609#	1611#	1616#	1617#	1619#	1621#	1627#	1629#	1631#	1636#	1638#
	1644#	1645#	1649#	1650#	1652#	1654#	1656#	1674#	1675#	1677#	1679#	1681#	1684#	1686#	1688#
	1690#	1695#	1697#	1698#	1700#	1703#	1705#	1710#	1720#	1721#	1722#	1723#	1724#	1729#	1738#
	1743#	1749#	1752#	1757#	1765#	1766#	1771#	1788#	1790#	1799#	1801#	1806#	1807#	1809#	1815#
	1826#	1831#	1837#	1840#	1845#	1850#	1856#	1859#	1864#	1866#	1874#	1880#	1883#	1888#	1909#
	1911#	1918#	1919#	1924#	1929#	1930#	1935#	1940#	1941#	1942#	1943#	1944#	1949#	1951#	1953#
	1955#	1957#	1959#	1960#	1965#	1990#	1991#	1996#	2001#	2003#	2005#	2007#	2012#	2014#	2019#
	2024#	2026#	2031#	2032#	2034#	2039#	2040#	2042#	2047#	2058#	2063#	2065#	2070#	2081#	2082#
	2084#	2086#	2088#	2093#	2094#	2096#	2097#	2103#	2125#	2130#	2142#	2145#	2151#	2152#	2154#
	2156#	2158#	2163#	2167#	2170#	2172#	2173#	2175#	2179#	2181#	2187#	2198#	2199#	2201#	2203#
	2205#	2211#	2212#	2221#	2224#	2231#	2232#	2241#	2244#	2249#	2251#	2253#	2264#	2265#	2267#
	2269#	2271#	2273#	2279#	2280#	2289#	2292#	2297#	2299#	2300#	2302#	2306#	2329#	2331#	2339#
	2340#	2347#	2350#	2355#	2363#	2364#	2366#	2368#	2370#	2372#	2377#	2378#	2383#	2384#	2386#
	2388#	2393#	2394#	2396#	2398#	2405#	2411#	2413#	2414#	2416#	2418#	2420#	2421#	2423#	2425#
	2431#	2432#	2439#	2442#	2444#	2450#	2451#	2453#	2455#	2462#	2465#	2467#	2472#	2491#	2493#
	2500#	2502#	2504#	2514#	2519#	2528#	2530#	2532#	2534#	2536#	2538#	2552#	2557#	2558#	2560#
	2562#	2567#	2568#	2575#	2577#	2579#	2585#	2586#	2592#	2593#	2602#	2603#	2608#	2609#	2619#
	2621#	2623#	2625#	2627#	2629#	2640#	2643#	2645#	2646#	2648#	2651#	2653#	2680#	2682#	2694#
	2699#	2703#	2704#	2709#	2711#	2713#	2718#	2719#	2724#	2725#	2727#	2729#	2731#	2733#	2735#
	2746#	2749#	2751#	2758#	2759#	2761#	2763#	2765#	2767#	2773#	2774#	2776#	2778#	2780#	2781#
	2790#	2792#	2793#	2805#	2806#	2808#	2809#	2814#	2816#	2822#	2824#	2826#	2828#	2837#	2839#
	2890#	2891#	2899#	2900#	2905#	2907#	2909#	2911#	2913#	2918#	2919#	2921#	2922#	2927#	2929#
	2931#	2932#	2937#	2938#	2940#	2944#	2946#	2948#	2950#	2957#	2958#	2960#	2962#	2964#	2995#
	3000#	3001#	3003#	3005#	3010#	3012#	3014#	3015#	3021#	3022#	3023#	3024#	3029#	3030#	3036#
	3038#	3040#	3042#	3044#	3046#	3047#	3049#	3051#	3053#	3055#	3057#	3059#	3061#	3062#	3064#
	3066#	3071#	3073#	3074#	3079#	3081#	3082#	3089#	3091#	3093#	3095#	3097#	3099#	3100#	3102#
	3104#	3106#	3108#	3110#	3111#	3117#	3119#	3121#	3127#	3129#	3134#	3135#	3140#	3145#	3146#
	3153#	3155#	3157#	3159#	3161#	3162#	3167#	3168#	3176#	3178#	3180#	3182#	3184#	3192#	3194#
	3200#	3205#	3207#	3208#	3210#	3212#	3214#	3219#	3220#	3226#	3228#	3232#	3233#	3235#	3242#
