

M8203

M8203 STATIC DIAG. # 2
CZDMSB0

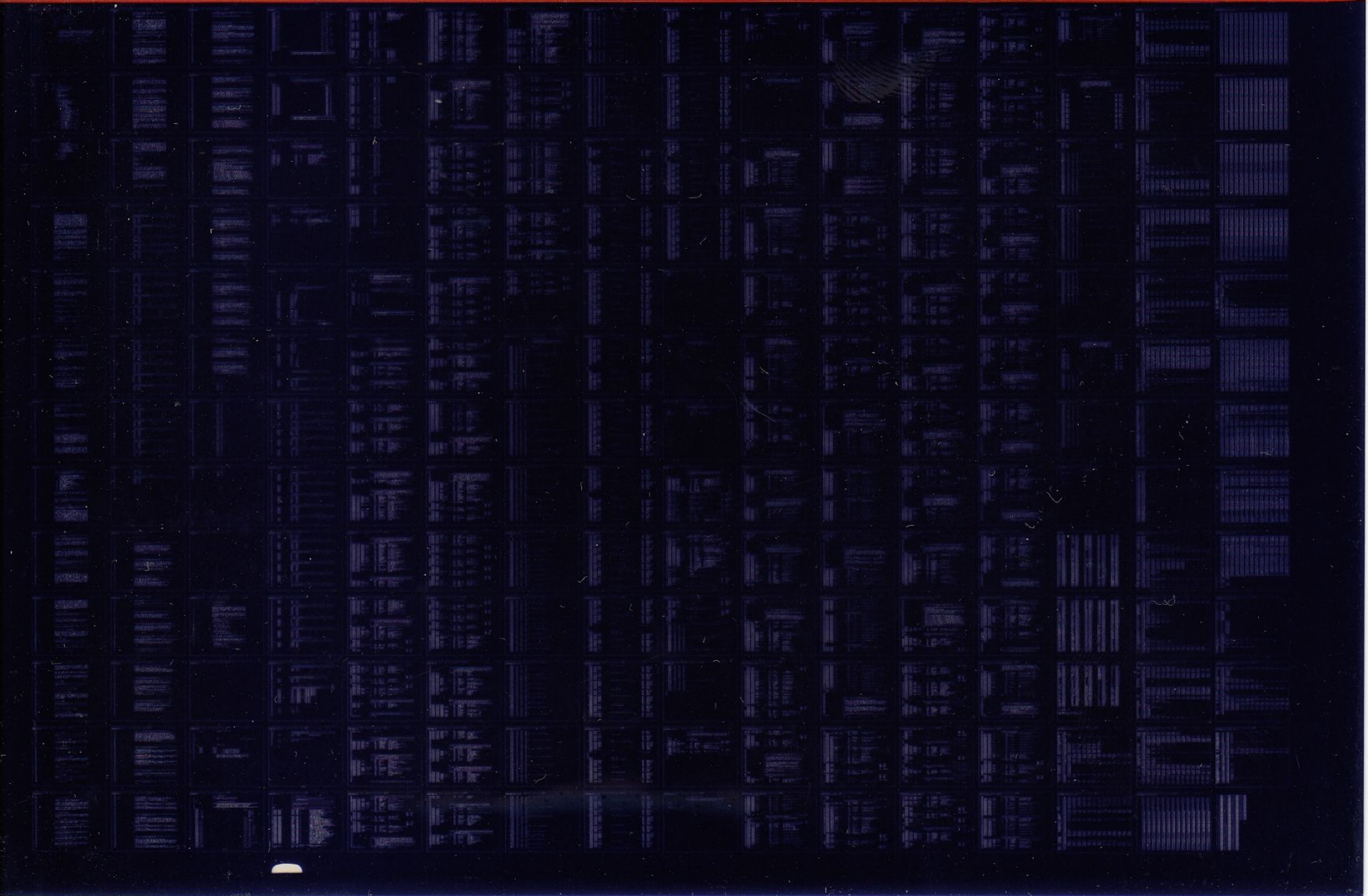
AH-E235B-MC

COPYRIGHT 1980
FICHE 1 OF 2

JAN 1980

digital

MADE IN USA



M8203

M8203 STATIC DIAG. # 2
CZDMSB0

AH-E235B-MC
COPYRIGHT 1980
FICHE 2 OF 2

JAN 1980
digital
MADE IN USA

Vertical strip of faint, illegible text or markings on the left edge of the page.

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

.REM @

IDENTIFICATION

PRODUCT CODE: AC-E234B-MC
PRODUCT NAME: CZDMSB0 M8203 STATIC DIAG #2
PRODUCT DATE: OCTOBER 1979
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: DAVID HOFFMAN

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

CONTENTS

- 1.0 INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
 - 4.1 DIAGNOSTIC SUPERVISOR
 - 4.2 EXECUTION TIME
 - 4.3 XXDP+
 - 4.4 ACT/SLIDE
 - 4.5 APT
 - 4.6 MEMORY MANAGEMENT
 - 4.7 MEMORY PARITY OPTION
 - 4.8 ERROR LOGGING
- 5.0 PROGRAM LOAD MEDIA
- 6.0 OPERATING INSTRUCTIONS
 - 6.1 LOADING AND STARTING PROCEDURES
 - 6.1.1 LOADING PROCEDURES
 - 6.1.2 STARTING PROCEDURES
 - 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
 - 6.2 INITIAL DIALOGUE
 - 6.3 PROGRAM OPTIONS
 - 6.3.1 START COMMAND
 - 6.3.1.1 TESTS SWITCH
 - 6.3.1.2 PASS SWITCH
 - 6.3.1.3 FLAGS SWITCH
 - 6.3.1.4 END OF PASS SWITCH
 - 6.3.1.5 EFFECT OF START COMMAND
 - 6.3.2 RESTART COMMAND
 - 6.3.2.1 TESTS, PASS, AND FLAG SWITCHES
 - 6.3.2.2 UNITS SWITCH
 - 6.3.2.3 EFFECT OF RESTART COMMAND
 - 6.3.3 CONTINUE COMMAND
 - 6.3.3.1 PASS SWITCH
 - 6.3.3.2 FLAGS SWITCH
 - 6.3.3.3 EFFECT OF CONTINUE COMMAND
 - 6.3.4 PROCEED COMMAND
 - 6.3.4.1 FLAGS SWITCH
 - 6.3.4.2 EFFECT OF PROCEED COMMAND
 - 6.3.5 ADD COMMAND
 - 6.3.5.1 UNITS SWITCH
 - 6.3.5.2 EFFECT OF ADD COMMAND
 - 6.3.6 DROP COMMAND
 - 6.3.6.1 UNITS SWITCH
 - 6.3.6.2 EFFECT OF DROP COMMAND
 - 6.3.7 PRINT COMMAND
 - 6.3.7.1 EFFECT OF PRINT COMMAND

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

6.3.8 DISPLAY COMMAND
6.3.8.1 UNITS SWITCH
6.3.8.2 EFFECT OF DISPLAY COMMAND
6.3.9 FLAGS COMMAND
6.3.9.1 EFFECT OF FLAGS COMMAND
6.3.10 ZFLAGS COMMAND
6.3.10.1 EFFECT OF ZFLAGS COMMAND
6.3.11 CONTROL CHARACTERS
6.3.12 HARDWARE PARAMETERS
6.3.13 SOFTWARE PARAMETERS
6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

7.0 DEVICE INFORMATION TABLES

8.0 TEST DESCRIPTIONS
8.1 DATA PATTERNS USED

9.0 ERROR INFORMATION
9.1 ERROR REPORTING

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1.0 INTRODUCTION

THE M8203 IS A SINGLE-LINE SYNCHRONOUS LINE UNIT MODULE WHICH SUPPORTS BOTH CHARACTER -ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS, AND WHICH IS CURRENTLY EMPLOYED IN THE DMP-11 DDCMP MULTIDROP PROJECT. THE PURPOSE OF THIS PROGRAM IS TO PERFORM DIAGNOSTIC TESTING OF ALL M8203 LOGIC IN A RELATIVELY STATIC MANNER. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: LINE UNIT REGISTER ADDRESSING, USYRT ADDRESSING, STATIC BIT INTERACTION AND READ/WRITE LOGIC TESTS, BASIC TRANSMITTER AND RECEIVER SEQUENCING AND DATA BUFFERING AND STATIC OPERATIONS IN CHARACTER AND BIT-STUFFING MODES. IN ADDITION DATA MESSAGES WILL BE SENT AT SPEEDS OF 2400 BAUD TO 1 MEGABAUD, WITH LOOPBACK IN THE USYRT, ON THE LINE UNIT AT TTL LEVEL, OR THROUGH AN EXTERNAL TEST CONNECTOR WITH A SPECIFIC MODEM INTERFACE SELECTED.

THE STATIC LOGIC TESTS WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO 'LOCK' ONTO INTERMITTENT ERRORS. IN ADDITION TESTS WILL BE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

THIS PROGRAM WILL BE IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN WILL CONFORM TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM WILL BE COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8203 STATIC LOGIC TESTS:

- PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70
- 16K MEMORY
- CONSOLE TERMINAL
- DMC-11 OR KMC-11 MICROPROCESSOR

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

M8203 LINE UNIT AND BC08S-1 CABLE AND BERG CONNECTORS
H3254 AND H3255 TEST CONNECTORS (IF NOT PRESENT, SOME TESTS
WILL BE SKIPPED)

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM OPERATES THE MICROPROCESSOR EXTENSIVELY IN
ORDER TO TEST THE LINE INIT. FOR THIS REASON, THE
MICROPROCESSOR DIAGNOSTIC AND SUBSYSTEM FUNCTIONAL TESTS
SHOULD BE RUN FIRST, AND ANY FAULTS FOUND IN THE
MICROPROCESSOR MODULE SHOULD BE REPAIRED, PRIOR TO RUNNING
THE M8203 STATIC LOGIC TESTS.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC
SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE
SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR
AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED
PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE MAXIMUM TIME REQUIRED TO RUN THE M8203 STATIC LOGIC TESTS
IS ABOUT 45 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN
DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN
IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING
APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS
INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE
DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT
THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR
RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE
ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM
ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE
ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST,
FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE
DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY
THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE
ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD
MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR
WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC
PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+, WITHOUT
READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS-C>)
- C) ENTER STA<CR>
- D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
- E) GET END OF PASS MESSAGES OR ERROR MESSAGES
- F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
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

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

DRS LOADED
DIAG. RUN-TIME SERVICES
CZDMS-B-0
M8203 STATIC LOGIC TESTS - PART 2 OF 2
UNIT IS M8203
DR>

THE OPERATOR THEN RESPONDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

```
*****  
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
<FLAG-LIST>/EOP:<INCR>  
*****
```

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
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

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
:SR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT	LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION '# UNITS?' TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM 'UNIT' REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION '# UNITS?' IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE 'TOO MANY UNITS' IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

```
*****  
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
  <FLAG-LIST>/UNITS:<UNIT-LIST>  
*****
```

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

```
*****  
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>  
*****
```

6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

```
*****  
PRO(CEED)/FLAGS:<FLAG-LIST>  
*****
```

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

NOTE THAT IF THE MESSAGE "TOO MANY UNITS" IS ISSUED, TWO OR MORE CORE IMAGES MUST BE CREATED (WITH DIFFERENT NAMES) TO TEST ALL UNITS.

NOTE THAT ALTHOUGH THE CHAINABLE IMAGE CAN BE EXECUTED ON A 16K MACHINE, THE ORIGINAL CCI CREATION MUST BE DONE ON A LARGE MACHINE, THE EXACT SIZE BEING DEPENDENT ON WHICH UPDATE UTILITY IS USED.

6.3.5 ADD COMMAND

ADD/UNITS:<UNIT-LIST>

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

DRO(P)/UNITS:<UNIT-LIST>

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST
START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT
STATISTICAL REPORTING) FLAG IS CLEARED.

6.3.8 DISPLAY COMMAND

DIS(PLAY)/UNITS:<UNIT-LIST>

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED
OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS
THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO
DESIGNATED.

6.3.9 FLAGS COMMAND

FLA(GS)

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

ZFL(AGS)

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

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
568
569
570

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- HARD CORE QUESTIONS (SEE 6.2), HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 8 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. DEVICE CSR ADDRESS : (O) 160170?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT VALUE IS 160170.

2. DEVICE VECTOR ADDRESS : (O) 300 ?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

3. DEVICE PRIORITY LEVEL : (O) 5 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 5.

4. M8203 SWITCH PACK #1 (REG 11) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF SWITCH PACK #1, WHICH RESIDES IN INBUS REG 11. THE ALLOWABLE RANGE IS 000-053, AND THE DEFAULT VALUE IS 000.

5. M8203 SWITCH PACK #2 (REG 15) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF SWITCH PACK #2, WHICH RESIDES IN INBUS REG 15. THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.

6. M8203 SWITCH PACK #3 (REG 16) : (O) 0 ?

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
622
623
624
625
626
627

THIS IS THE EXPECTED CONTENT (OCTAL) OF SWITCH PACK #3, WHICH RESIDES IN INBUS REG 16. THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.

7. TURNAROUND TYPE -
(0=H3254&H3255, 1=CABLE, 2=MOD LOC, 3=MOD REM, 4=NONE)
: (0) 0 ?

THIS INDICATOR TELLS THE PROGRAM WHETHER TEST CONNECTOR(S) WILL BE MOUNTED SO THAT CERTAIN TESTS CAN BE RUN IN EXTERNAL LOOPBACK MODE. THE ALLOWABLE ANSWERS ARE 0-4, AND THE DEFAULT VALUE IS 0. 0 MEANS THAT THE H3254 AND H3255 TEST CONNECTORS WILL BE USED. 1 MEANS THAT EXTERNAL TURNAROUND WILL BE PROVIDED AT THE FAR END OF A CABLE ATTACHED TO THE J1 OR J2 CONNECTOR. 2 MEANS THAT MODEM LOCAL LOOPBACK WILL BE USED, AND 3 MEANS THAT MODEM REMOTE LOOPBACK WILL BE USED. 4 MEANS THAT NO EXTERNAL TURNAROUND WILL BE PROVIDED. WHEN 0 IS SELECTED, ALL TESTS WILL BE RUN, AND IF 1-4 IS SELECTED, CERTAIN TESTS CANNOT BE RUN, AND THE PROGRAM WILL TYPE THE NUMBER(S) OF TEST(S) TO BE SKIPPED.

8. PLEASE SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K; '2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K; '6' FOR 500K; OR '7' FOR 1 MEG BAUD : (0) 4?

THIS IS THE BAUD RATE WHICH IS SELECTED IN THE SWITCH PACK ON THE M8203. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 4 (FOR 56K).

6.3.13 SOFTWARE PARAMETERS

FOUR SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THE M8203 STATIC LOGIC TESTS PROGRAM, PART 2. THESE QUESTIONS ARE THE FOLLOWING:

1. IS MAN. INTERVEN. DESIRED TO MOUNT TEST CONNECTOR(S)
(L) N?

IF THE OPERATOR ANSWERS THE QUESTION WITH Y (YES), THE PROGRAM WILL LATER PAUSE BEFORE TESTING EACH LOGICAL UNIT AND INFORM THE OPERATOR TO INSTALL THE APPROPRIATE TEST CONNECTOR(S) ON THAT UNIT, AND THEN PROCEED TO TEST THAT UNIT. IF THE OPERATOR ANSWERS N (NO) TO THE ABOVE QUESTION, THE PROGRAM WILL PERFORM TESTING ON ALL UNITS WITHOUT ALLOWING MANUAL INTERVENTION BETWEEN UNITS. IN THIS CASE, ALL TEST CONNECTOR(S) ON ALL UNITS SHOULD BE INSTALLED PRIOR TO RUNNING THE PROGRAM.

2. SHOULD SWITCH PACK AND AX3-15 PRINTOUT BE ALLOWED (L) N ?

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE PRINTOUT OF SWITCH PACKS 1-3 AND MODEM INTERFACE REG AX3-15 ON ANY PASS IN WHICH THE CORRESPONDING TESTS ARE RUN. THE DEFAULT IS NO, WHICH ONLY ALLOWS THE PRINTOUT ON THE FIRST PASS AFTER LOADING, IF THE TESTS ARE RUN.

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
673
674
675
676
677
678
679
680
681
682
683
684

3. SHOULD SWITCH PACK TESTS BE ALLOWED (L) N ?

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE READING AND COMPARISON OF SWITCH PACKS 1-3 TO VALUES ENTERED INTO THE HARDWARE P-TABLE FOR THIS UNIT, IF THE CORRESPONDING TEST IS RUN. IF ALLOWED, SWITCH PACK ERRORS WILL BE REPORTED. THE DEFAULT IS NO, AND THE TESTS ARE NOT RUN.

4. MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : (0) 0 ?

THIS VALUE CONTROLS THE DURATION OF THE RECEIVER MESSAGE TIME-OUT IN A NUMBER OF TESTS WHICH SEND AND RECEIVE MESSAGES ON THE VARIOUS MODEM INTERFACES WITH EXTERNAL LOOPBACK. THE SMALLER THE VALUE, THE LONGER THE TIME-OUT (UP TO SEVERAL SECONDS). THE DEFAULT IS 0.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION '# UNITS?' IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR. IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

UNITS (D) ? 16

UNIT 0

<QUESTION 1> ? 75

<QUESTION 2> ? 0-6

<QUESTION 3> ? 76

UNIT 7

<QUESTION 1> ?

<QUESTION 2> ? 7-11,,13-15

<QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

7.0 DEVICE INFORMATION TABLES

* MAINTENANCE REGISTER - BSEL1

RUN = BIT7
MCLR = BIT6
STEPLU = BIT4
LULOOP = BIT3
ROMC = BIT2
ROMI = BIT1
STEPMP = BIT0

* OBUS REG 10 - TRANSMITTER BUFFER

TX7 = BIT7
TX6 = BIT6
TX5 = BIT5
TX4 = BIT4
TX3 = BIT3
TX2 = BIT2
TX1 = BIT1
TX0 = BIT0

* OBUS REG 11

OC = BIT7
GOAH = BIT3
ABORT = BIT2
EOM = BIT1
SOM = BIT0

* OBUS REG 12

IC = BIT7
BPOLL = BIT6
LULP = BIT5

* OBUS REG 13

POLL = BIT7
DTR = BIT6
SELFR = BIT5
HDX = BIT4
MAINT1 = BIT3
MAINT2 = BIT2
SELSBY = BIT1

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

```
*****  
* OBUS REG 14  
*****  
TXEN = BIT6  
DISSI = BIT5  
RDAX = BIT4  
WAX = BIT3  
ENAX = BIT2  
AX2 = BIT1  
AX1 = BIT0  
  
*****  
* OBUS REG 17  
*****  
CRC2 = BIT7  
CRC1 = BIT6  
IDLE = BIT5  
SECA = BIT4  
STRIP = BIT3  
RDALL = BIT2  
IERR = BIT1  
DDCMP = BIT0  
  
*****  
* IBUS REG 10 - RECEIVER BUFFER  
*****  
RX7 = BIT7  
RX6 = BIT6  
RX5 = BIT5  
RX4 = BIT4  
RX3 = BIT3  
RX2 = BIT2  
RX1 = BIT1  
RX0 = BIT0  
  
*****  
* IBUS REG 11  
*****  
OC = BIT7  
OACT = BIT6  
SW3 = BIT5  
ORDY = BIT4  
SW2 = BIT3  
SW1 = BIT2  
SW0 = BIT1  
UNRR = BIT0  
  
*****  
* IBUS REG 12  
*****  
IC = BIT7  
IACT = BIT6  
LULP = BIT5  
IRDY = BIT4  
OVR = BIT3  
RAB = BIT2  
EBLK = BIT1
```

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

BCC = BIT0

* IBUS REG 13

RING = BIT7
DTR = BIT6
RTS = BIT5
HDX = BIT4
MODR = BIT3
CS = BIT2
STBY = BIT1
CARR = BIT0

* IBUS REG 14

READY = BIT7
TXEN = BIT6
DISSI = BIT5
RDAX = BIT4
WAX = BIT3
ENAX = BIT2
AX2 = BIT1
AX1 = BIT0

* IBUS REG 17

SIGR = BIT7
SIGQ = BIT6
TXDATA = BIT5
OCOR = BIT4
ICIR = BIT3
TESTMD = BIT2
MCLK = BIT1
DDCMP = BIT0

* AX0-15 - USYRT REG 0 (READ ONLY)

RX7 = BIT7
RX6 = BIT6
RX5 = BIT5
RX4 = BIT4
RX3 = BIT3
RX2 = BIT2
RX1 = BIT1
RX0 = BIT0

* AX0-16 - USYRT REG 1 (READ ONLY)

RERR = BIT7
ASBC2 = BIT6
ASBC1 = BIT5
ASBC0 = BIT4

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
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

ROR = BIT3
RABT = BIT2
REOM = BIT1
RSOM = BIT0

* AX1-15 - USYRT REG 2

TX7 = BIT7
TX6 = BIT6
TX5 = BIT5
TX4 = BIT4
TX3 = BIT3
TX2 = BIT2
TX1 = BIT1
TX0 = BIT0

* AX1-16 - USYRT REG 3

TERR = BIT7
TXGA = BIT3
TXAB = BIT2
TEOM = BIT1
TSOM = BIT0

* AX2-15 - USYRT REG 4

SYN7 = BIT7
SYN6 = BIT6
SYN5 = BIT5
SYN4 = BIT4
SYN3 = BIT3
SYN2 = BIT2
SYN1 = BIT1
SYN0 = BIT0
SYNCH = 226

* AX2-16 - USYRT REG 5

APA = BIT7
DDC = BIT6
STR = BIT5
SEC = BIT4
IDL = BIT3
CRCTY2 = BIT2
CRCTY1 = BIT1
CRCTY0 = BIT0

* AX3-15 - USYRT REG 6

I422 = BIT7
XYZ = BIT6
C32BCC = BIT5

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250

V35 = BIT4
INTGRL = BIT3
C32ENB = BIT2
OP = BIT1
TEST = BIT0
AX315U = I422!XYZ!C32BCC!V35!INTGRL!OP

* AX3-16 - USYRT REG 7

TXLEN2 = BIT7
TXLEN1 = BIT6
TXLENO = BIT5
RXLEN2 = BIT2
RXLEN1 = BIT1
RXLENO = BIT0

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

8.0 TEST DESCRIPTIONS

```
*****  
TEST 1 - BIT STUFFING TEST  
*  
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS  
* INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,  
* FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN  
* CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS  
* WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH  
* ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED  
* BY THE RECEIVER.  
* PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,  
* 177,376,377  
*****
```

```
*****  
TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST  
*  
* IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.  
* IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE  
* RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV  
* OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT  
* CLEARED BY THE LINE UNIT READING THE USYRT STATUS.  
* THEN, IC IS SET TO CLEAR THE ERROR, AND THIS IS VERIFIED.  
*  
* IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, AND A MASTER CLEAR  
* IS ISSUED TO CLEAR THE ERROR, AND THIS IS VERIFIED.  
*****
```

```
*****  
TEST 3 - ABORT SEQUENCE TEST  
*  
* SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR  
* TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).  
* AS THE FIRST DATA CHAR IS BEING TRANSMITTED,  
* SET THE ABORT BIT (REG 11).  
* ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR  
* AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.  
* ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB  
* IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,  
* RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.  
*
```


58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

* REPEAT THE ABOVE SEQUENCE, SET IC, AND CHECK THAT
* THIS CLEARS RAB.
*
* REPEAT THE ABOVE SEQUENCE, ISSUE MASTER CLEAR, CHECK THAT THIS
* CLEARS RAB.
*
:*****

:*****
TEST 4 - ABORT AND IDLE FLAGS TEST
*
* TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
* WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
* (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
* NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
:*****

:*****
TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
*
* A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
* IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
* UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
* CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
* THE DEVICE IDLES ABORT CHARACTERS.
:*****

:*****
TEST 6 - RECEIVER DISABLE TEST
*
* TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
* ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
* TO BE RECEIVED, IC IS SET.
* THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
* THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
:*****

:*****
TEST 7 - ASSEMBLED BIT COUNT TEST
*
* THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

* DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
* BITS :
* A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
* 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
* TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
* IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
* THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
:*****

:*****

TEST 8 - SECONDARY STATION ADDRESS BIT TEST

*
* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
* BIT MODE, AND THE SECA BIT (REG 17) IS SET.
* 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
* RECEIVED.
*
* NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
* WITH THE FIRST WORD OF DATA PATTERN T. 2 FLAGS ARE SENT,
* FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
* AND A TERMINATING FLAG.
* THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
* ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
*
* THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
* DATA PATTERN T.
* PATTERN T = 000,125,252,176,177
:*****

:*****

TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST

*
* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
* BIT MODE, AND THE SECA BIT IS SET..
* 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
* RECEIVED.
* NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
* ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
* THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
* IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
:*****

:*****

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
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

TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC

*
* THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
* SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
* TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
* CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
* THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
* AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
* 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
* AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
* THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
:*****

:*****
TEST 11 - SWITCH PACK PRINTOUT AND TEST

*
* - READ AND PRINT SWITCH PACK 1 :
* THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
* SWITCHES ARE IN BITS 1,2,3,5.
*
* - READ AND PRINT SWITCH PACK 2 :
* THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
* SWITCHES ARE IN BITS 0-7.
*
* - READ AND PRINT SWITCH PACK 3 :
* THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
* SWITCHES ARE IN BITS 0-7.
:*****

:*****
TEST 12 - REG AX3-15 PRINTOUT

*
* IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
* THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
* PRINT THE REG.
:*****

:*****
TEST 13 - CRC GENERATION TEST

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
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

*
* - CRC-16, CHAR MODE:
* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
* TO CLOCK THE DATA. AT THE END OF THE MESSAGE THE
* PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
*
* - CRC-CCITT - 1'S PRESET:
* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
*
* - CRC-CCITT - 0'S PRESET:
* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
;*****

;*****
TEST 14 - CRC ERROR DETECTION TEST
*
* - CRC-16, CHAR MODE :
* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
* TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
* THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
* ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
* AN ERROR.
*
* - CRC-CCITT - 1'S PRESET :
* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
*
* - CRC-CCITT - 0'S PRESET :
* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
;*****

;*****
TEST 15 - VRC PARITY GENERATION TEST
*
* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
* LAST 4 CHARS IT SHOULD = 0.
*
* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

```
* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.  
* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE  
* LAST 4 CHARS IT SHOULD = 1.  
*  
* DATA PATTERN Q = 000,120,125,137,040,052,057,177  
;*****  
  
;*****  
* TEST 16 - VRC ERROR DETECTION TEST  
*  
* SUBTEST 1 - FORCING OF BCC USING ODD VRC  
* THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.  
* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER  
* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM  
* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING  
* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE  
* RECEIVED (INDICATING AN ERROR).  
*  
* SUBTEST 2 - FORCING OF BCC USING EVEN VRC  
* THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.  
* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER  
* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM  
* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING  
* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE  
* RECEIVED (INDICATING AN ERROR).  
*  
* DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,  
* 172,176.  
;*****  
  
;*****  
* TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC  
*  
* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A  
* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR  
* ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF  
* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT  
* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE  
* SKIPPED FOR THAT UNIT.  
;*****  
  
;*****  
* TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC  
*  
* THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
```

343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
* SKIPPED FOR THAT UNIT.

; TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC

*
* THE RS232C RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
* AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
* AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
* OR A MODEM TEST MODE. THE MESSAGE CONSISTS
* OF 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
* P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
* PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.

; TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC

*
* THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
* SKIPPED FOR THAT UNIT.

; TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST

*
* THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 DOES NOT INHIBIT
* LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
* A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
* BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
* CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
* THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
* BE UNLOADED (ALL CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
* IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

```
*****  
; TEST 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED  
*  
* THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND  
* THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.  
* THE DATA IS SENT WITH LULOOK SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS  
* THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED  
* WHEN THE HDX BIT IS SET.  
*****
```

```
*****  
; TEST 23 - INTERACTION OF MODEM CONTROL BITS  
*  
* THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT  
* THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. OTHERWISE, THE TEST WILL  
* BE SKIPPED FOR THE UNIT.  
* THE FOLLOWING SUBTESTS ARE PERFORMED:  
* - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED  
* STATE, WITH LULOOK SET TO 1. THEN, LULOOK IS CLEARED AND REG 13 IS READ  
* AND CHECKED FOR THE PROPER STATE, WITH LULOOK CLEARED.  
* REG 13 IS THEN LOADED WITH 0'S. AND READ AND CHECKED FOR THE  
* INITIALIZED STATE.  
* REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.  
* - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.  
* - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGQ SET.  
* - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.  
* - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.  
* - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.  
* - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.  
* - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.  
* - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.  
* - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED INTO THE TX SILO, THE LINE  
* UNIT IS CLOCKED UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND  
* CHECKED FOR RTS, CS, CARR SET.  
*****
```

```
*****  
; TEST 24 - DATA TEST - BIT MODE, NO ERR DET  
*  
* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION  
* INHIBITED. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,  
* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,  
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.  
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE  
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
```

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

```
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
*           375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
* INHIBITED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
*           375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
*
* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
*           375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
*
* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
```


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
568
569
570

```
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
*           375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 28 - DATA TEST - CHAR MODE, CRC-16
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
* DETECTION. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
*           375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 29 - DATA TEST - CHAR MODE, ODD VRC
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
*           375,373,367,357,337,277,177
* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
;*****

;*****
TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
```

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
622
623
624
625
626
627

* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
;*****

;*****
* TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
*
* IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
* IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
* FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
* THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
* ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
* ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
* AND TRANSMITTED AND RECEIVED CORRECTLY.
;*****

;*****
* TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
*
* IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
* DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
* FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
* AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
* AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1' IS
* SENT WITH ITS CORRESPONDING CRC.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,SYNCH,SYNCH
;*****

8.1 DATA PATTERNS USED

***** DATA PATTERN A *****

PATA:
.BYTE 125
.BYTE 252
.BYTE 000
.BYTE 377
.BYTE 001
.BYTE 002
.BYTE 004

628	.BYTE	010
629	.BYTE	020
630	.BYTE	040
631	.BYTE	100
632	.BYTE	200
633	.BYTE	376
634	.BYTE	375
635	.BYTE	373
636	.BYTE	367
637	.BYTE	357
638	.BYTE	337
639	.BYTE	277
640	.BYTE	177

***** DATA PATTERN Q *****

PATQ:	.BYTE	000
	.BYTE	120
	.BYTE	125
	.BYTE	137
	.BYTE	040
	.BYTE	052
	.BYTE	057
	.BYTE	177

***** DATA PATTERN R *****

PATR:	.BYTE	000
	.BYTE	100
	.BYTE	120
	.BYTE	124
	.BYTE	164
	.BYTE	172
	.BYTE	176
	.BYTE	177
	.BYTE	000
	.BYTE	100
	.BYTE	120
	.BYTE	124
	.BYTE	164
	.BYTE	172
	.BYTE	176

***** DATA PATTERN S *****

PATS:	.BYTE	000
	.BYTE	017
	.BYTE	036
	.BYTE	074
	.BYTE	170
	.BYTE	360
	.BYTE	037
	.BYTE	076
	.BYTE	174
	.BYTE	370
	.BYTE	077
	.BYTE	176
	.BYTE	374
	.BYTE	177
	.BYTE	376

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
673
674
675
676
677
678
679
680
681
682
683
684

685
686
687
688
689
690
691
692
693
694
695

.BYTE 377

***** DATA PATTERN I *****

PATT: .BYTE 000
.BYTE 125
.BYTE 252
.BYTE 176
.BYTE 177

1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES AN 'IRDY NOT SET' ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE PC OF THE CALL TO THE SUBROUTINE REPORTING IT, THE FAILING REGISTER NAME, AND DEVICE REGISTER CONTENTS :

CZDMS DVC FTL ERR 00017 ON UNIT 00 TST 034 SUB 000 PC: 006210
IRDY NOT SET
PC OF SUBR CALL: 030044
DEVICE CSR ADDRESS : 160170

FAILING REG: INBUS/OUTBUS REG 12

LINE UNIT INBUS REGS:
REG10 REG11 REG12 REG13
000 120 000 257
REG14 REG15 REG16 REG17
024 377 377 035

LINE UNIT EXTENDED REGS:
AX0-15 AX0-16 AX1-15 AX1-16
000 000 000 000
AX2-15 AX2-16 AX3-15 AX3-16
000 000 000 000

FOR OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE, AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

IF EXTENDED ERROR INFORMATION HAD BEEN INHIBITED USING THE IXE FLAG PRIOR TO RUNNING THE TEST, THE ABOVE ERROR WOULD HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :

CZDMS DVC FTL ERR 00017 ON UNIT 00 TST 034 SUB 000 PC:006210
IRDY NOT SET
PC OF SUBR CALL: 030044
DEVICE CSR ADDRESS : 160170

FAILING REG: INBUS/OUTBUS REG 12

1
2
3
4
5
6
7
8
9
10

a

```
1          .TITLE CZDMSB M8203 STATIC TESTS #2
10         002000          .=2000
11
12
13
14
15
16
17         .MCALL  SVC
18 002000  SVC          ; INITIALIZE SUPERVISOR MACROS
19
20
21
22
23
24 002000  BGNMOD  LU2MOD
25
26
27         000001  $LSTIN= 1
28         000001  $LSTTAG= 1
29         000001  SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
30         000001  SVCTST= 1     ; LIST TEST TAGS, SHIFTED RIGHT
31         000001  SVCSUB= 1    ; LIST SUBTEST TAGS, SHIFTED RIGHT
32         000001  SVCGBL= 1    ; LIST GLOBAL TAGS, SHIFTED RIGHT
33         000001  SVCTAG= 1    ; LIST OTHER TAGS, SHIFTED RIGHT
34
35         ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
36         ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
37         ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
38         ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
39
40
```