	3244#	3246#	3248#	3250#	3280#	3290#	3295#	3297#	3298#	3304#	3309#	3310#	3316#	3318#	3319#
	3324#	3326#	3328#	3330#	3332#	3333#	3335#	3337#	3342#	3344#	3345#	3355#	3357#	3360#	3362#
	3363#	3368#	3372#	3373#	3377#	3379#	3381#	3383#	3385#	3402#	3405#	3407#	3424#	3447#	3449#
	3459#	3461#	3463#	3465#	3466#	3467#	3469#	3471#	3473#	3475#	3480#	3482#	3484#	3486#	3488#
	3490#	3492#	3496#	3498#	3500#	3502#	3507#	3509#	3511#	3516#	3517#	3519#	3527#	3581#	3586#

	3587#	3589#	3594#	3599#	3606#	3621#	3626#	3635#	3637#	3652#	3658#	3659#	3671#	3676#	3678#
	3685#	3687#	3688#	3693#	3694#	3696#	3698#	3704#	3706#	3721#	3731#	3738#	3740#	3755#	3757#
	3758#	3760#	3765#	3767#	3772#	3787#	3789#	3794#	3795#	3800#	3806#	3808#	3823#	3847#	3849#
	3851#	3856#	3867#	3869#	3871#	3876#	3878#	3880#	3883#	3885#	3911#	3912#	3917#	3918#	3923#
	3925#	3926#	3936#	3938#	3940#	3942#	3944#	3946#	3948#	3950#	3952#	3964#	3967#	3972#	3973#
	3988#	3991#	4006#	4009#	4011#	4013#	4021#	4061#	4063#	4065#	4070#	4072#	4078#	4080#	4095#
	4102#	4104#	4105#	4107#	4109#	4110#	4112#	4114#	4118#	4120#	4121#	4126#	4127#	4129#	4134#
	4135#	4137#	4144#	4149#	4154#	4165#	4176#	4178#	4180#	4186#	4188#	4198#	4210#	4218#	4220#
	4230#	4234#	4235#	4237#	4243#	4245#	4250#	4265#	4267#	4272#	4273#	4278#	4285#	4287#	4293#
	4295#	4300#	4310#	4346#	4348#	4350#	4361#	4362#	4364#	4366#	4368#	4384#	4387#	4392#	4394#
	4396#	4401#	4402#	4410#											
\$GENBR	1095#	1231#	1362#	1371#	1384#	1389#	1405#	1439#	1455#	1460#	1471#	1488#	1594#	1607#	1616#
	1650#	1675#	1679#	1682#	1722#	1724#	1743#	1750#	1765#	1831#	1838#	1850#	1857#	1874#	1881#
	1942#	1944#	1959#	2130#	2143#	2152#	2163#	2168#	2179#	2212#	2222#	2232#	2242#	2280#	2290#
	2340#	2348#	2378#	2394#	2406#	2432#	2440#	2450#	2463#	2519#	2557#	2568#	2586#	2593#	2603#
	2609#	2641#	2645#	2649#	2704#	2719#	2725#	2747#	2891#	2900#	2918#	2922#	2931#	2957#	3001#
	3015#	3022#	3024#	3030#	3047#	3062#	3073#	3080#	3082#	3100#	3110#	3134#	3146#	3161#	3168#
	3208#	3219#	3233#	3296#	3298#	3310#	3319#	3333#	3344#	3358#	3362#	3403#	3467#	3516#	3599#
	3635#	3659#	3688#	3704#	3738#	3758#	3765#	3795#	3806#	3876#	3881#	3918#	3926#	3965#	3973#
\$GENTA	3989#	4007#	4078#	4105#	4109#	4121#	4186#	4218#	4235#	4243#	4273#	4293#	4362#	4385#	4402#
	1094#	1099#	1230#	1235#	1371#	1372	1388#	1393#	1395#	1397#	1409#	1432#	1444#	1459#	1464#
	1468#	1479#	1593#	1600#	1606#	1616#	1617	1621#	1649#	1654#	1656#	1657	1674#	1684#	1690#
	1720#	1721#	1723#	1752#	1765#	1766	1771#	1773	1840#	1859#	1883#	1888#	1890	1940#	1941#
	1943#	1959#	1960	1965#	1966	2103#	2104	2145#	2151#	2170#	2181#	2211#	2224#	2231#	2244#
	2279#	2292#	2306#	2308	2339#	2350#	2377#	2393#	2411#	2431#	2442#	2444#	2450#	2451	2465#
	2467#	2472#	2474	2509#	2557#	2558	2562#	2567#	2579#	2592#	2598#	2608#	2643#	2645#	2646
	2651#	2653#	2655	2701#	2718#	2724#	2749#	2751#	2839#	2841	2890#	2899#	2918#	2919	2921#
	2931#	2932	2948#	2950#	2957#	2958	2964#	3000#	3010#	3014#	3021#	3023#	3029#	3046#	3051#
	3061#	3066#	3073#	3074	3079#	3081#	3083	3099#	3104#	3110#	3111	3119#	3121#	3129#	3134#
	3135	3145#	3161#	3162	3167#	3207#	3212#	3219#	3220	3232#	3244#	3246#	3248#	3250#	3285#
	3295#	3299	3309#	3318#	3332#	3337#	3344#	3345	3360#	3362#	3363	3405#	3407#	3409	3424#
	3425	3465#	3466#	3516#	3517	3527#	3528	3621#	3652#	3658#	3671#	3680#	3721#	3726#	3755#
	3757#	3787#	3789#	3823#	3883#	3885#	3887	3917#	3925#	3967#	3972#	3991#	4009#	4011#	4013#
	4021#	4023	4095#	4104#	4109#	4110	4114#	4116#	4198#	4205#	4230#	4234#	4265#	4267#	4310#
	4355#	4361#	4387#	4410#	4412										
\$IF	1094#	1230#	1388#	1459#	1593#	1606#	1649#	1674#	2151#	2211#	2231#	2279#	2339#	2377#	2393#
	2431#	2567#	2592#	2608#	2718#	2724#	2890#	2899#	2921#	3000#	3014#	3021#	3029#	3046#	3061#
	3079#	3099#	3145#	3167#	3207#	3232#	3309#	3318#	3332#	3658#	3757#	3917#	3925#	3972#	4104#
	4234#	4361#													
\$IFCOD	1094#	1230#	1388#	1438#	1459#	1470#	1487#	1593#	1606#	1649#	1674#	1721#	1723#	1941#	1943#
	2151#	2211#	2231#	2279#	2339#	2377#	2393#	241#	2567#	2585#	2592#	2602#	2608#	2703#	2718#
	2724#	2890#	2899#	2921#	3000#	3014#	3021#	3023#	3029#	3046#	3061#	3081#	3099#	3145#	3167#
	3207#	3232#	3295#	3297#	3309#	3318#	3332#	3466#	3658#	3687#	3757#	3794#	3917#	3925#	3972#
	4104#	4120#	4234#	4272#	4361#	4401#									
\$IFCON	1362#	1384#	1405#	1455#	1679#	1743#	1831#	1850#	1874#	2130#	2163#	2519#	3599#	3635#	3704#
	3738#	3765#	3806#	3876#	4078#	4186#	4218#	4243#	4293#						
\$IFOPR	1094#	1095	1230#	1231	1362#	1384#	1388#	1389	1405#	1438#	1439	1455#	1459#	1460	1470#
	1471	1487#	1488	1593#	1594	1606#	1607	1649#	1650	1674#	1675	1679#	1721#	1722	1723#
	1724	1743#	1831#	1850#	1874#	1941#	1942	1943#	1944	2130#	2151#	2152	2163#	2211#	2212
	2231#	2232	2279#	2280	2339#	2340	2377#	2378	2393#	2394	2431#	2432	2519#	2567#	2568
	2585#	2586	2592#	2593	2602#	2603	2608#	2609	2703#	2704	2718#	2719	2724#	2725	2890#
	2891	2899#	2900	2921#	2922	3000#	3001	3014#	3015	3021#	3022	3023#	3024	3029#	3030
	3046#	3047	3061#	3062	3081#	3082	3099#	3100	3145#	3146	3167#	3168	3207#	3208	3232#
	3233	3297#	3298	3309#	3310	3318#	3319	3332#	3333	3466#	3467	3599#	3635#	3658#	3659
	3687#	3688	3704#	3738#	3757#	3758	3765#	3794#	3795	3806#	3876#	3917#	3918	3925#	3926
	3972#	3973	4078#	4104#	4105	4120#	4121	4186#	4218#	4234#	4235	4243#	4272#	4273	4293#

	3696#	3698#	3704#	3706#	3707#	3708	3709	3731#	3732	3733	3738#	3740#	3741#	3742	3743