002046 000000
002050
002050 003
002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003162
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 023136
002072
002072 023054
002074
002074 000000
002076
002076 003170
002100
002100 104035
002102
002102 000000
002104
002104 022020
002106
002106 023052
002110
002110 022772
002112
002112 022012
002114
002114 000000
002116
002116 000000
002120
002120 000000

LSMREV:: .WORD 0
 .BYTE CSREVISION
LSEF:: .BYTE CSREDIT
 .WORD 0
LSSPC:: .WORD 0
L\$DEVP:: .WORD 0
L\$REPP:: .WORD LSDVTYP
L\$EXP4:: .WORD 0
L\$EXP5:: .WORD 0
L\$AUT:: .WORD 0
L\$DUT:: .WORD L\$AU
L\$LUN:: .WORD L\$DU
L\$DESP:: .WORD 0
L\$LOAD:: .WORD L\$DESC
 EMT ESLOAD
L\$ETP:: .WORD 0
L\$ICP:: .WORD L\$INIT
L\$CCP:: .WORD L\$CLEAN
L\$ACP:: .WORD L\$AUTO
L\$PRT:: .WORD L\$PROT
L\$TEST:: .WORD 0
L\$DLY:: .WORD 0
L\$HIME:: .WORD 0

18
20
21
22
24
25
26
27
28
29
30
31

:XX
: CHANGE THE 'HEADER' TO CONTAIN THE PROPER ARGUMENTS.
:XX

.EVEN

.SBTTL DEFAULT HARDWARE P-TABLE

:/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
:/ THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
:/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.

```
1
2
3
4
5
6
7
8
9 002224          BGNHW  DFPTBL
   002224 000013
   002226
   002226
10
11 002226 000007      .WORD      7      :MICROPROCESSOR TYPE = M8207
12 002230 160170      .WORD     160170    :DMC11 OR KMC11 CSR UNIBUS ADDRESS
13 002232 000300      .WORD      300     :DMC11 OR KMC11 INTERRUPT VECTOR
14 002234 005000      .WORD     5000     :DMC11 OR KMC11 INTERRUPT PRIORITY LEVEL = 5
15 002236 000003      .WORD      3       :LINE UNIT = M8203
16 002240 000056      .WORD     056     :SWITCH PACK #1 (REG 11)
17 002242 000000      .WORD      000     :SWITCH PACK #2 (REG 15)
18 002244 000000      .WORD      000     :SWITCH PACK #3 (REG 16)
19 002246 000000      .WORD      0       :H3254&H3255 USED
20 002250 000004      .WORD      4       :BAUD RATE = 56 K
21 002252 000001      .WORD      1       :RUN SWITCH ON MICROPROCESSOR IS ON
22
23 002254          ENDDHW
   002254
24
25
26
27
28
```

.WORD L10000-L\$HW/2
L\$HW::
DFPTBL::

L10000:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

.SBTTL SOFTWARE P-TABLE

:/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
:/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.

BGNSW SFPTBL

.WORD L10001-L\$SW/2

L\$SW::
SFPTBL::

MIFLAG: .WORD 0 ; =1 IF MAN. INTERVENTION DESIRED, =0 IF NOT
PRNFLG: .WORD 0 ; =1 IF SW PACK AND AX3-15 PRINTOUT ALLOWED ALWAYS
SWIFLG: .WORD 0 ; =1 IF SWITCH PACK VERIFICATION TEST SHOULD BE RUN
TCOUNT: .WORD 0 ; INITIAL MSG TIME-OUT VALUE (0=LONGEST TIME-OUT)

ENDSW

L10001:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 002266

.SBTTL GLOBAL EQUATES SECTION

:/
:/ THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
:/ ARE USED IN MORE THAN ONE TEST.
:/

EQUALS

:
: BIT DIFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

:
: EVENT FLAG DEFINITIONS
: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	: START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED

000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED

.....
: PRIORITY LEVEL DEFINITIONS
.....

000340 PRI07== 340
000300 PRI06== 300
000240 PRI05== 240
000200 PRI04== 200
000140 PRI03== 140
000100 PRI02== 100
000040 PRI01== 40
000000 PRI00== 0

.....
: OPERATOR FLAG BITS
.....

000004 EVL== 4
000010 LOT== 10
000020 ADR== 20
000040 IDU== 40
000100 ISR== 100
000200 UAM== 200
000400 BOE== 400
001000 PNT== 1000
002000 PRI== 2000
004000 IXE== 4000
010000 IBE== 10000
020000 IER== 20000
040000 LOE== 40000
100000 HOE== 100000

.....
: * PROGRAM EVENT FLAG DEFINITIONS
: *

.....
: * MAINTENANCE REGISTER - BSEL1
: *

000200 RUN = BIT7
000100 MCLR = BIT6
000020 STEPLU = BIT4
000010 LULOOOP = BIT3
000004 ROMO = BIT2
000002 ROMI = BIT1
000001 STEPMP = BIT0

.....
: * OBUS REG 10 - TRANSMITTER BUFFER
: *

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

47	000200	TX7	=	BIT7
48	000100	TX6	=	BIT6
49	000040	TX5	=	BIT5
50	000020	TX4	=	BIT4
51	000010	TX3	=	BIT3
52	000004	TX2	=	BIT2
53	000002	TX1	=	BIT1
54	000001	TX0	=	BIT0

;* OBUS REG 11

59	000200	OC	=	BIT7
60	000010	GOAH	=	BIT3
61	000004	ABORT	=	BIT2
62	000002	EOM	=	BIT1
63	000001	SOM	=	BIT0

;* OBUS REG 12

68	000200	IC	=	BIT7
69	000100	BPOLL	=	BIT6
70	000040	LULP	=	BIT5

;* OBUS REG 13

75	000200	POLL	=	BIT7
76	000100	DTR	=	BIT6
77	000040	SEFR	=	BIT5
78	000020	HDX	=	BIT4
79	000010	MAINT1	=	BIT3
80	000004	MAINT2	=	BIT2
81	000002	SELSBY	=	BIT1

;* OBUS REG 14

86	000100	TXEN	=	BIT6
87	000040	DISSI	=	BIT5
88	000020	RDAX	=	BIT4
89	000010	WAX	=	BIT3
90	000004	ENAX	=	BIT2
91	000002	AX2	=	BIT1
92	000001	AX1	=	BIT0

;* OBUS REG 17

97	000200	CRC2	=	BIT7
98	000100	CRC1	=	BIT6
99	000040	IDLE	=	BIT5
100	000020	SECA	=	BIT4
101	000010	STRIP	=	BIT3
102	000004	RDALL	=	BIT2
103	000002	IERR	=	BIT1

104 000001 DDCMP = BIT0

105
106 ::*****

107 :* IBUS REG 10 - RECEIVER BUFFER

108 ::*****

109 000200 RX7 = BIT7
110 000100 RX6 = BIT6
111 000040 RX5 = BIT5
112 000020 RX4 = BIT4
113 000010 RX3 = BIT3
114 000004 RX2 = BIT2
115 000002 RX1 = BIT1
116 000001 RX0 = BIT0

117
118 ::*****

119 :* IBUS REG 11

120 ::*****

121 000200 OC = BIT7
122 000100 OACT = BIT6
123 000040 SW3 = BIT5
124 000020 ORDY = BIT4
125 000010 SW2 = BIT3
126 000004 SW1 = BIT2
127 000002 SW0 = BIT1
128 000001 UNRR = BIT0

129
130 ::*****

131 :* IBUS REG 12

132 ::*****

133 000200 IC = BIT7
134 000100 IACT = BIT6
135 000040 LULP = BIT5
136 000020 IRDY = BIT4
137 000010 OVRR = BIT3
138 000004 RAB = BIT2
139 000002 EBLK = BIT1
140 000001 BCC = BIT0

141
142 ::*****

143 :* IBUS REG 13

144 ::*****

145 000200 RING = BIT7
146 000100 DTR = BIT6
147 000040 RTS = BIT5
148 000020 HDX = BIT4
149 000010 MODR = BIT3
150 000004 CS = BIT2
151 000002 STBY = BIT1
152 000001 CARR = BIT0

153
154 ::*****

155 :* IBUS REG 14

156 ::*****

157 000200 READY = BIT7
158 000100 TXEN = BIT6
159 000040 DISSI = BIT5
160 000020 RDAX = BIT4

161 000010 WAX = BIT3
162 000004 ENAX = BIT2
163 000002 AX2 = BIT1
164 000001 AX1 = BIT0

:* IBUS REG 17

168 000200 SIGR = BIT7
170 000100 SIGQ = BIT6
171 000040 TXDATA = BIT5
172 000020 OCOR = BIT4
173 000010 ICIR = BIT3
174 000004 TESTMD = BIT2
175 000002 MCLK = BIT1
176 000001 DDCMP = BIT0

:* AX0-15 - USYRT REG 0 (READ ONLY)

181 000200 RX7 = BIT7
182 000100 RX6 = BIT6
183 000040 RX5 = BIT5
184 000020 RX4 = BIT4
185 000010 RX3 = BIT3
186 000004 RX2 = BIT2
187 000002 RX1 = BIT1
188 000001 RX0 = BIT0

:* AX0-16 - USYRT REG 1 (READ ONLY)

193 000200 RERR = BIT7
194 000100 ASBC2 = BIT6
195 000040 ASBC1 = BIT5
196 000020 ASBC0 = BIT4
197 000010 ROR = BIT3
198 000004 RABT = BIT2
199 000002 REOM = BIT1
200 000001 RSOM = BIT0

:* AX1-15 - USYRT REG 2

205 000200 TX7 = BIT7
206 000100 TX6 = BIT6
207 000040 TX5 = BIT5
208 000020 TX4 = BIT4
209 000010 TX3 = BIT3
210 000004 TX2 = BIT2
211 000002 TX1 = BIT1
212 000001 TX0 = BIT0

:* AX1-16 - USYRT REG 3

216 000200 TERR = BIT7

218 000010 TXGA = BIT3
219 000004 TXAB = BIT2
220 000002 TEOM = BIT1
221 000001 TSOM = BIT0

222
223 ::*****
224 :* AX2-15 - USYRT REG 4
225 ::*****

226 000200 SYN7 = BIT7
227 000100 SYN6 = BIT6
228 000040 SYN5 = BIT5
229 000020 SYN4 = BIT4
230 000010 SYN3 = BIT3
231 000004 SYN2 = BIT2
232 000002 SYN1 = BIT1
233 000001 SYN0 = BIT0
234 000226 SYNCH = 226

235
236 ::*****
237 :* AX2-16 - USYRT REG 5
238 ::*****

239 000200 APA = BIT7
240 000100 DDC = BIT6
241 000040 STR = BIT5
242 000020 SEC = BIT4
243 000010 IDL = BIT3
244 000004 CRCTY2 = BIT2
245 000002 CRCTY1 = BIT1
246 000001 CRCTY0 = BIT0

247
248 ::*****
249 :* AX3-15 - USYRT REG 6
250 ::*****

251 000200 I422 = BIT7
252 000100 XYZ = BIT6
253 000040 C32BCC = BIT5
254 000020 V35 = BIT4
255 000010 INTGRL = BIT3
256 000004 C32ENB = BIT2
257 000002 OP = BIT1
258 000001 TEST = BIT0
259 000372 AX315U = I422!XYZ!C32BCC!V35!INTGRL!OP

260
261 ::*****
262 :* AX3-16 - USYRT REG 7
263 ::*****

264 000200 TXLEN2 = BIT7
265 000100 TXLEN1 = BIT6
266 000040 TXLEN0 = BIT5
267 000004 RXLEN2 = BIT2
268 000002 RXLEN1 = BIT1
269 000001 RXLEN0 = BIT0

270
271
272
273
274

275
276
277
278 004000
279 002000
280 001000
281 000400
282
283
284
285
286
287
288
289
290 004000
291 002000
292 001000
293 000400
294
295
296
297
298
299
300
301 002266
302 002270
303 002272
304 002274
305 002276
306 002300
307 002302
308 002304
309 002306
310 002310
311 002312
312 002314
313 002316
314 002320
315 002322
316 002324
317
318
319
320
321
322 100000
323
324 100000
325 100000
326
327 100000
328
329
330
331

:* TX CONTROL BITS DEFINED ON WORD BASIS

TXGOA = BIT11
TXABT = BIT10
TXEOM = BIT9
TXSOM = BIT8

:* RCV CONTROL BITS DEFINED ON WORD BASIS

RXOVR = BIT11
RXABT = BIT10
RXEBL = BIT9
RXBCC = BIT8

:* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)

LUR10 = LUREG+0 ;LINE UNIT IBUS REG 10
LUR11 = LUREG+2 ;LINE UNIT IBUS REG 11
LUR12 = LUREG+4 ;LINE UNIT IBUS REG 12
LUR13 = LUREG+6 ;LINE UNIT IBUS REG 13
LUR14 = LUREG+10 ;LINE UNIT IBUS REG 14
LUR15 = LUREG+12 ;LINE UNIT IBUS REG 15
LUR16 = LUREG+14 ;LINE UNIT IBUS REG 16
LUR17 = LUREG+16 ;LINE UNIT IBUS REG 17
AX0.15 = LUREG+20 ;USYRT REG 0
AX0.16 = LUREG+22 ;USYRT REG 1
AX1.15 = LUREG+24 ;USYRT REG 2
AX1.16 = LUREG+26 ;USYRT REG 3
AX2.15 = LUREG+30 ;USYRT REG 4
AX2.16 = LUREG+32 ;USYRT REG 5
AX3.15 = LUREG+34 ;USYRT REG 6
AX3.16 = LUREG+36 ;USYRT REG 7

CHPCHK = BIT15
BCCCHK = BIT15
CRCCHK = BIT15
TCCHEK = BIT15

332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348

021000
122000

```
*****  
;* MICROINSTRUCTION DEFINITIONS  
*****  
MVI0X = 021000 ;MOVE IBUS TO OBUS*  
MVI0X = 122000 ;MOVE IBUS* TO OBUS
```

```
***** ERROR1 BIT FLAG DEFINITIONS *****  
RRDYTO = BIT0  
WRDYTO = BIT1
```

```
1      .SBTTL GLOBAL DATA SECTION
2
3      :////////////////////////////////////////////////////////////////////
4      :/ THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5      :/ IN MORE THAN ONE TEST.
6      :////////////////////////////////////////////////////////////////////
7
8      :*****
9      :* STORAGE FOR DEVICE REGISTERS
10     :*****
11     002266 LUREG: .BLKW 16.
12
13     :*****
14     :* MISCELLANEOUS STORAGE
15     :*****
16     002326 000000 SCRACH: .WORD 0 ;GEN'L PURPOSE SCRATCH WORD
17     002330 000000 LOGDEV: .WORD 0 ;LOGICAL DEVICE NUMBER
18     002332 000000 PSTACK: .WORD 0 ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
19     002334 000000 PRIOR: .WORD 0 ;CPU PRIORITY FOR PRINTOUT
20     002336 000000 SUBRPC: .WORD 0 ;PC OF SUBR CALL FOR ERROR REPORTS
21     002340 000000 INTFLG: .WORD 0 ;INTERRUPT RECEIVED FLAGS
22     ; BIT 0 FOR TX, BIT 1 FOR RCV
23     002342 000000 ERRFLG: .WORD 0 ;SUBROUTINE ERROR FLAG
24     002344 000000 TIMFLG: .WORD 0 ;EVENT TIME-OUT FLAG
25     002346 000000 RETADR: .WORD 0 ;SUBR ERROR RETURN ADDRESS
26     002350 000000 REDBYT: .WORD 0 ;LO BYTE CONTAINS BYTE READ FROM LU REG
27     002352 000000 WRIBYT: .WORD 0 ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
28     002354 000000 RAX15: .WORD 0 ;LO BYTE CONTAINS BYTE READ FROM REG 15
29     002356 000000 RAX16: .WORD 0 ;LO BYTE CONTAINS BYTE READ FROM REG 16
30     002360 000000 WAX15: .WORD 0 ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
31     002362 000000 WAX16: .WORD 0 ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
32     002364 000000 REGNUM: .WORD 0 ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
33     002366 000000 AXNUM: .WORD 0 ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
34     002370 000000 GOODAT: .WORD 0 ;STORAGE FOR EXPECTED DATA
35     002372 000000 BADDAT: .WORD 0 ;STORAGE FOR ACTUAL DATA
36     002374 000000 LOADAT: .WORD 0 ;CONTAINS TEST DATA LOADED INTO REG
37     002376 000000 FRSTIM: .WORD 0 ;FLAG=0 IF PROGRAM JUST LOADED
38     002400 000000 FRSPAS: .WORD 0 ;FLAG=0 IF FIRST PASS AFTER LOAD
39     002402 000000 STARES: .WORD 0 ;FLAG=0 IF FIRST TIME THRU AFTER STA OR RES
40     002404 000000 SAVE4: .WORD 0 ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
41     002406 000000 SAVE6: .WORD 0 ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
42     002410 000000 ERROR1: .WORD 0 ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
43     002412 000000 TXWORD: .WORD 0 ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
44     002414 000000 RXWORD: .WORD 0 ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
45     002416 000000 DISILO: .WORD 0 ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
46     002420 000000 CHPTYP: .WORD 0 ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
47     002422 000000 MODINT: .WORD 0 ;MODEM INTERFACE SELECTION
48     002424 000000 SAVLEN: .WORD 0 ;SAVED TX AND RCV CHAR LENGTHS
49     002426 000000 DEVMAP: .WORD 0 ;BIT MAP OF ACTIVE DEVICES
50     002430 000000 DEVPTR: .WORD 0 ;DEVICE MAP BIT POINTER
51     002432 000000 UNIT: .WORD 0 ;CONTAINS UNIT NUMBER (1 TO N)
52     002434 000000 TSTNUM: .WORD 0 ;CONTAINS TEST NUMBER FOR SOME TESTS
53
54     :***** CURRENT DEVICE PARAMETERS *****
55     002436 160170 MPCSR: .WORD 160170 ;POINTER TO MICROPROCESSOR CSR'S
56     002440 160171 BSEL1: .WORD 160171 ;POINTER TO BSEL1
57     002442
```

```

58 002442 160174 SEL4: .WORD 160174 :POINTER TO SEL4
59 002444 160176 SEL6: .WORD 160176 :POINTER TO SEL6
60 002446 000300 MPIVEC: .WORD 300 :MICROPROCESSOR INPUT INTERRUPT VECTOR
61 002450 000304 MPOVEC: .WORD 304 :MICROPROCESSOR OUTPUT INTERRUPT VECTOR
62 002452 000240 MPRIOR: .WORD 240 :MICROPROCESSOR DEVICE PRIORITY
63 002454 000000 LUSWI1: .WORD 0 :LINE UNIT SWITCH PACK #1
64 002456 000000 LUSWI2: .WORD 0 :LINE UNIT SWITCH PACK #2
65 002460 000000 LUSWI3: .WORD 0 :LINE UNIT SWITCH PACK #3
66 002462 000000 TSTCON: .WORD 0 :TEST CONNECTOR INDICATOR
67 002464 000004 BDRATE: .WORD 4 :BAUD RATE
68
69 ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
70 002466 000 REDDAT: .BYTE 0
71 002467 000 .BYTE 0
72 002470 000 .BYTE 0
73 002471 000 .BYTE 0
74 002472 000 .BYTE 0
75 002473 000 .BYTE 0
76 002474 000 .BYTE 0
77 002475 000 .BYTE 0
78
79 ;***** GEN'L PURPOSE SCRATCH STORAGE *****
80 002476 000000 REG0: .WORD 0
81 002500 000000 REG1: .WORD 0
82 002502 000000 REG2: .WORD 0
83 002504 000000 REG3: .WORD 0
84 002506 000000 REG4: .WORD 0
85 002510 000000 REG5: .WORD 0
86 002512 000000 REG6: .WORD 0
87 002514 000000 REG7: .WORD 0
88
89 ;***** SCRATCH STORAGE FOR MESSAGE REPORTING *****
90 002516 000000 TMP0: .WORD 0
91 002520 000000 TMP1: .WORD 0
92 002522 000000 TMP2: .WORD 0
93 002524 000000 TMP3: .WORD 0
94 002526 000000 TMP4: .WORD 0
95 002530 000000 TMP5: .WORD 0
96 002532 000000 TMP6: .WORD 0
97 002534 000000 TMP7: .WORD 0
98
99 ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
100 002536 000 UPBITS: .BYTE 000 :MASK FOR REG 10
101 002536 056 .BYTE 056 :MASK FOR REG 11
102 002537 000 .BYTE 000 :MASK FOR REG 12
103 002540 257 .BYTE 257 :MASK FOR REG 13
104 002541 100 .BYTE 100 :MASK FOR REG 14
105 002542 377 .BYTE 377 :MASK FOR REG 15
106 002543 377 .BYTE 377 :MASK FOR REG 16
107 002544 377 .BYTE 377 :MASK FOR REG 16
108 002545 306 .BYTE 306 :MASK FOR REG 17
109
110 002546 200 R14NRW: .BYTE 200 :REG 14 NON-R/W BITS
111
112 ;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****
113 002547 ANBITS:
114 002547 377 .BYTE 377 :MASK FOR AX0-15

```

115	002550	377	.BYTE	377	:MASK FOR AX0-16
116	002551	000	.BYTE	000	:MASK FOR AX1-15
117	002552	360	.BYTE	360	:MASK FOR AX1-16
118	002553	000	.BYTE	000	:MASK FOR AX2-15
119	002554	000	.BYTE	000	:MASK FOR AX2-16
120	002555	004	.BYTE	004	:MASK FOR AX3-15
121	002556	030	.BYTE	030	:MASK FOR AX3-16

***** DATA PATTERN A *****

122					
123					
124	002557				
125	002557	125	.BYTE	125	
126	002560	252	.BYTE	252	
127	002561	000	.BYTE	000	
128	002562	377	.BYTE	377	
129	002563	001	.BYTE	001	
130	002564	002	.BYTE	002	
131	002565	004	.BYTE	004	
132	002566	010	.BYTE	010	
133	002567	020	.BYTE	020	
134	002570	040	.BYTE	040	
135	002571	100	.BYTE	100	
136	002572	200	.BYTE	200	
137	002573	376	.BYTE	376	
138	002574	375	.BYTE	375	
139	002575	373	.BYTE	373	
140	002576	367	.BYTE	367	
141	002577	357	.BYTE	357	
142	002600	337	.BYTE	337	
143	002601	277	.BYTE	277	
144	002602	177	.BYTE	177	

***** DATA PATTERN B *****

145					
146					
147	002603				
148	002603	000	.BYTE	000	
149	002604	000	.BYTE	000	
150	002605	040	.BYTE	040	
151	002606	100	.BYTE	100	
152	002607	220	.BYTE	220	
153	002610	000	.BYTE	000	
154	002611	000	.BYTE	000	
155	002612	051	.BYTE	051	

***** DATA PATTERN Q *****

156					
157					
158					
159	002613	000	.BYTE	000	
160	002614	120	.BYTE	120	
161	002615	125	.BYTE	125	
162	002616	137	.BYTE	137	
163	002617	040	.BYTE	040	
164	002620	052	.BYTE	052	
165	002621	057	.BYTE	057	
166	002622	177	.BYTE	177	

***** DATA PATTERN R *****

167					
168					
169	002623	000	.BYTE	000	
170	002624	100	.BYTE	100	
171	002625	120	.BYTE	120	

172	002626	124	.BYTE	124
173	002627	164	.BYTE	164
174	002630	172	.BYTE	172
175	002631	176	.BYTE	176
176	002632	177	.BYTE	177
177	002633	000	.BYTE	000
178	002634	100	.BYTE	100
179	002635	120	.BYTE	120
180	002636	124	.BYTE	124
181	002637	164	.BYTE	164
182	002640	172	.BYTE	172
183	002641	176	.BYTE	176

184				
185				
186	002642	000	:***** DATA PATTERN S *****	
187	002643	017	PATS:	.BYTE 000
188	002644	036		.BYTE 017
189	002645	074		.BYTE 036
190	002646	170		.BYTE 074
191	002647	360		.BYTE 170
192	002650	037		.BYTE 360
193	002651	076		.BYTE 037
194	002652	174		.BYTE 076
195	002653	370		.BYTE 174
196	002654	077		.BYTE 370
197	002655	176		.BYTE 077
198	002656	374		.BYTE 176
199	002657	177		.BYTE 374
200	002660	376		.BYTE 177
201	002661	377		.BYTE 376

202				
203				
204	002662	000	:***** DATA PATTERN T *****	
205	002663	125	PATT:	.BYTE 000
206	002664	252		.BYTE 125
207	002665	176		.BYTE 252
208	002666	177		.BYTE 176

209				
210	002667		ENDPAT:	
211				.EVEN

212				
213				
214				
215				
216				
217			:*** TEST MESSAGES TO BE TRANSMITTED ***	

218				
219	002670	000400	MSG1:	TXSOM
220	002672	000400		TXSOM
221	002674	000000		000
222	002676	000125		125
223	002700	000252		252
224	002702	000377		377
225	002704	000000		000
226	002706	001000		TXEOM
227	002710	001000		TXEOM
228	002712	001000		TXEOM

229	002714	001000		TXEOM
230				
231	002716	000400	MSG2:	TXSOM
232	002720	000400		TXSOM
233	002722	000000		000
234	002724	000377		377
235	002726	001000		TXEOM
236	002730	001000		TXEOM
237				
238	002732	000001	MSG3:	001
239	002734	000001		001
240	002736	000001		001
241	002740	000001		001
242	002742	002000		TXABT
243	002744	000400		TXSOM
244	002746	000400		TXSOM
245	002750	000003		003
246	002752	000003		003
247	002754	000003		003
248	002756	000003		003
249	002760	000003		003
250				
251				
252				
253				
254				
255				
256	002762			
257				
258				
259				
260				
261				
262				
263				
264				
265				
266				

*** RECEIVED DATA BUFFER (64. WORDS) ***
RCVBUF: .BLKW 64.

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

.SBTTL GLOBAL TEXT SECTION

;%
;% THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
;% MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
;% MORE THAN ONE TEST.
;%

;* NAMES OF DEVICES SUPPORTED BY PROGRAM

DEV TYP <M8203>

003162			
003162	115	070	062
003162	060	063	000
003165			

L\$DVTYP::
.ASCIZ /M8203/
.EVEN

;* TITLE OF PROGRAM

DESCRIPT <M8203 STATIC LOGIC TESTS - PART 2 OF 2>

003170			
003170	115	070	062
003173	060	063	040
003176	123	124	101
003201	124	111	103
003204	040	114	117
003207	107	111	103
003212	040	124	105
003215	123	124	123
003220	040	055	040
003223	120	101	122
003226	124	040	062
003231	040	117	106
003234	040	062	000

L\$DESC::
.ASCIZ /M8203 STATIC LOGIC

.EVEN

;
; FORMAT STATEMENTS USED IN PRINT CALLS
;

;%
; INSERT THE FORMAT STATEMENTS USED IN THE VARIOUS PRINT CALLS.
; USE THE .ASCIZ STATEMENT.
;%

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```
.SBTTL GLOBAL SUBROUTINES
:
://////
:/ THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
://////
```

```
*****
:* STPCLK - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
:* EXECUTE AN INSTRUCTION WHICH IS PASSED IN THE WORD FOLLOWING THE CALL.
*****
```

15 003240
16 003240 152777 000006 177172
17 003246 017677 000000 177170
18 003254 152777 000007 177156
19 003262 142777 000007 177150
20 003270 062716 000002
21 003274 000207

```
STPCLK:
BISB #ROMO!ROMI,@BSEL1 ;SET ROMO, ROMI BITS IN BSEL1
MOV @ (SP),@SEL6 ;PUT INSTRUCTION INTO SEL6
BISB #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
BICB #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
ADD #2,(SP) ;FIX UP RETURN PC
RTS PC ;RETURN
```

```
*****
:* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOK
*****
```

30 003276
31 003276 010146
32 003300 112777 000100 177132
33 003306 142777 000300 177124
34 003314 012701 000024
35 003320 000240
36 003322 005301
37 003324 001375
38 003326 152777 000010 177104
39 003334 012601
40 003336 005037 002424
41 003342 000207

```
MSTCLR:
MOV R1,-(SP) ;SAVE R1
MOVB #MCLR,@BSEL1 ;SET MASTER CLEAR BIT
BICB #RUN!MCLR,@BSEL1 ;CLEAR RUN AND MCLR BITS
MOV #20.,R1 ;INITIALIZE STALL COUNTER
2$: NOP ;STALL IN LOOP FOR SEVERAL MICRO-SEC
DEC R1
BNE 2$
BISB #LULOOK,@BSEL1 ;SET LU LOOP
MOV (SP)+,R1 ;RESTORE R1
CLR SAVLEN ;CLEAR SAVED CHAR LENGTH FROM SETUP
RTS PC ;RETURN
```

```
*****
:* READLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR
:* TO EXECUTE AN INSTRUCTION WHICH READS THE LINE UNIT REG WHOSE
:* NUMBER IS PASSED IN REGNUM, INTO REDBYT.
*****
```

52 003344
53 003344 010146
54 003346 013701 002364
55 003352 006301
56 003354 006301
57 003356 006301

```
READLU:
MOV R1,-(SP) ;SAVE R1
MOV REGNUM,R1 ;GET LINE UNIT REG NUMBER
ASL R1 ;SHIFT INTO SOURCE BITS 4-7
ASL R1
ASL R1
```

```
58 003360 006301 ASL R1
59 003362 052701 000004 BIS #4,R1 ;SET DESTINATION = BSEL4
60 003366 052701 021000 BIS #MVIOX,R1 ;SET REST OF MOVE INSTRUCTION
61 003372 010137 003402 MOV R1,2$ ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
62 003376 004737 003240 JSR PC,STPCLK ;EXECUTE MOVE INSTRUCTION
63 003402 000000 2$: .WORD 0 ;INSTRUCTION GOES HERE
64 003404 117737 177032 002350 MOVB @BSEL4,REDBYT ;GET LU REG CONTENTS INTO REDBYT
65 003412 105037 002351 CLRB REDBYT+1 ;CLR HI BYTE OF STORAGE
66 003416 012601 MOV (SP)+,R1 ;RESTORE R1
67 003420 000207 RTS PC ;RETURN
```

```
*****
;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
;* EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
;* INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM,
*****
```

```
WRITLU:
MOV R1,-(SP) ;SAVE R1
MOV REGNUM,R1 ;GET LINE UNIT REG NUMBER
BIS #100,R1 ;SET SOURCE = BSEL4
BIS #MVIXO,R1 ;SET REST OF MOVE INSTRUCTION
MOV R1,2$ ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
CLRB WRIBYT+1 ;CLR HI BYTE OF STORAGE
MOVB WRIBYT,@BSEL4 ;LOAD BYTE INTO BSEL4
JSR PC,STPCLK ;EXECUTE MOVE INSTRUCTION
2$: .WORD 0
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN
```

```
*****
;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
;* REGISTER STORAGE TABLE (LUREG:).
*****
```

```
GETREG: MOV R1,-(SP) ;SAVE R1
MOV REGNUM,-(SP) ;SAVE CURRENT REG NO.
MOV #LUR10,R1 ;INIT POINTER TO REG STORAGE TABLE
MOV #10,REGNUM ;INIT LU REG NO. TO 10
3$: JSR PC,READLU ;READ A LINE UNIT REG
MOVB REDBYT,(R1)+ ;PUT BYTE READ INTO TABLE
CLRB (R1)+ ;CLEAR UPPER BYTE OF TABLE ENTRY
INC REGNUM ;INCREMENT REG NO.
CMP REGNUM,#20 ;SEE IF ALL REGS READ YET
BLT 3$ ;BR IF NOT
MOV (SP)+,REGNUM ;RESTORE CURRENT REG NO.
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN
```

112
113
114

```

115
116
117
118
119
120
121
122
123
124
125 003546
126 003546 152777 000006 176664
127 003554 017677 000000 176662
128 003562 152777 000206 176650
129 003570 062716 000002
130 003574 000207
131
132
133
134
135
136
137
138
139
140
141
142
143 003576 010146
144 003600 013746 002364
145 003604 042737 000001 002410
146 003612 012737 000014 002364
147 003620 113737 002366 002352
148 003626 006237 002352
149 003632 152737 000024 002352
150 003640 053737 002416 002352
151 003646 004737 003422
152 003652 005001
153 003654 004737 003344 6$:
154 003660 132737 000200 002350
155 003666 001006
156 003670 005201
157 003672 001370
158 003674 052737 000001 002410
159 003702 000424
160 003704 012737 000015 002364 9$:
161 003712 004737 003344
162 003716 113737 002350 002354
163 003724 105037 002355
164 003730 012737 000016 002364
165 003736 004737 003344
166 003742 113737 002350 002356
167 003750 105037 002357
168 003754 012637 002364 12$:
169 003760 012601
170 003762 000207
171

```

```

:*****
:* LOOPIN - THIS SUBROUTINE PLACES THE MICROPROCESSOR IN A LOOP ON AN
:* INSTRUCTION, BY MOVING THE INSTRUCTION FROM THE WORD FOLLOWING THE CALL
:* INTO SEL6, AND SETTING RUN AND ROMI IN BSEL1. THE SUBROUTINE RETURNS
:* WITH THE MICROPROCESSOR STUCK IN THE LOOP, AND IF IT IS DESIRED TO
:* TERMINATE THE LOOP, THE PDP-11 PROGRAM MUST CLEAR THE RUN BIT IN
:* BSEL1, OR CALL SUBROUTINE MSTCLR TO DO THIS.
:*****

```

```

LOOPIN:
BISB #ROMO!ROMI,@BSEL1 ;SET ROMO, ROMI BITS IN BSEL1
MOV @ (SP),@SEL6 ;PUT MICROINSTRUCTION INTO SEL6
BISB #RUN!ROMO!ROMI,@BSEL1 ;SET RUN, ROMO, ROMI IN BSEL1
ADD #2,(SP) ;FIX UP RETURN PC
RTS PC ;RETURN WITH MICROPROCESSOR STUCK IN SINGLE
; INSTRUCTION LOOP

```