	3757#	3758#	3760#	3765#	3767#	3768#	3769	3770	3772#	3773#	3774	3775	3794#	3795#	3800#
	3806#	3808#	3809#	3810	3811	3847#	3849#	3851#	3856#	3857	3858#	3859	3860	3861	3862
	3867#	3869#	3871#	3876#	3878#	3880#	3881#	3885#	3886	3888	3902#	3903#	3904#	3905#	3906#
	3907#	3911#	3912#	3917#	3918#	3923#	3925#	3926#	3936#	3938#	3940#	3942#	3944#	3946#	3948#
	3950#	3952#	3958#	3959#	3960#	3961#	3962#	3963#	3964#	3965#	3972#	3973#	3982#	3983#	3984#
	3985#	3986#	3987#	3988#	3989#	4000#	4001#	4002#	4003#	4004#	4005#	4006#	4007#	4015#	4016#
	4017#	4018#	4019#	4020#	4021#	4022	4024	4061#	4063#	4065#	4070#	4072#	4078#	4080#	4081#
	4082	4083	4100#	4101#	4102#	4104#	4105#	4107#	4109#	4112#	4118#	4120#	4121#	4126#	4127#
	4129#	4134#	4135#	4137#	4139#	4140#	4144#	4149#	4154#	4155	4156#	4157	4158	4159	4160
	4165#	4166	4167#	4168	4169	4170	4171	4176#	4178#	4180#	4186#	4188#	4189#	4190	4191
	4210#	4211	4212	4218#	4220#	4221#	4222	4223	4234#	4235#	4237#	4243#	4245#	4246#	4247
	4248	4250#	4251#	4252	4253	4272#	4273#	4278#	4285#	4287#	4293#	4295#	4296#	4297	4298
	4300#	4301#	4302	4303	4335#	4336#	4337#	4338#	4339#	4340#	4346#	4348#	4350#	4361#	4362#
	4364#	4366#	4368#	4369#	4370	4371	4378#	4379#	4380#	4381#	4382#	4383#	4384#	4385#	4392#
\$OPEQU	4394#	4396#	4401#	4402#	4404#	4405#	4406#	4407#	4408#	4409#	4410#	4411	4413		
	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#
	1596#	1598#	1609#	1611#	1629#	1636#	1638#	1645#	1652#	1698#	1807#	1919#	1930#	1951#	1953#
	1955#	1957#	1991#	2012#	2014#	2024#	2032#	2040#	2082#	2094#	2097#	2173#	2199#	2249#	2251#
	2265#	2267#	2297#	2300#	2364#	2370#	2384#	2388#	2414#	2418#	2421#	2425#	2552#	2575#	2699#
	2713#	2759#	2763#	2765#	2774#	2778#	2781#	2793#	2806#	2809#	2822#	2824#	2826#	2828#	2837#
	2907#	2911#	2913#	2927#	2929#	2938#	2940#	2944#	2946#	2962#	3012#	3042#	3044#	3055#	3059#
	3095#	3097#	3106#	3117#	3127#	3155#	3157#	3159#	3178#	3182#	3184#	3194#	3200#	3205#	3214#
	3226#	3228#	3235#	3290#	3326#	3330#	3373#	3471#	3473#	3475#	3482#	3484#	3486#	3490#	3492#
	3496#	3498#	3502#	3507#	3511#	3587#	3685#	3694#	3878#	3912#	3936#	3946#	3950#	4127#	4135#
	4392#	4394#	4396#												
\$OPNAN	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#
	1596#	1598#	1609#	1611#	1629#	1636#	1638#	1645#	1652#	1698#	1807#	1919#	1930#	1951#	1953#
	1955#	1957#	1991#	2012#	2014#	2024#	2032#	2040#	2082#	2094#	2097#	2173#	2199#	2249#	2251#
	2265#	2267#	2297#	2300#	2364#	2370#	2384#	2388#	2414#	2418#	2421#	2425#	2552#	2575#	2699#
	2713#	2759#	2763#	2765#	2774#	2778#	2781#	2793#	2806#	2809#	2822#	2824#	2826#	2828#	2837#
	2907#	2911#	2913#	2927#	2929#	2938#	2940#	2944#	2946#	2962#	3012#	3042#	3044#	3055#	3059#