```

:*****
:* READAX - THIS SUBROUTINE READS THE USYRT REG PAIR WHOSE NUMBER (0-3)
:* IS PASSED IN BITS 1,2 OF AXNUM ON ENTRY, AND RETURNS THE BYTES READ IN
:* RAX15 AND RAX16. IF THE LINE UNIT DOES NOT RESPOND WITH READY IN REG 14,
:* RRDYTO BIT IS SET IN ERROR1 ON RETURN.
:*****

```

```

READAX: MOV R1,-(SP) ;SAVE R1
MOV REGNUM,-(SP) ;STORE CURRENT REG NO.
BIC #RRDYTO,ERROR1 ;CLEAR ERROR BIT
MOV #14,REGNUM ;SET LU REG NO. = 14
MOVB AXNUM,WRIBYT ;SET UP AX REG NO. BITS
ASR WRIBYT
BISB #RDAX!ENAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
BIS DISILO,WRIBYT ;SET PROPER STATE OF DISSI BIT
JSR PC,WRITLU ;SET RDAX AND ENAX IN REG 14
CLR R1 ;INIT TIMER
6$: JSR PC,READLU ;READ REG 14
BITB #READY,REDBYT ;SEE IF READY BIT SET IN REG 14 YET
BNE 9$ ;BR IF READY SET
INC R1 ;INCR TIMER
BNE 6$ ;BR IF TIMER DIDN'T TIME OUT YET
BIS #RRDYTO,ERROR1 ;SET ERROR FLAG FOR TIME OUT ON READ RDY
BR 12$ ;BR TO RETURN
9$: MOV #15,REGNUM ;SET REG NO. = 15
JSR PC,READLU ;READ REG 15
MOVB REDBYT,RAX15 ;STORE REG AX-15
CLRB RAX15+1 ;CLR HI BYTE OF STORAGE
MOV #16,REGNUM ;SET REG NO. = 16
JSR PC,READLU ;READ REG 16
MOVB REDBYT,RAX16 ;STORE REG AX-16
CLRB RAX16+1 ;CLR HI BYTE OF STORAGE
12$: MOV (SP)+,REGNUM ;RESTORE CURRENT REG NO.
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

```

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
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

003764 010146
003766 013746 002364
003772 042737 000002 002410
004000 012737 000014 002364
004006 113737 002366 002352
004014 006237 002352
004020 053737 002416 002352
004026 004737 003422
004032 012737 000015 002364
004040 105037 002361
004044 113737 002360 002352
004052 004737 003422
004056 005237 002364
004062 105037 002363
004066 113737 002362 002352
004074 004737 003422
004100 012737 000014 002364
004106 113737 002366 002352
004114 006237 002352
004120 152737 000014 002352
004126 053737 002416 002352
004134 004737 003422
004140 005001
004142 004737 003344 6\$:
004146 132737 000200 002350
004154 001005
004156 005201
004160 001370
004162 052737 000002 002410
004170 012637 002364 9\$:
004174 012601
004176 000207

```

:*****
:* WRITAX - THIS SUBROUTINE WRITES THE USYRT REG PAIR WHOSE NUMBER (0-3) IS
:* PASSED IN BITS 1,2 OF AXNUM ON ENTRY, WITH THE DATA FROM WAX15 AND
:* WAX16. IF LINE UNIT DOES NOT RESPOND WITH READY IN REG 14, WRDYTO BIT
:* IS SET IN ERROR1 ON RETURN.
:*****
WRITAX: MOV     R1,-(SP)           ;SAVE R1
        MOV     REGNUM,-(SP)     ;SAVE CURRENT REG NO.
        BIC     #WRDYTO,ERROR1   ;CLEAR ERROR BIT
        MOV     #14,REGNUM       ;SET LU REG NO. = 14
        MOV     AXNUM,WRIBYT     ;SET AX REG NO. BITS
        ASR     WRIBYT
        BIS     DISILO,WRIBYT    ;SET PROPER STATE OF DISSI BIT
        JSR     PC,WRITLU        ;SET AX NO. BITS IN REG 14
        MOV     #15,REGNUM       ;SET REG NO. = 15
        CLRB    WAX15+1         ;CLR HI BYTE OF STORAGE
        MOV     WAX15,WRIBYT     ;SET UP BYTE TO WRITE INTO REG 15
        JSR     PC,WRITLU        ;WRITE BYTE INTO REG 15
        INC     REGNUM           ;SET REG NO. = 16
        CLRB    WAX16+1         ;CLR HI BYTE OF STORAGE
        MOV     WAX16,WRIBYT     ;SET UP BYTE TO WRITE INTO REG 16
        JSR     PC,WRITLU        ;WRITE BYTE INTO REG 16
        MOV     #14,REGNUM       ;SET REG NO. = 14
        MOV     AXNUM,WRIBYT     ;SET AX REG NO. BITS
        ASR     WRIBYT
        BISB    #ENAX!WAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
        BIS     DISILO,WRIBYT    ;SET PROPER STATE OF DISSI BIT
        JSR     PC,WRITLU        ;SET ENAX AND WAX IN REG 14
        CLR     R1               ;INIT PROGRAM TIMER
6$:     JSR     PC,READLU        ;READ REG 14
        BITB    #READY,REDBYT    ;SEE IF READY BIT SET IN REG 14 YET
        BNE     9$               ;BR IF READY SET
        INC     R1               ;INCR TIMER
        BNE     6$               ;BR IF TIMER DIDN'T TIME OUT YET
9$:     BIS     #WRDYTO,ERROR1   ;SET ERROR FLAG BIT FOR TIME OUT ON WRITE RDY
        MOV     (SP)+,REGNUM     ;RESTORE CURRENT REG NO.
        MOV     (SP)+,R1         ;RESTORE R1
        RTS     PC               ;RETURN
    
```

```

:*****
:* GETALL - THIS SUBROUTINE READS THE LINE UNIT REGS 10-17 AND THE EXTENDED
:* REGISTERS AX0-AX3 INTO REGISTER STORAGE TABLE (LUREG:).
:*****
GETALL: MOV     R1,-(SP)           ;SAVE R1
        MOV     AXNUM,-(SP)     ;SAVE CURRENT AX REG BYTE NO.
        MOV     #DH5,TMP0       ;SET AX LO BYTE NO.
        BIT     #BIT0,AXNUM     ;SEE IF LO OR HI BYTE
        BEQ     1$               ;BR IF LO BYTE
        MOV     #DH6,TMP0       ;SET AX HI BYTE NO.
1$:     JSR     PC,GETREG        ;READ AND STORE REGS 10-17
    
```

```
229 004236 142777 000010 176174      BICB    #LULOOP,@BSEL1  :CLEAR LULOOP
230 004244 012701 002306                MOV     #AX0.15,R1      :INIT POINTER TO REG STORAGE TABLE
231 004250 005037 002366                CLR     AXNUM           :INIT AX REG BYTE NO. TO 0
232 004254 004737 003576      3$:    JSR     PC,READAX      :READ 2 AX REG BYTES
233 004260 113721 002354                MOV     RAX15,(R1)+     :PUT LO BYTE READ INTO TABLE
234 004264 105021                CLR     (R1)+           :CLEAR UPPER BYTE OF TABLE ENTRY
235 004266 113721 002356                MOV     RAX16,(R1)+     :PUT HI BYTE READ INTO TABLE
236 004272 105021                CLR     (R1)+           :CLEAR UPPER BYTE OF TABLE ENTRY
237 004274 062737 000002 002366      ADD     #2,AXNUM        :INCR AX REG BYTE NO.
238 004302 023727 002366 000010      CMP     AXNUM,#10      :SEE IF ALL REGS READ YET
239 004310 002761                BLT     3$              :BR IF NOT
240 004312 012637 002366                MOV     (SP)+,AXNUM     :RESTORE CURRENT AX REG BYTE NO.
241 004316 012601                MOV     (SP)+,R1       :RESTORE R1
242 004320 013737 002366 002520      MOV     AXNUM,TMP1
243 004326 006237 002520                ASR     TMP1            :GET EXTENDED REG NO. FOR PRINTOUT
244 004332 000207                RTS     PC              :RETURN
```

245
246
247
248
249
250
251
252
253
254
255
256
257

```
:*****  
:* OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)  
:* AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET  
:* AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE  
:* CALL.  
:* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN  
:* RETADR.  
:*****
```

```
258 004334 013746 002364      OSIRDY: MOV     REGNUM,-(SP)  :SAVE LU REG NO.
259 004340 013746 002336      MOV     SUBRPC,-(SP)
260 004344 005737 002336      TST     SUBRPC         :SEE IF THIS IS A NESTED CALL
261 004350 001006                BNE     1$             :BR IF YES
262 004352 016637 000004 002336      MOV     4(SP),SUBRPC
263 004360 162737 000004 002336      SUB     #4,SUBRPC      :GET PC OF SUBROUTINE CALL
264 004366 012737 000011 002364      1$:    MOV     #11,REGNUM   :SET REG NO. TO 11
265 004374 004737 003344      JSR     PC,READLU     :READ REG 11
266 004400 032776 000001 000004      BIT     #BIT0,@4(SP)  :GET EXPECTED STATE OF ORDY
267 004406 001413                BEQ     3$             :BR IF EXPECTED ORDY = 0
268 004410 132737 000020 002350      BIT     #ORDY,REDBYT  :SEE IF ORDY = 1
269 004416 001022                BNE     9$             :BR IF ORDY = 1
270 004420 004737 004200      JSR     PC,GETALL     :GET REGS FOR PRINTOUT
271                :REPORT ORDY NOT SET
272                ERRDF 7,EM7,ERR4
```

```
004424 104455                TRAP    C$ERDF
004426 000007                .WORD  7
004430 013276                .WORD  EM7
004432 016350                .WORD  ERR4
```

```
273 004434 000451                BR      16$           :TAKE ERROR RETURN
274 004436 132737 000020 002350      3$:    BIT     #ORDY,REDBYT  :SEE IF ORDY = 0
275 004444 001407                BEQ     9$             :BR IF ORDY = 0
276 004446 004737 004200      JSR     PC,GETALL     :GET REGS FOR PRINTOUT
277                :REPORT ORDY NOT CLEARED
278                ERRDF 8,EM8,ERR4
```

```
004452 104455                TRAP    C$ERDF
004454 000010                .WORD  8
004456 013313                .WORD  EM8
```



```

004460 016350
279 004462 000436
280 004464 012737 000017 002364 9$: BR 16$ ;TAKE ERROR RETURN
;MOV #17,REGNUM ;SET REG NO. = 17
281 004472 004737 003344 JSR PC,READLU ;READ LU REG 17
282 004476 132776 000002 000004 BITB #BIT1,@4(SP) ;GET EXPECTED STATE OF OCOR
283 004504 001413 BEQ 12$ ;BR IF EXPECTED OCOR = 0
284 004506 132737 000020 002350 BITB #OCOR,REDBYT ;SEE IF OCOR = 1
285 004514 001031 BNE 20$ ;BR IF OCOR = 1
286 004516 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
287 ;REPORT OCOR NOT SET
288 004522 ERRDF 9,EM9,ERR4
004522 104455 TRAP C$ERDF
004524 000011 .WORD 9
004526 013334 .WORD EM9
004530 016350 .WORD ERR4
289 004532 000412 BR 16$ ;TAKE ERROR RETURN
290 004534 132737 000020 002350 12$: BITB #OCOR,REDBYT ;SEE IF OCOR = 0
291 004542 001416 BEQ 20$ ;BR IF OCOR = 0
292 004544 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
293 ;REPORT OCOR NOT CLEARED
294 004550 ERRDF 10,EM10,ERR4
004550 104455 TRAP C$ERDF
004552 000012 .WORD 10
004554 013351 .WORD EM10
004556 016350 .WORD ERR4
295 004560 016637 000002 002364 16$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.
296 004566 013706 002332 MOV PSTACK,SP ;RESTORE STACK POINTER TO BASE LEVEL
297 004572 013746 002346 MOV RETADR,-(SP) ;FIX ERROR RETURN PC
298 004576 000407 BR 23$
299 004600 062766 000002 000004 20$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
300 004606 012637 002336 MOV (SP)+,SUBRPC
301 004612 012637 002364 MOV (SP)+,REGNUM ;RESTORE LU REG NO.
302 004616 000207 23$: RTS PC ;RETURN
303
304
305
306
307
308
309 ;*****
;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
;*****
310
311 004620 010146 WAIT50: MOV R1,-(SP) ;SAVE R1
312 004622 012701 000310 MOV #200.,R1 ;INIT COUNTER
313 004626 005301 3$: DEC R1 ;DECREMENT COUNTER
314 004630 001376 BNE 3$ ;BR IF NOT DONE YET
315 004632 012601 MOV (SP)+,R1 ;RESTORE R1
316 004634 000207 RTS PC ;RETURN
317
318
319
320
321
322
323 ;*****
;* STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
;*****
324
325 004636 000240 STALL: NOP
326 004640 000240 NOP

```

327 004642 000240
328 004644 000207
329
330
331
332
333
334
335
336
337
338 004646 013746 002364
339 004652 042737 170000 002412
340 004660 012737 000011 002364
341 004666 113737 002413 002352
342 004674 004737 003422
343 004700 012737 000010 002364
344 004706 113737 002412 002352
345 004714 004737 003422
346 004720 012637 002364
347 004724 000207

NOP
RTS PC

;* LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
;* IN BITS 0-11 OF TXWORD.

LDTXSI: MOV REGNUM, -(SP) ;SAVE LU REG NO.
BIC #170000, TXWORD ;CLEAR UNUSED BITS
MOV #11, REGNUM ;SET REG NO. = 11
MOVB TXWORD+1, WRIBYT ;SET DATA TO BE WRITTEN INTO REG 11
JSR PC, WRITLU ;LOAD DATA INTO REG 11
MOV #10, REGNUM ;SET REG NO. = 10
MOVB TXWORD, WRIBYT ;SET DATA TO BE WRITTEN INTO REG 10
JSR PC, WRITLU ;LOAD DATA INTO REG 10
MOV (SP)+, REGNUM ;RESTORE LU REG NO.
RTS PC ;RETURN

348
349
350
351
352
353
354
355
356
357
358

;* STPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES PASSED
;* IN BITS 0-14 OF THE WORD FOLLOWING THE CALL.
;* IF BIT 15 = 1, A CHECK IS MADE TO DETERMINE IF THE USYRT CHIP TYPE
;* REQUIRES DECREMENTING THE NO. OF CYCLES BY 1.

359 004726 010146
360 004730 017601 000002
361 004734 001426
362 004736 100006
363 004740 042701 100000
364 004744 005737 002420
365 004750 001401
366 004752 005301
367 004754 152777 000010 175456 2\$:
368 004762 152777 000020 175450 3\$:
369 004770 004737 004636
370 004774 142777 000020 175436
371 005002 004737 004636
372 005006 005301
373 005010 001364
374 005012 062766 000002 000002 6\$:
375 005020 012601
376 005022 000207

STPLU: MOV R1, -(SP) ;SAVE R1
MOV @2(SP), R1 ;GET DESIRED NO. OF CYCLES
BEQ 6\$;IF DESIRED CYCLES = 0, RETURN
BPL 2\$;BR IF CHIP TYPE CHECK NOT NECESSARY
BIC #BIT15, R1 ;CLEAR FLAG BIT
TST CHPTYP ;SEE IF SIG USYRT
BEQ 2\$;BR IF YES
DEC R1 ;DECREMENT CYCLE COUNT
BISB #LULOOP, @BSEL1 ;SET LU LOOP BIT
BISB #STEPLU, @BSEL1 ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
JSR PC, STALL ;STALL
BICB #STEPLU, @BSEL1 ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
JSR PC, STALL ;STALL
DEC R1 ;DECREMENT CYCLE COUNTER
BNE 3\$;BR IF NOT DONE YET
ADD #2, 2(SP) ;FIX UP RETURN PC
MOV (SP)+, R1 ;RESTORE R1
RTS PC ;RETURN

377
378
379
380
381
382
383

;* OACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF OACT (REG 11) AND