	3095#	3097#	3106#	3117#	3127#	3155#	3157#	3159#	3178#	3182#	3184#	3194#	3200#	3205#	3214#
	3226#	3228#	3235#	3290#	3326#	3330#	3373#	3471#	3473#	3475#	3482#	3484#	3486#	3490#	3492#
	3496#	3498#	3502#	3507#	3511#	3587#	3685#	3694#	3878#	3912#	3936#	3946#	3950#	4127#	4135#
	4392#	4394#	4396#												
\$OPNEG	2005#	2086#	2203#	2271#	2368#	2577#	2711#	3180#	3509#						
\$OPNOR	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#
	1596#	1598#	1609#	1611#	1629#	1636#	1638#	1645#	1652#	1698#	1807#	1919#	1930#	1951#	1953#
	1955#	1957#	1991#	2012#	2014#	2024#	2032#	2040#	2082#	2094#	2097#	2173#	2199#	2249#	2251#
	2265#	2267#	2297#	2300#	2364#	2370#	2384#	2388#	2414#	2418#	2421#	2425#	2552#	2575#	2699#
	2713#	2759#	2763#	2765#	2774#	2778#	2781#	2793#	2806#	2809#	2822#	2824#	2826#	2828#	2837#
	2907#	2911#	2913#	2927#	2929#	2938#	2940#	2944#	2946#	2962#	3012#	3042#	3044#	3055#	3059#
	3095#	3097#	3106#	3117#	3127#	3155#	3157#	3159#	3178#	3182#	3184#	3194#	3200#	3205#	3214#
	3226#	3228#	3235#	3290#	3326#	3330#	3373#	3471#	3473#	3475#	3482#	3484#	3486#	3490#	3492#
	3496#	3498#	3502#	3507#	3511#	3587#	3685#	3694#	3878#	3912#	3936#	3946#	3950#	4127#	4135#
	4392#	4394#	4396#												
\$OPNOT	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#
	1596#	1598#	1609#	1611#	1629#	1636#	1638#	1645#	1652#	1698#	1807#	1919#	1930#	1951#	1953#
	1955#	1957#	1991#	2012#	2014#	2024#	2032#	2040#	2082#	2094#	2097#	2173#	2199#	2249#	2251#
	2265#	2267#	2297#	2300#	2364#	2370#	2384#	2388#	2414#	2418#	2421#	2425#	2552#	2575#	2699#
	2713#	2759#	2763#	2765#	2774#	2778#	2781#	2793#	2806#	2809#	2822#	2824#	2826#	2828#	2837#
	2907#	2911#	2913#	2927#	2929#	2938#	2940#	2944#	2946#	2962#	3012#	3042#	3044#	3055#	3059#
	3095#	3097#	3106#	3117#	3127#	3155#	3157#	3159#	3178#	3182#	3184#	3194#	3200#	3205#	3214#
	3226#	3228#	3235#	3290#	3326#	3330#	3373#	3471#	3473#	3475#	3482#	3484#	3486#	3490#	3492#
	3496#	3498#	3502#	3507#	3511#	3587#	3685#	3694#	3878#	3912#	3936#	3946#	3950#	4127#	4135#

	2828#	2837#	2905#	2907#	2909#	2911#	2913#	2927#	2929#	2937#	2940#	2944#	2946#	2960#	2962#
	2995#	3003#	3005#	3012#	3036#	3038#	3040#	3042#	3044#	3049#	3053#	3055#	3057#	3059#	3064#
	3071#	3089#	3091#	3093#	3095#	3097#	3102#	3106#	3108#	3117#	3127#	3140#	3153#	3155#	3157#
	3159#	3176#	3178#	3180#	3182#	3184#	3192#	3194#	3200#	3205#	3210#	3214#	3226#	3228#	3235#
	3242#	3280#	3290#	3304#	3316#	3324#	3326#	3328#	3330#	3335#	3342#	3355#	3368#	3372#	3377#
	3379#	3381#	3383#	3385#	3459#	3461#	3463#	3469#	3471#	3473#	3475#	3480#	3482#	3484#	3486#
	3488#	3490#	3492#	3496#	3498#	3500#	3502#	3507#	3509#	3511#	3519#	3581#	3586#	3589#	3676#
	3678#	3685#	3693#	3696#	3847#	3849#	3851#	3867#	3869#	3878#	3911#	3923#	3936#	3938#	3940#
	3942#	3944#	3946#	3948#	3950#	3952#	4061#	4063#	4065#	4070#	4102#	4107#	4112#	4118#	4126#
	4129#	4134#	4137#	4144#	4149#	4176#	4178#	4285#	4346#	4348#	4350#	4364#	4366#	4392#	4394#
\$OPR2	1092#	1097#	1101#	1155#	1228#	1233#	1237#	1367#	1369#	1386#	1391#	1407#	1415#	1417#	1422#
	1424#	1426#	1428#	1430#	1434#	1436#	1446#	1457#	1462#	1466#	1477#	1481#	1483#	1485#	1493#
	1496#	1498#	1500#	1502#	1533#	1535#	1537#	1544#	1582#	1588#	1596#	1598#	1609#	1611#	1619#
	1627#	1629#	1631#	1636#	1638#	1644#	1652#	1686#	1688#	1695#	1697#	1700#	1703#	1705#	1799#
	1801#	1806#	1809#	1864#	1918#	1924#	1929#	1935#	1949#	1951#	1953#	1955#	1957#	1990#	1996#
	2001#	2003#	2005#	2007#	2012#	2014#	2019#	2024#	2026#	2031#	2034#	2039#	2042#	2058#	2063#
	2065#	2081#	2084#	2086#	2088#	2093#	2096#	2154#	2156#	2172#	2175#	2198#	2201#	2203#	2205#
	2249#	2251#	2264#	2267#	2269#	2271#	2273#	2297#		2302#	2363#	2366#	2368#	2370#	2372#
	2383#	2386#	2388#	2396#	2398#	2413#	2416#	2418#		2423#	2425#	2453#	2455#	2500#	2502#
	2504#	2528#	2530#	2532#	2534#	2536#	2538#	2552#		2575#	2577#	2619#	2621#	2623#	2625#
	2627#	2629#	2694#	2699#	2709#	2711#	2713#	2727#		2731#	2733#	2735#	2758#	2761#	2763#
	2765#	2767#	2773#	2776#	2778#	2780#	2790#	2792#	2805#	2808#	2814#	2816#	2822#	2824#	2826#
	2828#	2837#	2905#	2907#	2909#	2911#	2913#	2927#	2929#	2937#	2940#	2944#	2946#	2960#	2962#
	2995#	3003#	3005#	3012#	3036#	3038#	3040#	3042#	3044#	3049#	3053#	3055#	3057#	3059#	3064#
	3071#	3089#	3091#	3093#	3095#	3097#	3102#	3106#	3108#	3117#	3127#	3140#	3153#	3155#	3157#
	3159#	3176#	3178#	3180#	3182#	3184#	3192#	3194#	3200#	3205#	3210#	3214#	3226#	3228#	3235#
	3242#	3280#	3290#	3304#	3316#	3324#	3326#	3328#	3330#	3335#	3342#	3355#	3368#	3372#	3377#
	3379#	3381#	3383#	3385#	3459#	3461#	3463#	3469#	3471#	3473#	3475#	3480#	3482#	3484#	3486#
	3488#	3490#	3492#	3496#	3498#	3500#	3502#	3507#	3509#	3511#	3519#	3581#	3586#	3589#	3676#
	3678#	3685#	3693#	3696#	3847#	3849#	3851#	3867#	3869#	3878#	3911#	3923#	3936#	3938#	3940#
	3942#	3944#	3946#	3948#	3950#	3952#	4061#	4063#	4065#	4070#	4102#	4107#	4112#	4118#	4126#
	4129#	4134#	4137#	4144#	4149#	4176#	4178#	4285#	4346#	4348#	4350#	4364#	4366#	4392#	4394#
\$OPSHF	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#
	1596#	1598#	1609#	1611#	1629#	1636#	1638#	1645#	1652#	1698#	1807#	1919#	1930#	1951#	1953#
	1955#	1957#	1991#	2012#	2014#	2024#	2032#	2040#	2082#	2094#	2097#	2173#	2199#	2249#	2251#
	2265#	2267#	2297#	2300#	2364#	2370#	2384#	2388#	2414#	2418#	2421#	2425#	2552#	2575#	2699#
	2713#	2759#	2763#	2765#	2774#	2778#	2781#	2793#	2806#	2809#	2822#	2824#	2826#	2828#	2837#