```

384          ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
385          ;*      WORD FOLLOWING THE CALL.
386          ;*****
387 005024 013746 002364 OACTIV: MOV    REGNUM,-(SP) ;SAVE LU REG NO.
388 005030 013746 002336      MOV    SUBRPC,-(SP)
389 005034 005737 002336      TST    SUBRPC ;SEE IF THIS IS A NESTED CALL
390 005040 001006      BNE    1$ ;BR IF YES
391 005042 016637 000004 002336      MOV    4(SP),SUBRPC
392 005050 162737 000004 002336      SUB    #4,SUBRPC ;GET PC OF SUBROUTINE CALL
393 005056 012737 000011 002364 1$: MOV    #11,REGNUM ;SET REG NO. = 11
394 005064 004737 003344      JSR    PC,READLU ;READ REG 11
395 005070 032776 000001 000004      BIT    #BIT0,@4(SP) ;GET EXPECTED STATE OF OACT
396 005076 001413      BEQ    3$ ;BR IF EXPECTED OACT = 0
397 005100 132737 000100 002350      BITB   #OACT,REDBYT ;SEE IF OACT = 1
398 005106 001031      BNE    9$ ;BR IF OACT = 1
399 005110 004737 004200      JSR    PC,GETALL ;GET REGS FOR PRINTOUT
400          ;REPORT OACT NOT SET
401 005114      ERRDF  11,EM11,ERR4
402          TRAP    C$ERDF
403          .WORD  11
404          .WORD  EM11
405          .WORD  ERR4
406          BR     6$ ;TAKE ERROR RETURN
407 005114 104455      BR     6$ ;SEE IF OACT = 0
408 005116 000013      BEQ    9$ ;BR IF OACT = 0
409 005120 013372      JSR    PC,GETALL ;GET REGS FOR PRINTOUT
410 005122 016350      ;REPORT OACT NOT CLEARED
411 005124 000412      ERRDF  12,EM12,ERR4
412 005126 132737 000100 002350 3$: TRAP    C$ERDF
413 005134 001416      .WORD  12
414 005136 004737 004200      .WORD  EM12
415          .WORD  ERR4
416 005142      MOV    2(SP),REGNUM ;RESTORE LU REG NO.
417 005142 104455      MOV    PSTACK,SP ;RESTORE PROGRAM STACK TO BASE LEVEL
418 005144 000014      MOV    RETADR,-(SP) ;FIX UP ERROR RETURN PC
419 005146 013407      BR     12$
420 005150 016350      ADD    #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
421 005152 016637 000002 002364 6$: MOV    (SP)+,SUBRPC
422 005160 013706 002332      MOV    (SP)+,REGNUM ;RESTORE LU REG NO.
423 005164 013746 002346      RTS    PC ;RETURN
424 005170 000407      ;*****
425 005172 062766 000002 000004 9$: ;* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY DOING A
426 005200 012637 002336      ;* MASTER CLEAR, LOADING AX2-15 AND REG 17 WITH THE DATA PASSED IN THE 2
427 005204 012637 002364      ;* WORDS FOLLOWING THE CALL, LOADING 2 SOM CHARS INTO THE TX SILO, AND
428 005210 000207      ;* CLOCKING THE LINE UNIT UNTIL THE FIRST SYNCH OR FLAG HAS BEEN SERIALIZED
429          ;* IN THE USYRT. THE PROGRAM MONITORS ORDY,OCOR, AND OACT FOR VALID STATES,
430          ;* THROUGHOUT THE PROCESS.
431          ;* IF THE SUBROUTINE DETECTS AN ERROR, A RETURN IS MADE TO THE TEST, AT THE
432          ;* ADDRESS CONTAINED IN RETADR.
433          ;*****
434 005212 010146      INITRN: MOV    R1,-(SP) ;SAVE R1
435 005214 013746 002364      MOV    REGNUM,-(SP) ;SAVE LU REG NO.

```

```

433 005220 013746 002366      MOV      AXNUM,-(SP)      ;SAVE AX BYTE NO.
434 005224 016637 000006 002336  MOV      6(SP),SUBRPC    ;
435 005232 162737 000004 002336  SUB      #4,SUBRPC      ;GET PC OF SUBR CALL
436 005240 004737 003276      JSR      PC,MSTCLR      ;ISSUE A MASTER CLEAR
437 005244 004737 004334      JSR      PC,OSIRDY      ;CHECK ORDY=1, OCOR=0
438 005250 000001      1
439 005252 004737 005024      JSR      PC,OACTIV      ;CHK OACT=0
440 005256 000000      0
441 005260 012737 000004 002366  MOV      #4,AXNUM        ;SET AX BYTE NO. = 4 FOR AX2
442 005266 117637 000006 002360  MOVVB   @6(SP),WAX15     ;SET DATA BYTE TO LOAD INTO AX2-15
443 005274 012737 000400 002412  MOV      #TXSOM,TXWORD   ;SET TSOM BIT
444 005302 113737 002360 002412  MOVVB   WAX15,TXWORD     ;SET SYNCH CHAR
445 005310 005037 002362      CLR      WAX16
446 005314 004737 003764      JSR      PC,WRITAX      ;LOAD AX2
447 005320 012737 000017 002364  MOV      #17,REGNUM      ;SET REG NO. = 17
448 005326 062766 000002 000006  ADD      #2,6(SP)        ;INCR POINTER TO NEXT DATA BYTE
449 005334 117637 000006 002352  MOVVB   @6(SP),WRIBYT   ;SET DATA BYTE TO LOAD INTO REG 17
450 005342 004737 003422      JSR      PC,WRITLU      ;LOAD REG 17
451 005346 004737 004646      JSR      PC,LDTXSI      ;LOAD THE SILO WITH SOM CHAR
452 005352 004737 004646      JSR      PC,LDTXSI      ;LOAD ANOTHER SOM INTO SILO
453 005356 004737 004620      JSR      PC,WAIT50      ;WAIT FOR DATA TO RIPPLE
454 005362 004737 004334      JSR      PC,OSIRDY      ;CHK ORDY=1, OCOR=1
455 005366 000003      3
456 005370 004737 005024      JSR      PC,OACTIV      ;CHK FOR OACT = 0
457 005374 000000      0
458 005376 005001      CLR      R1              ;INIT CYCLE COUNTER
459 005400 012737 000011 002364  MOV      #11,REGNUM      ;SET LU REG NO. = 11
460 005406 152777 000010 175024 6$:  BISB   #LULOOP,@BSEL1   ;SET LINE UNIT LOOP BIT
461 005414 152777 000020 175016  BISB   #STEPLU,@BSEL1   ;SET CLOCK BIT
462 005422 004737 004636      JSR      PC,STALL      ;STALL FOR MICRO-SEC
463 005426 004737 003344      JSR      PC,READLU      ;READ REG 11
464 005432 132737 000100 002350  BITB   #OACT,REDBYT     ;SEE IF OACT = 1 YET
465 005440 001014      BNE     9$              ;BR IF OACT = 1
466 005442 142777 000020 174770  BICB   #STEPLU,@BSEL1   ;CLEAR CLOCK BIT
467 005450 004737 004636      JSR      PC,STALL      ;STALL FOR A MICRO-SEC
468 005454 005201      INC     R1              ;INCR CYCLE COUNT
469 005456 020127 000003      CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
470 005462 002751      BLT     6$              ;BR IF NOT
471 005464 004737 005024      JSR      PC,OACTIV      ;CHK FOR OACT = 1
472 005470 000001      1
473 005472 012737 000017 002364 9$:  MOV      #17,REGNUM      ;SET REG NO. = 17
474 005500 005037 002420      CLR     CHPTYP          ;CLEAR USYRT CHIP INDICATOR
475 005504 004737 003344      JSR      PC,READLU      ;READ REG 17
476 005510 132737 000020 002350  BITB   #OCOR,REDBYT     ;CHK FOR OCOR CLEARED YET
477 005516 001403      BEQ     12$            ;BR IF YES - IT IS SIG CHIP
478 005520 012737 000001 002420  MOV      #1,CHPTYP      ;SET INDICATOR FOR OTHER CHIP TYPE
479 005526 142777 000020 174704 12$: BICB   #STEPLU,@BSEL1   ;CLEAR CLOCK BIT
480 005534 004737 004636      JSR      PC,STALL      ;STALL FOR MICRO-SEC
481 005540 004737 004334      JSR      PC,OSIRDY      ;CHK FOR ORDY = 1, OCOR = 0
482 005544 000001      1
483 005546 062766 000002 000006  ADD      #2,6(SP)        ;FIX UP RETURN PC
484 005554 012637 002366      MOV      (SP)+,AXNUM     ;RESTORE AX BYTE NO.
485 005560 012637 002364      MOV      (SP)+,REGNUM    ;RESTORE LU REG NO.
486 005564 012601      MOV      (SP)+,R1       ;RESTORE R1
487 005566 005037 002336      CLR     SUBRPC          ;CLEAR SUBR CALL PC
488 005572 000207      RTS     PC              ;RETURN
489

```

490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546

005574	010146		
005576	010246		
005600	016637	000004	002336
005606	162737	000004	002336
005614	017637	000004	002412
005622	004737	004646	
005626	004737	004620	
005632	062766	000002	000004
005640	005001		
005642	017602	000004	
005646	005702		
005650	100006		
005652	042702	100000	
005656	005737	002420	
005662	001401		
005664	005302		
005666	004737	005024	
005672	000001		
005674	020102		
005676	001410		
005700	004737	004334	
005704	000003		
005706	004737	004726	
005712	000001		
005714	005201		
005716	000763		
005720	004737	004334	
005724	000001		
005726	062766	000002	000004
005734	005037	002336	
005740	012602		
005742	012601		
005744	000207		

```

*****
* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHARACTER, BY LOADING
* THE TX SILO WITH DATA PASSED IN BITS 0-11 OF THE WORD FOLLOWING THE CALL
* AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES PASSED IN BITS 0-14
* OF THE SECOND WORD FOLLOWING THE CALL. IF BIT 15 = 1, A CHK IS MADE TO
* DETERMINE IF THE USYRT CHIP TYPE REQUIRES DECREMENTING THE NO. OF CYCLES
* BY 1. THE PROGRAM CHECKS FOR VALID STATES OF ORDY,
* OCOR, AND OACT THROUGHOUT THE PROCESS.
* IF AN ERROR IS DETECTED, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
* CONTAINED IN RETADR.
*****

```

```

TXCHAR: MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
MOV 4(SP),SUBRPC
SUB #4,SUBRPC ;GET PC OF SUBR CALL
MOV @4(SP),TXWORD ;GET DATA TO BE TRANSMITTED
JSR PC,LDTXSI ;LOAD THE TX SILO WITH THE DATA
JSR PC,WAIT50 ;WAIT FOR DATA TO RIPPLE DOWN SILO
ADD #2,4(SP) ;INCR POINTER
CLR R1 ;INIT CYCLE COUNT
MOV @4(SP),R2 ;GET DESIRED NO. OF CYCLES
TST R2 ;SEE IF CHIP TYPE CHK SHOULD BE MADE
BPL 9$ ;BR IF NOT
BIC #BIT15,R2 ;CLEAR FLAG BIT
TST CHPTYP ;SEE IF SIG USYRT
BEQ 9$ ;BR IF YES
DEC R2 ;DECREMENT NO. OF CYCLES
9$: JSR PC,OACTIV ;CHK OACT = 1
1 ;
CMP R1,R2 ;SEE IF REQUIRED CYCLES DONE YET
BEQ 12$ ;BR IF YES
JSR PC,OSIRDY ;CHK ORDY=1, OCOR=1
3 ;
JSR PC,STPLU ;STEP LU ONE CYCLE
1 ;
INC R1 ;INCR CYCLE COUNT
BR 9$ ;
12$: JSR PC,OSIRDY ;CHK ORDY=1, OCOR=0
1 ;
ADD #2,4(SP) ;FIX UP RETURN PC
CLR SUBRPC ;CLEAR SUBR CALL PC
MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

```

```

*****
* ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
* AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
* AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE

```

```

547
548
549
550
551 005746 013746 002364
552 005752 013746 002336
553 005756 005737 002336
554 005762 001006
555 005764 016637 000004 002336
556 005772 162737 000004 002336
557 006000 012737 000012 002364 1$:
558 006006 004737 003344
559 006012 032776 000002 000004
560 006020 001413
561 006022 132737 000020 002350
562 006030 001022
563 006032 004737 004200
564
565 006036
   006036 104455
   006040 000021
   006042 013430
   006044 016350
566 006046 000451
567 006050 132737 000020 002350 3$:
568 006056 001407
569 006060 004737 004200
570
571 006064
   006064 104455
   006066 000022
   006070 013445
   006072 016350
572 006074 000436
573 006076 012737 000017 002364 9$:
574 006104 004737 003344
575 006110 132776 000001 000004
576 006116 001413
577 006120 132737 000010 002350
578 006126 001031
579 006130 004737 004200
580
581 006134
   006134 104455
   006136 000023
   006140 013466
   006142 016350
582 006144 000412
583 006146 132737 000010 002350 12$:
584 006154 001416
585 006156 004737 004200
586
587 006162
   006162 104455
   006164 000024
   006166 013503
   006170 016350

: * CALL.
: * IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
: * IN RETADR.
: *****
1SIRDY: MOV REGNUM,-(SP) ;SAVE LU REG NO.
        MOV SUBRPC,-(SP)
        TST SUBRPC ;SEE IF THIS IS A NESTED CALL
        BNE 1$ ;BR IF YES
        MOV 4(SP),SUBRPC
        SUB #4,SUBRPC ;GET PC OF SUBR CALL
1$: MOV #12,REGNUM ;SET REG NO. TO 12
     JSR PC,READLU ;READ REG 12
     BIT #BIT1,@4(SP) ;GET EXPECTED STATE OF IRDY
     BEQ 3$ ;BR IF EXPECTED IRDY = 0
     BITB #IRDY,REDBYT ;SEE IF IRDY = 1
     BNE 9$ ;BR IF IRDY = 1
     JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT IRDY NOT SET
ERRDF 17,EM17,ERR4
TRAP C$ERDF
.WORD 17
.WORD EM17
.WORD ERR4

BR 16$ ;TAKE ERROR EXIT
3$: BITB #IRDY,REDBYT ;SEE IF IRDY = 0
    BEQ 9$ ;BR IF IRDY = 0
    JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT IRDY NOT CLEARED
ERRDF 18,EM18,ERR4
TRAP C$ERDF
.WORD 18
.WORD EM18
.WORD ERR4

BR 16$ ;TAKE ERROR RETURN
9$: MOV #17,REGNUM ;SET REG NO. = 17
    JSR PC,READLU ;READ REG 17
    BITB #BIT0,@4(SP) ;GET EXPECTED STATE OF ICIR
    BEQ 12$ ;BR IF EXPECTED ICIR = 0
    BITB #ICIR,REDBYT ;SEE IF ICIR = 1
    BNE 20$ ;BR IF ICIR = 1
    JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT ICIR NOT SET
ERRDF 19,EM19,ERR4
TRAP C$ERDF
.WORD 19
.WORD EM19
.WORD ERR4

BR 16$ ;TAKE ERROR RETURN
12$: BITB #ICIR,REDBYT ;SEE IF ICIR = 0
     BEQ 20$ ;BR IF ICIR = 0
     JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT ICIR NOT CLEARED
ERRDF 20,EM20,ERR4
TRAF C$ERDF
.WORD 20
.WORD EM20
.WORD ERR4

```

```
588 006172 016637 000002 002364 16$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.  
589 006200 013706 002332 MOV PSTACK,SP ;RESTORE STACK POINTER TO BASE LEVEL  
590 006204 013746 002346 MOV RETADR,-(SP) ;FIX ERROR RETURN PC  
591 006210 000407 BR 23$  
592 006212 062766 000002 000004 20$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC  
593 006220 012637 002336 MOV (SP)+,SUBRPC  
594 006224 012637 002364 MOV (SP)+,REGNUM ;RESTORE LU REG NO.  
595 006230 000207 23$: RTS PC ;RETURN
```

596
597
598
599
600

601
602
603
604
605
606
607

```
*****  
* IACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF IACT (REG 12) AND  
* REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE  
* WORD FOLLOWING THE CALL.  
* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN  
* RETADR.  
*****
```

```
608 006232 013746 002364 IACTIV: MOV REGNUM,-(SP) ;SAVE LU REG NO.  
609 006236 013746 002336 MOV SUBRPC,-(SP)  
610 006242 005737 002336 TST SUBRPC ;SEE IF THIS IS A NESTED CALL  
611 006246 001006 BNE 1$ ;BR IF YES  
612 006250 016637 000004 002336 MOV 4(SP),SUBRPC  
613 006256 162737 000004 002336 SUB #4,SUBRPC ;GET PC OF SUBR CALL  
614 006264 012737 000012 002364 1$: MOV #12,REGNUM ;SET REG NO. = 12  
615 006272 004737 003344 JSR PC,READLU ;READ REG 12  
616 006276 032776 000001 000004 BIT #BIT0,@4(SP) ;GET EXPECTED STATE OF IACT  
617 006304 001413 BEQ 3$ ;BR IF EXPECTED IACT = 0  
618 006306 132737 000100 002350 BITB #IACT,REDBYT ;SEE IF IACT = 1  
619 006314 001031 BNE 9$ ;BR IF IACT = 1  
620 006316 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT  
621 ;REPORT IACT NOT SET  
622 006322 ERRDF 21,EM21,ERR4
```

```
TRAP C$ERDF  
.WORD 21  
.WORD EM21  
.WORD ERR4
```

```
623 006332 000412 BR 6$ ;TAKE ERROR EXIT  
624 006334 132737 000100 002350 3$: BITB #IACT,REDBYT ;SEE IF IACT = 0  
625 006342 001416 BEQ 9$ ;BR IF IACT = 0  
626 006344 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT  
627 ;REPORT IACT NOT CLEARED  
628 006350 ERRDF 22,EM22,ERR4
```