	2907#	2911#	2913#	2927#	2929#	2938#	2940#	2944#	2946#	2962#	3012#	3042#	3044#	3055#	3059#
	3095#	3097#	3106#	3117#	3127#	3155#	3157#	3159#	3178#	3182#	3184#	3194#	3200#	3205#	3214#
	3226#	3228#	3235#	3290#	3326#	3330#	3373#	3471#	3473#	3475#	3482#	3484#	3486#	3490#	3492#
	3496#	3498#	3502#	3507#	3511#	3587#	3685#	3694#	3878#	3912#	3936#	3946#	3950#	4127#	4135#
	4392#	4394#	4396#												
\$OPSUB	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#
	1596#	1598#	1609#	1611#	1629#	1636#	1638#	1645#	1652#	1698#	1807#	1919#	1930#	1951#	1953#
	1955#	1957#	1991#	2012#	2014#	2024#	2032#	2040#	2082#	2094#	2097#	2173#	2199#	2249#	2251#
	2265#	2267#	2297#	2300#	2364#	2370#	2384#	2388#	2414#	2418#	2421#	2425#	2552#	2575#	2699#
	2713#	2759#	2763#	2765#	2774#	2778#	2781#	2793#	2806#	2809#	2822#	2824#	2826#	2828#	2837#
	2907#	2911#	2913#	2927#	2929#	2938#	2940#	2944#	2946#	2962#	3012#	3042#	3044#	3055#	3059#
	3095#	3097#	3106#	3117#	3127#	3155#	3157#	3159#	3178#	3182#	3184#	3194#	3200#	3205#	3214#
	3226#	3228#	3235#	3290#	3326#	3330#	3373#	3471#	3473#	3475#	3482#	3484#	3486#	3490#	3492#
	3496#	3498#	3502#	3507#	3511#	3587#	3685#	3694#	3878#	3912#	3936#	3946#	3950#	4127#	4135#
	4392#	4394#	4396#												
\$OPSWB	2005#	2086#	2203#	2271#	2368#	2577#	2711#	3180#	3509#						
\$OPXOR	1101#	1237#	1369#	1386#	1436#	1446#	1457#	1483#	1485#	1494#	1498#	1500#	1537#	1544#	1589#

SSPOP	3344#	3362#	3967#	3991#	4109#												
	1099#	1235#	1371#	1372	1393#	1395#	1397#	1409#	1438#	1464#	1468#	1470#	1487#	1600#	1616#		
	1617	1621#	1654#	1656#	1684#	1690#	1752#	1765#	1766	1771#	1840#	1859#	1883#	1888#	1959#		
	1960	1965#	2103#	2145#	2170#	2181#	2224#	2244#	2292#	2306#	2350#	2411#	2442#	2444#	2450#		
	2451	2465#	2467#	2472#	2557#	2558	2562#	2579#	2585#	2602#	2643#	2645#	2646	2651#	2653#		
	2703#	2749#	2751#	2839#	2918#	2919	2931#	2932	2948#	2950#	2957#	2958	2964#	3010#	3051#		
	3066#	3073#	3074	3104#	3110#	3111	3119#	3121#	3129#	3134#	3135	3161#	3162	3212#	3219#		
	3220	3244#	3246#	3248#	3250#	3295#	3337#	3344#	3345	3360#	3362#	3363	3405#	3407#	3424#		
	3516#	3517	3527#	3621#	3652#	3671#	3687#	3721#	3755#	3787#	3789#	3794#	3823#	3883#	3885#		
	3967#	3991#	4009#	4011#	4013#	4021#	4095#	4109#	4110	4114#	4120#	4198#	4230#	4265#	4267#		
	4272#	4310#	4387#	4401#	4410#												
SSPUSH	1094#	1096	1230#	1232	1362#	1363	1371#	1373	1384#	1385	1388#	1390	1405#	1406	1432#		
	1433	1444#	1445	1455#	1456	1459#	1461	1479#	1480	1570#	1571	1593#	1595	1606#	1608		
	1616#	1618	1649#	1651	1672#	1673	1674#	1676	1679#	1680	1720#	1721	1725	1743#	1744		
	1765#	1786#	1787	1831#	1832	1850#	1851	1874#	1875	1907#	1908	1940#	1941	1945	1959#		