```
TRAP C$ERDF  
.WORD 22  
.WORD EM22  
.WORD ERR4
```

```
629 006360 016637 000002 002364 6$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.  
630 006366 013706 002332 MOV PSTACK,SP ;RESTORE PROGRAM STACK TO BASE LEVEL  
631 006372 013746 002346 MOV RETADR,-(SP) ;FIX UP ERROR RETURN PC  
632 006376 000407 BR 12$  
633 006400 062766 000002 000004 9$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC  
634 006406 012637 002336 MOV (SP)+,SUBRPC  
635 006412 012637 002364 MOV (SP)+,REGNUM ;RESTORE LU REG NO.  
636 006416 000207 12$: RTS PC ;RETURN
```

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
673
674
675
676
677
678
679
680
681

006420 013746 002366
006424 013746 002336
006430 005737 002336
006434 001006
006436 016637 000004 002336
006444 162737 000004 002336
006452 012737 000001 002366
006460 004737 003576
006464 032776 000001 000004
006472 001413
006474 132737 000001 002356
006502 001022
006504 004737 004200

006510 104455
006512 000035
006514 013603
006516 017540
006520 000444
006522 132737 000001 002356
006530 001407
006532 004737 004200

006536 104455
006540 000034
006542 013562
006544 017540
006546 000431
006550 132776 000002 000004
006556 001413
006560 132737 000002 002356
006566 001031
006570 004737 004200

006574 104455
006576 000037
006600 013641
006602 017540
006604 000412
006606 132737 000002 002356
006614 001416
006616 004737 004200

```

:*****
:* RSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM AND REOM IN
:* AX0-16, AND REPORTS AN ERROR IF EITHER IS NOT SET TO THE STATE PASSED IN BITS
:* 0,1, RESPECTIVELY, OF THE WORD FOLLOWING THE CALL.
:* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN RETADR.
:*****
RSEOM:  MOV    AXNUM,-(SP)      ;SAVE AX BYTE NO.
        MOV    SUBRPC,-(SP)
        TST    SUBRPC        ;SEE IF THIS IS A NESTED CALL
        BNE    1$           ;BR IF YES
        MOV    4(SP),SUBRPC
        SUB    #4,SUBRPC     ;GET PC OF SUBR CALL
1$:     MOV    #1,AXNUM      ;SET AX BYTE NO. FOR AX0-16
        JSR    PC,READAX    ;READ AX0
        BIT    #BIT0,@4(SP) ;GET EXPECTED STATE OF RSOM
        BEQ    3$           ;BR IF EXPECTED RSOM = 0
        BITB  #RSOM,RAX16   ;SEE IF RSOM = 1
        BNE    9$           ;BR IF RSOM = 1
        JSR    PC,GETALL    ;GET REGS FOR PRINTOUT
;REPORT RSOM NOT SET
ERRDF  29,EM29,ERR6

        BR    16$          ;TAKE ERROR EXIT
3$:     BITB  #RSOM,RAX16   ;SEE IF RSOM = 0
        BEQ    9$           ;BR IF RSOM = 0
        JSR    PC,GETALL    ;GET REGS FOR PRINTOUT
;REPORT RSOM NOT CLEARED
ERRDF  28,EM28,ERR6

        TRAP  C$ERDF
        .WORD 29
        .WORD EM29
        .WORD ERR6

        BR    16$          ;TAKE ERROR RETURN
9$:     BITB  #BIT1,@4(SP) ;GET EXPECTED STATE OF REOM
        BEQ    12$          ;BR IF EXPECTED REOM = 0
        BITB  #REOM,RAX16  ;SEE IF REOM = 1
        BNE    20$          ;BR IF REOM = 1
        JSR    PC,GETALL    ;GET REGS FOR PRINTOUT
;REPORT REOM NOT SET
ERRDF  31,EM31,ERR6

        TRAP  C$ERDF
        .WORD 31
        .WORD EM31
        .WORD ERR6

        BR    16$          ;TAKE ERROR RETURN
12$:    BITB  #REOM,RAX16  ;SEE IF REOM = 0
        BEQ    20$          ;BR IF REOM = 0
        JSR    PC,GETALL    ;GET REGS FOR PRINTOUT
;REPORT REOM NOT CLEARED

```



```
682 006622          ERRDF  30,EM30,ERR6
    006622 104455
    006624 000036
    006626 013620
    006630 017540
683 006632 016637 000002 002366 16$:  MOV  2(SP),AXNUM  ;RESTORE AX BYTE NO.
684 006640 013706 002332          MOV  PSTACK,SP   ;RESTORE STACK POINTER TO BASE LEVEL
685 006644 013746 002346          MOV  RETADR,-(SP) ;FIX ERROR RETURN PC
686 006650 000407
687 006652 062766 000002 000004 20$:  ADD  #2,4(SP)     ;FIX UP ERROR-FREE RETURN PC
688 006660 012637 002336          MOV  (SP)+,SUBRPC
689 006664 012637 002366          MOV  (SP)+,AXNUM  ;RESTORE AX BYTE NO.
690 006670 000207          RTS   PC         ;RETURN
```

```
TRAP  C$ERDF
.WORD 30
.WORD EM30
.WORD ERR6
```

691
692
693
694
695
696
697
698
699

```
*****
;* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
;* SILO ENTRY IN BITS 0-11 OF RXWORD.
*****
```

```
700 006672 013746 002364
701 006676 012737 000012 002364
702 006704 004737 003344
703 006710 113737 002350 002415
704 006716 042737 170000 002414
705 006724 012737 000010 002364
706 006732 004737 003344
707 006736 113737 002350 002414
708 006744 012637 002364
709 006750 000207
```

```
RDRXSI: MOV  REGNUM,-(SP) ;SAVE LU REG NO.
        MOV  #12,REGNUM ;SET REG NO. = 12
        JSR  PC,READLU  ;READ LU REG 12
        MOV  REDBYT,RXWORD+1 ;GET HI BITS OF SILO ENTRY
        BIC  #170000,RXWORD ;CLEAR UNUSED BITS
        MOV  #10,REGNUM  ;SET REG NO. = 10
        JSR  PC,READLU  ;READ REG 10
        MOV  REDBYT,RXWORD ;GET LOW BITS OF SILO ENTRY
        MOV  (SP)+,REGNUM ;RESTORE LU REG NO.
        RTS   PC         ;RETURN
```

710
711
712
713
714
715
716
717
718
719
720
721
722
723

```
*****
;* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE, AND MONITORS
;* STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR IACT = 0, IRDY = 0,
;* ICIR = 1, AND RSOM = 0. THEN, THE LINE UNIT IS CLOCKED USING
;* STEPLU UNTIL IRDY = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3
;* CYCLES AFTER THE NO. OF CYCLES PASSED IN THE WORD FOLLOWING THE CALL.
;* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
;* CONTAINED IN RETADR.
*****
```

```
724 006752 010146
725 006754 010346
726 006756 013746 002364
727 006762 016637 000006 002336
728 006770 162737 000004 002336
729 006776 012737 000012 002364
730 007004 005001
731 007006 017603 000006
732 007012 062703 000003
733 007016 005776 000006
734 007022 001414
```

```
RCV1ST: MOV  R1,-(SP) ;SAVE R1
        MOV  R3,-(SP) ;SAVE R3
        MOV  REGNUM,-(SP) ;SAVE LU REG NO.
        MOV  6(SP),SUBRPC
        SUB  #4,SUBRPC ;GET PC OF SUBROUTINE CALL
        MOV  #12,REGNUM ;SET LU REG NO. = 12
        CLR  R1 ;INIT CYCLE COUNT TO 0
        MOV  @6(SP),R3 ;GET CYCLE COUNT LIMIT
        ADD  #3,R3
        TST  @6(SP) ;SEE IF DESIRED CYCLES = 0
        BEQ  8$ ;BR IF YES
```

```

735 007024 004737 006232      JSR    PC,IACTIV      ;CHK FOR IACT = 0
736 007030 000000              0
737 007032 004737 005746      JSR    PC,ISIRDY      ;CHK FOR ICIR = 1, IRDY = 0
738 007036 000001              1
739 007040 004737 00642C      JSR    PC,RSEOM       ;CHK RSOM = 0, REOM = 0 IN AX0-16
740 007044 000000              0
741 007046 004737 004726      6$:   JSR    PC,STPLU      ;CLOCK LU FOR 1 CYCLE
742 007052 000001              1
743 007054 004737 004620      8$:   JSR    PC,WAIT50     ;ALLOW SILO DATA TO RIPPLE
744 007060 005201              INC    R1              ;INCREMENT CYCLE COUNT
745 007062 004737 003344      JSR    PC,READLU      ;READ REG 12
746 007066 132737 000020 002350 BITB   #IRDY,REDBYT     ;SEE IF IRDY = 1 YET
747 007074 001005              BNE   9$              ;BR IF IRDY = 1
748 007076 020103              CMP   R1,R3           ;SEE IF LIMIT EXCEEDED
749 007100 002762              BLT   6$              ;BR IF NOT YET
750 007102 004737 005746      JSR    PC,ISIRDY      ;CHK FOR ICIR = 1, IRDY = 1
751 007106 000003              3
752 007110 020176 000006      9$:   CMP   R1,@6(SP)     ;SEE IF LESS THAN REQUIRED CYCLES
753 007114 002003              BGE   12$            ;BR IF NOT
754 007116 004737 005746      JSR    PC,ISIRDY      ;CHK FOR ICIR = 1, IRDY = 0
755 007122 000001              1
756 007124 004737 006232      12$:  JSR    PC,IACTIV      ;CHK FOR IACT = 1
757 007130 000001              1
758 007132 004737 005746      JSR    PC,ISIRDY      ;CHK FOR ICIR = 1, IRDY = 1
759 007136 000003              3
760 007140 062766 000002 000006 ADD    #2,6(SP)        ;FIX UP RETURN PC
761 007146 012637 002364      MOV   (SP)+,REGNUM    ;RESTORE LU REG NO.
762 007152 012603              MOV   (SP)+,R3        ;RESTORE R3
763 007154 012601              MOV   (SP)+,R1        ;RESTORE R1
764 007156 005037 002336      CLR   SUBRPC          ;CLEAR SUBR CALL PC
765 007162 000207              RTS    PC              ;RETURN

```

766
767
768
769
770
771
772
773
774
775
776

```

:*****
:* STPERR - THIS SUBROUTINE LOADS THE CONTENTS OF THE FIRST WORD FOLLOWING THE
:* CALL INTO REG 17, AND SETS THE IERR BIT, AND CLOCKS THE LINE UNIT
:* FOR THE NO. OF CYCLES PASSED IN THE 2ND WORD FOLLOWING THE CALL. THEN,
:* IT RESTORES REG 17 TO ITS ORIGINAL CONTENTS, CLEARING THE IERR BIT.
:*****

```

```

777 007164 013746 002364      STPERR: MOV   REGNUM,-(SP) ;SAVE LU REG NO.
778 007170 012737 000017 002364 MOV   #17,REGNUM       ;SET LU REG NO. = 17
779 007176 017637 000002 002352 MOV   @2(SP),WRIBYT    ;
780 007204 152737 000002 002352 BISB  #IERR,WRIBYT     ;
781 007212 004737 003422      JSR    PC,WRITLU      ;SET IERR BIT IN REG 17
782 007216 062766 000002 000002 ADD    #2,2(SP)        ;INCREMENT SUBR ARGUMENT POINTER
783 007224 017637 000002 007236 MOV   @2(SP),3$        ;GET DESIRED NO. OF CYCLES
784 007232 004737 004726      JSR    PC,STPLU      ;CLOCK LU FOR DESIRED NO. OF CYCLES
785 007236 000000              .WORD 0               ;NO. OF CYCLES GOES HERE
786 007240 142737 000002 002352 3$:   BICB  #IERR,WRIBYT    ;
787 007246 004737 003422      JSR    PC,WRITLU      ;CLEAR IERR BIT IN REG 17
788 007252 062766 000002 000002 ADD    #2,2(SP)        ;FIX UP RETURN PC
789 007260 012637 002364      MOV   (SP)+,REGNUM    ;RESTORE LU REG NO.
790 007264 000207              RTS    PC              ;RETURN
791

```

792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840

007266 010146
007270 013746 002364
007274 016637 000004 002336
007302 162737 000004 002336
007310 017601 000004
007314 042701 170000
007320 004737 006672
007324 023727 002424 000347
007332 001005
007334 042701 000200
007340 042737 000200 002414
007346 120137 002414
007352 001445
007354 005037 002370
007360 110137 002370
007364 005037 002372
007370 113737 002414 002372
007376 012737 000011 002364
007404 004737 003344
007410 132737 000001 002350
007416 001410
007420 004737 004200
007424 104455
007426 000066
007430 014533
007432 016350
007434 000137 010116
007440 012737 000010 002364
007446 004737 004200
007452 104455
007454 000042
007456 013656
007460 020650
007462 000137 010116
007466 000301
007470 012737 000012 002364
007476 120137 002415
007502 001002
007504 000137 010072
007510 005037 002370

```

*****
* CKDATA - THIS SUBROUTINE READS THE RCV SILO AND COMPARES THE SILO ENTRY
* TO BITS 0-11 OF THE FIRST WORD FOLLOWING THE CALL. IF THERE IS A
* MISMATCH, THE ERROR IS REPORTED AND A RETURN IS MADE TO THE TEST AT THE
* ADDRESS CONTAINED IN RETADR. IF BIT 15 = 0 IN THE FIRST WORD
* FOLLOWING THE CALL, THE SUBROUTINE WILL NOT CHECK THE BCC BIT (SILO
* BIT 8). IF THERE ARE NO ERRORS, THE LINE UNIT IS CLOCKED FOR THE
* NUMBER OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
*****
CKDATA: MOV R1, -(SP) ;SAVE R1
MOV REGNUM, -(SP) ;SAVE LU REG NO.
MOV 4(SP), SUBRPC
SUB #4, SUBRPC ;GET PC OF SUBR CALL
MOV @4(SP), R1 ;GET EXPECTED SILO ENTRY
BIC #170000, R1 ;CLEAR UNUSED BITS FOR COMPARE
JSR PC, RDRXSI ;READ RCV SILO
CMP SAVLEN, #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
BNE 4$ ;BR IF CHAR LENGTH NOT = 7
BIC #BIT7, R1 ;MASK OFF 8TH BIT
BIC #BIT7, RXWORD
4$: CMPB R1, RXWORD ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
BEQ 6$ ;BR IF MATCH
CLR GOODAT
MOVB R1, GOODAT ;GET EXPECTED DATA
CLR BADDAT
MOVB RXWORD, BADDAT ;GET ACTUAL DATA
MOV #11, REGNUM ;SET REG NO. = 11
JSR PC, READLU ;READ REG 11
BITB #UNRR, REDBYT ;SEE IF TX UNDERRUN ERROR
BEQ 5$ ;BR IF NOT, TO REPORT DATA ERROR
JSR PC, GETALL ;GET REGS FOR PRINTOUT
;REPORT TX UNDERRUN ERROR
ERRDF 54, EM54, ERR4
TRAP C$ERDF
.WORD 54
.WORD EM54
.WORD ERR4

5$: JMP 36$ ;TAKE ERROR EXIT
MOV #10, REGNUM ;SET REG NO. = 10
JSR PC, GETALL ;GET REGS FOR PRINTOUT
;REPORT RCV'D DATA MISCOMPARE
ERRDF 34, EM34, ERR8
TRAP C$ERDF
.WORD 34
.WORD EM34
.WORD ERR8

6$: JMP 36$ ;TAKE ERROR EXIT
SWAB R1
MOV #12, REGNUM ;SET LU REG NO. FOR ERROR REPORTS
CMPB R1, RXWORD+1 ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
BNE 7$ ;BR IF MISMATCH
JMP 22$ ;CONTINUE

7$: CLR GOODAT

```

```

841 007514 110137 002370      MOVB    R1,GOODAT      ;SET EXPECTED DATA
842 007520 005037 002372      CLR     BADDAT
843 007524 113737 002415 002372  MOVB    RXWORD+1,BADDAT ;SET ACTUAL DATA
844 007532 032776 100000 000004  BIT     #BCCCHK,@4(SP) ;SEE IF BCC SHOULD BE IGNORED
845 007540 001433          BEQ     10$             ;BR IF YES
846 007542 132701 000001          BITB   #BCC,R1         ;SEE IF EXPECTED BIT = 1
847 007546 001014          BNE     8$             ;BR IF YES
848 007550 132737 000001 002415  BITB   #BCC,RXWORD+1 ;SEE IF ACTUAL BIT = 0
849 007556 001424          BEQ     10$             ;BR IF YES
850 007560 004737 004200          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
851          ;REPORT BCC NOT CLEARED
852 007564          ERRDF  35,EM35,ERR8
                                TRAP   C$ERDF
                                .WORD  35
                                .WORD  EM35
                                .WORD  ERR8
853 007574 000137 010116          JMP     36$            ;TAKE ERROR EXIT
854 007600 132737 000001 002415  8$:    BITB   #BCC,RXWORD+1 ;SEE IF ACTUAL BIT = 1
855 007606 001010          BNE     10$            ;BR IF YES
856 007610 004737 004200          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
857          ;REPORT BCC NOT SET
858 007614          ERRDF  36,EM36,ERR8
                                TRAP   C$ERDF
                                .WORD  36
                                .WORD  EM36
                                .WORD  ERR8
859 007624 000137 010116          JMP     36$            ;TAKE ERROR EXIT
860 007630 10$:
861 007630 132701 000002          BITB   #EBLK,R1       ;SEE IF EXPECTED BIT = 1
862 007634 001014          BNE     12$            ;BR IF YES
863 007636 132737 000002 002415  BITB   #EBLK,RXWORD+1 ;SEE IF ACTUAL BIT = 0
864 007644 001424          BEQ     14$            ;BR IF YES
865 007646 004737 004200          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
866          ;REPORT EBLK NOT CLEARED
867 007652          ERRDF  37,EM37,ERR8
                                TRAP   C$ERDF
                                .WORD  37
                                .WORD  EM37
                                .WORD  ERR8
868 007662 000137 010116          JMP     36$            ;TAKE ERROR EXIT
869 007666 132737 000002 002415  12$:   BITB   #EBLK,RXWORD+1 ;SEE IF ACTUAL BIT = 1
870 007674 001010          BNE     14$            ;BR IF YES
871 007676 004737 004200          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
872          ;REPORT EBLK NOT SET
873 007702          ERRDF  38,EM38,ERR8
                                TRAP   C$ERDF
                                .WORD  38
                                .WORD  EM38
                                .WORD  ERR8
874 007712 000137 010116          JMP     36$            ;TAKE ERROR EXIT
875 007716 14$:
876 007716 132701 000004          BITB   #RAB,R1        ;SEE IF EXPECTED BIT = 1
877 007722 001014          BNE     16$            ;BR IF YES
878 007724 132737 000004 002415  BITB   #RAB,RXWORD+1 ;SEE IF ACTUAL BIT = 0
879 007732 001424          BEQ     18$            ;BR IF YES
880 007734 004737 004200          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
881          ;REPORT RAB NOT CLEARED

```