	1981#	1982	2115#	2116	2130#	2131	2151#	2153	2163#	2164	2211#	2213	2231#	2233	2279#		
	2281	2327#	2328	2339#	2341	2350#	2351	2377#	2379	2393#	2395	2411#	2412	2431#	2433		
	2450#	2452	2489#	2490	2509#	2510	2519#	2520	2557#	2559	2567#	2569	2592#	2594	2598#		
	2599	2608#	2610	2645#	2647	2678#	2679	2701#	2702	2718#	2720	2724#	2726	2890#	2892		
	2899#	2901	2918#	2920	2921#	2923	2931#	2933	2957#	2959	3000#	3002	3014#	3016	3021#		
	3025	3029#	3031	3046#	3048	3061#	3063	3073#	3075	3079#	3084	3099#	3101	3110#	3112		
	3134#	3136	3145#	3147	3161#	3163	3167#	3169	3207#	3209	3219#	3221	3232#	3234	3275#		
	3276	3285#	3286	3309#	3311	3318#	3320	3332#	3334	3344#	3346	3362#	3364	3422#	3423		
	3445#	3446	3465#	3466	3468	3516#	3599#	3600	3635#	3636	3658#	3660	3680#	3681	3704#		
	3705	3726#	3727	3738#	3739	3757#	3759	3765#	3766	3806#	3807	3841#	3842	3876#	3877		
	3900#	3901	3917#	3919	3925#	3927	3967#	3968	3972#	3974	3991#	3992	4078#	4079	4104#		
	4106	4109#	4111	4116#	4117	4186#	4187	4205#	4206	4218#	4219	4234#	4236	4243#	4244		
	4293#	4294	4333#	4334	4355#	4356	4361#	4363									
SSSETS	1094#	1096#	1230#	1232#	1362#	1363#	1373#	1384#	1385#	1388#	1390#	1405#	1406#	1432#	1433#		
	1444#	1445#	1455#	1456#	1459#	1461#	1479#	1480#	1571#	1593#	1595#	1606#	1608#	1618#	1649#		
	1651#	1673#	1674#	1676#	1679#	1680#	1720#	1721#	1725#	1743#	1744#	1765#	1787#	1831#	1832#		
	1850#	1851#	1874#	1875#	1908#	1940#	1941#	1945#	1959#	1982#	2116#	2130#	2131#	2151#	2153#		
	2163#	2164#	2211#	2213#	2231#	2233#	2279#	2281#	2328#	2339#	2341#	2351#	2377#	2379#	2393#		
	2395#	2412#	2431#	2433#	2452#	2490#	2509#	2510#	2519#	2520#	2559#	2567#	2569#	2592#	2594#		
	2598#	2599#	2608#	2610#	2647#	2679#	2701#	2702#	2718#	2720#	2724#	2726#	2890#	2892#	2899#		
	2901#	2920#	2921#	2923#	2933#	2959#	3000#	3002#	3014#	3016#	3021#	3025#	3029#	3031#	3046#		
	3048#	3061#	3063#	3075#	3079#	3084#	3099#	3101#	3112#	3136#	3145#	3147#	3163#	3167#	3169#		
	3207#	3209#	3221#	3232#	3234#	3276#	3285#	3286#	3309#	3311#	3318#	3320#	3332#	3334#	3346#		
	3364#	3423#	3446#	3465#	3466#	3468#	3516#	3599#	3600#	3635#	3636#	3658#	3660#	3680#	3681#		
	3704#	3705#	3726#	3727#	3738#	3739#	3757#	3759#	3765#	3766#	3806#	3807#	3842#	3876#	3877#		
	3901#	3917#	3919#	3925#	3927#	3968#	3972#	3974#	3992#	4078#	4079#	4104#	4106#	4111#	4116#		
	4117#	4186#	4187#	4205#	4206#	4218#	4219#	4234#	4236#	4243#	4244#	4293#	4294#	4334#	4355#		
	4356#	4361#	4363#														

. ABS. 053416 000

ERRORS DETECTED: 0

CZDZCB.BIN,CZDZCB.SEQ/CRF/SOL/NL:TOC=SMAC/ML,SPMAC/ML,CZDZCB.P11
 RUN-TIME: 100 97 8 SECONDS
 RUN-TIME RATIO: 531/207=2.5
 CORE USED: 33k (65 PAGES)