```

882 007740          ERRDF 39,EM39,ERR8
      007740 104455
      007742 000047
      007744 013776
      007746 020650
883 007750 000137 010116          JMP 36$          ;TAKE ERROR EXIT
884 007754 132737 000004 002415 16$: BITB #RAB,RXWORD+1 ;SEE IF ACTUAL BIT = 1
885 007762 001010          BNE 18$          ;BR IF YES
886 007764 004737 004200          JSR PC,GETALL ;GET REGS FOR PRINTOUT
887          ;REPORT RAB NOT SET
888 007770          ERRDF 40,EM40,ERR8
      007770 104455
      007772 000050
      007774 014016
      007776 020650
889 010000 000137 010116          JMP 36$          ;TAKE ERROR EXIT
890 010004          18$:
891 010004 132701 000010          BITB #OVRR,R1 ;SEE IF EXPECTED BIT = 1
892 010010 001014          BNE 20$          ;BR IF YES
893 010012 132737 000010 002415 BITB #OVRR,RXWORD+1 ;SEE IF ACTUAL BIT = 0
894 010020 001424          BEQ 22$          ;BR IF YES
895 010022 004737 004200          JSR PC,GETALL ;GET REGS FOR PRINTOUT
896          ;REPORT OVRR NOT CLEARED
897 010026          ERRDF 41,EM41,ERR8
      010026 104455
      010030 000051
      010032 014032
      010034 020650
898 010036 000137 010116          JMP 36$          ;TAKE ERROR EXIT
899 010042 132737 000010 002415 20$: BITB #OVRR,RXWORD+1 ;SEE IF ACTUAL BIT = 1
900 010050 001010          BNE 22$          ;BR IF YES
901 010052 004737 004200          JSR PC,GETALL ;GET REGS FOR PRINTOUT
902          ;REPORT OVRR NOT SET
903 010056          ERRDF 42,EM42,ERR8
      010056 104455
      010060 000052
      010062 014053
      010064 020650
904 010066 000137 010116          JMP 36$          ;TAKE ERROR EXIT
905 010072          22$:
906 010072 062766 000002 000004 ADD #2,4(SP) ;INCR SUBROUTINE ARGUMENT POINTER
907 010100 017637 000004 010112 MOV @4(SP),24$ ;GET DESIRED CYCLE COUNT
908 010106 004737 004726          JSR PC,STPLU ;CLOCK LU FOR DESIRED CYCLES
909 010112 000000          24$: .WORD 0
910 010114 000407          BR 38$          ;TAKE ERROR-FREE EXIT
911 010116 011637 002364          36$: MOV (SP),REGNUM ;RESTORE LU REG NO.
912 010122 013706 002332          MOV PSTACK,SP ;RESTORE PROGRAM STACK TO BASE LEVEL
913 010126 013746 002346          MOV RETADR,-(SP) ;FIX UP ERROR RETURN PC
914 010132 000406          BR 40$
915 010134 062766 000002 000004 38$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
916 010142 012637 002364          MOV (SP)+,REGNUM ;RESTORE LU REG NO.
917 010146 012601          MOV (SP)+,R1 ;RESTORE R1
918 010150 005037 002336          40$: CLR SUBRPC ;CLEAR SUBROUTINE PC
919 010154 000207          RTS PC ;RETURN
920
921
922

```

TRAP C\$ERDF
.WORD 39
.WORD EM39
.WORD ERR8

TRAP C\$ERDF
.WORD 40
.WORD EM40
.WORD ERR8

TRAP C\$ERDF
.WORD 41
.WORD EM41
.WORD ERR8

TRAP C\$ERDF
.WORD 42
.WORD EM42
.WORD ERR8

```

923
924
925
926
927
928
929
930
931 010156 010146
932 010160 010346
933 010162 010446
934 010164 004737 010720
935 010170 000400
936 010172 000005
937 010174 012701 000002
938 010200 012704 002762
939 010204 012703 002557
940 010210 005037 002412
941 010214 112337 002412
942 010220 013724 002412
943 010224 004737 004646
944 010230 020327 002603
945 010234 103765
946 010236 005301
947 010240 001361
948 010242 052764 100400 177776
949 010250 012737 001000 002412
950 010256 004737 004646
951 010262 004737 004646
952 010266 012604
953 010270 012603
954 010272 012601
955 010274 000207
956
957
958
959
960
961
962
963
964
965
966 010276 013746 002366
967 010302 013746 002364
968 010306 012737 000004 002366
969 010314 017637 000004 002360
970 010322 005037 002362
971 010326 004737 003764
972 010332 012737 000017 002364
973 010340 062766 000002 000004
974 010346 017637 000004 002352
975 010354 004737 003422
976 010360 012737 000006 002366
977 010366 062766 000002 000004
978 010374 017637 000004 002360
979 010402 062766 000002 000004
    
```

```

*****
;* LODATA - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH 5 SOM'S, THE DATA
;* IN PATTERN A REPEATED 2 TIMES (40 CHARS), AND 2 EOM'S. IN ADDITION, THE
;* DATA CHARS ARE ALSO LOADED INTO THE RECEIVED MSG BUFFER (RCVBUF:), AS
;* EXPECTED DATA FOR LATER COMPARISON.
*****
LODATA: MOV R1,-(SP) ;SAVE R1
        MOV R3,-(SP) ;SAVE R3
        MOV R4,-(SP) ;SAVE R4
        JSR PC,LODSIL ;LOAD 5 SOM'S INTO TX SILO
        TXSOM
        5
        MOV #2,R1 ;INIT COUNTER
        MOV #RCVBUF,R4 ;GET POINTER TO RCV BUF
3$: MOV #PATA,R3 ;GET POINTER TO DATA PATTERN
6$: CLR TXWORD
    MOVB (R3)+,TXWORD ;GET A DATA CHAR
    MOV TXWORD,(R4)+ ;LOAD A DATA CHAR INTO RCV BUF
    JSR PC,LDTXSI ;LOAD DATA CHAR INTO TX SILO
    CMP R3,#PATB ;SEE IF AT END OF PATTERN A YET
    BLO 6$ ;BR IF NOT YET
    DEC R1 ;DECREMENT COUNTER
    BNE 3$ ;BR IF NOT DONE YET
    BIS #CRCCHK!RXBCC,-2(R4) ;SET UP TO CHK BCC = 1 ON LAST DATA CHAR
    MOV #TXEOM,TXWORD
    JSR PC,LDTXSI ;LOAD AN EOM INTO TX SILO
    JSR PC,LDTXSI ;LOAD ANOTHER EOM
    MOV (SP)+,R4 ;RESTORE R4
    MOV (SP)+,R3 ;RESTORE R3
    MOV (SP)+,R1 ;RESTORE R1
    RTS PC ;RETURN
    
```

```

*****
;* SETUP - THIS SUBROUTINE LOADS THE FIRST WORD AFTER THE CALL INTO AX2-15
;* (SYNCH CHAR), LOADS THE SECOND WORD AFTER THE CALL INTO REG 17
;* LOADS THE THIRD WORD INTO AX3-15, AND LOADS THE FOURTH WORD INTO AX3-16.
*****
SETUP: MOV AXNUM,-(SP) ;SAVE AX BYTE NO.
        MOV REGNUM,-(SP) ;SAVE LU REG NO.
        MOV #4,AXNUM ;SET AX BYTE NO. FOR AX2
        MOV @4(SP),WAX15
        CLR WAX16
        JSR PC,WRITAX ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
        MOV #17,REGNUM ;SET LU REG NO. = 17
        ADD #2,4(SP) ;INCREMENT ARGUMENT POINTER
        MOV @4(SP),WRIBYT
        JSR PC,WRITLU ;LOAD REG 17
        MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
        ADD #2,4(SP) ;INCREMENT ARGUMENT POINTER
        MOV @4(SP),WAX15
        ADD #2,4(SP) ;INCREMENT ARGUMENT POINTER
    
```

```

980 010410 017637 000004 002362      MOV      @4(SP),WAX16
981 010416 013737 002362 002424      MOV      WAX16,SAVLEN      ;STORE TX AND RCV CHAR LENGTHS
982 010424 142777 000010 172006      BICB     #LLOOP,@BSEL1    ;CLEAR LLOOP
983 010432 004737 003764      JSR      PC,WRITAX        ;LOAD AX3-15, AX3-16
984 010436 152777 000010 171774      BISB     #LLOOP,@BSEL1    ;SET LLOOP
985 010444 062766 000002 000004      ADD      #2,4(SP)         ;FIX RETURN PC
986 010452 012637 002364      MOV      (SP)+,REGNUM     ;RESTORE LU REG NO.
987 010456 012637 002366      MOV      (SP)+,AXNUM      ;RESTORE AX BYTE NO.
988 010462 005037 002336      CLR      SUBRPC          ;CLEAR SUBROUTINE PC STORAGE
989 010466 000207      RTS      PC              ;RETURN

```

990
991
992
993
994
995
996
997
998
999

```

:*****
;* LODMSG - THIS SUBROUTINE LOADS THE NO. OF WORDS PASSED IN THE SECOND WORD
;* FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
;* WORD FOLLOWING THE CALL, INTO THE TRANSMITTER SILO.
:*****

```

```

1000 010470 010146      LODMSG: MOV      R1,-(SP)      ;SAVE R1
1001 010472 010246      MOV      R2,-(SP)      ;SAVE R2
1002 010474 017601 000004      MOV      @4(SP),R1      ;GET MSG POINTER INTO R1
1003 010500 062766 000002 000004      ADD      #2,4(SP)      ;INCR ARG POINTER
1004 010506 017602 000004      MOV      @4(SP),R2      ;GET WORD COUNT INTO R2
1005 010512 062766 000002 000004      ADD      #2,4(SP)      ;FIX UP RETURN PC
1006 010520 012137 002412      6$: MOV      (R1)+,TXWORD ;GET NEXT MSG WORD
1007 010524 004737 004646      JSR      PC,LDTXSI     ;LOAD A WORD INTO TX SILO
1008 010530 005302      DEC      R2            ;DECR COUNT
1009 010532 001372      BNE     6$            ;BR IF NOT DONE YET
1010 010534 004737 004620      JSR      PC,WAIT50    ;WAIT FOR SILO TO RIPPLE
1011 010540 012602      MOV      (SP)+,R2      ;RESTORE R2
1012 010542 012601      MOV      (SP)+,R1      ;RESTORE R1
1013 010544 000207      RTS      PC            ;RETURN

```

1014
1015
1016
1017
1018
1019

```

:*****
;* LDBYTS - THIS SUBROUTINE LOADS THE NO. OF BYTES PASSED IN THE SECOND WORD
;* FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
;* WORD FOLLOWING THE CALL, INTO THE LOW BYTE OF THE TX SILO. FOR EACH
;* BYTE LOADED, A 0 IS LOADED INTO THE HI 4 BITS OF THE TX SILO.
:*****

```

```

1025 010546 010146      LDBYTS: MOV      R1,-(SP)      ;SAVE R1
1026 010550 010246      MOV      R2,-(SP)      ;SAVE R2
1027 010552 017601 000004      MOV      @4(SP),R1      ;GET DATA POINTER INTO R1
1028 010556 062766 000002 000004      ADD      #2,4(SP)      ;INCR ARGUMENT POINTER
1029 010564 017602 000004      MOV      @4(SP),R2      ;GET BYTE COUNT INTO R2
1030 010570 062766 000002 000004      ADD      #2,4(SP)      ;FIX UP RETURN PC
1031 010576 112137 002412      6$: MOV      (R1)+,TXWORD ;GET NEXT DATA BYTE
1032 010602 105037 002413      CLRB    TXWORD+1      ;CLEAR HI BYTE
1033 010606 004737 004646      JSR      PC,LDTXSI     ;LOAD A SILO ENTRY
1034 010612 005302      DEC      R2            ;DECR BYTE COUNT
1035 010614 001370      BNE     6$            ;BR IF NOT DONE YET
1036 010616 004737 004620      JSR      PC,WAIT50    ;WAIT FOR SILO TO RIPPLE

```

1037 010622 012602
1038 010624 012601
1039 010626 000207

MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

1040
1041
1042
1043
1044
1045
1046
1047
1048
1049

* LDMSG1 - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH MSG1, AND LOADS
* THE DATA CHARS INTO THE RCV MSG BUFFER (RCVBUF:), AS EXPECTED DATA
* FOR LATER COMPARISON.

1050 010630 010146
1051 010632 010246
1052 010634 004737 010720
1053 010640 000400
1054 010642 000003
1055 010644 004737 010470
1056 010650 002670
1057 010652 000010
1058 010654 012701 002674
1059 010660 012702 002762
1060 010664 012122
1061 010666 020127 002706
1062 010672 103774
1063 010674 052762 100400 177776
1064 010702 012722 000160
1065 010706 012722 000034
1066 010712 012602
1067 010714 012601
1068 010716 000207

LDMSG1: MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
JSR PC,LODSIL ;LOAD 3 SOM'S INTO TX SILO
TXSOM
3
JSR PC,LODMSG ;LOAD MSG1 INTO TX SILO (WITH 2 SOM'S, 1 EOM)
MSG1
8.
MOV #MSG1+4,R1 ;GET POINTER TO MSG1
MOV #RCVBUF,R2 ;GET POINTER TO MSG BUF
3\$: MOV (R1)+,(R2)+ ;LOAD A CHAR INTO MSG BUF
CMP R1,#MSG1+14. ;SEE IF DID LAST DATA CHAR YET
BLO 3\$;BR IF NOT
BIS #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
MOV #160,(R2)+ ;LOAD HI CRC BYTE
MOV #034,(R2)+ ;LOAD LO CRC BYTE
MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

1069
1070
1071
1072
1073
1074
1075
1076
1077
1078

* LODSIL - THIS SUBROUTINE REPEATEDLY LOADS THE DATA PASSED IN THE FIRST WORD
* FOLLOWING THE CALL INTO THE TX SILO, FOR THE NO. OF TIMES PASSED IN THE
* SECOND WORD FOLLOWING THE CALL.

1079 010720 010146
1080 010722 017637 000002 002412
1081 010730 062766 000002 000002
1082 010736 017601 000002
1083 010742 004737 004646
1084 010746 005301
1085 010750 001374
1086 010752 004737 004620
1087 010756 062766 000002 000002
1088 010764 012601
1089 010766 000207

LODSIL: MOV R1,-(SP) ;SAVE R1
MOV @2(SP),TXWORD ;GET DATA
ADD #2,2(SP) ;INCR ARGUMENT POINTER
MOV @2(SP),R1 ;GET COUNT
3\$: JSR PC,LDTXSI ;LOAD WORD INTO TX SILO
DEC R1 ;DECR COUNT
BNE 3\$;BR IF NOT ALL LOADED YET
JSR PC,WAIT50 ;ALLOW SILO DATA TO RIPPLE
ADD #2,2(SP) ;FIX UP RETURN PC
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

1090
1091
1092
1093

1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113 010770 013746 002364
1114 010774 013746 002366
1115 011000 010246
1116 011002 016637 000006 002336
1117 011010 162737 000004 002336
1118 011016 012737 000333 002422
1119 011024 032776 100000 000006
1120 011032 001405
1121 011034 005737 002462
1122 011040 001074
1123 011042 000137 012074
1124
1125 011046 023727 002462 000004
1126 011054 001466
1127 011056 142777 000010 171354
1128 011064 012737 000006 002366
1129 011072 004737 003576
1130
1131
1132
1133 011076 027627 000006 000010
1134 011104 001422
1135 011106 027627 000006 000020
1136 011114 001512
1137 011116 027627 000006 000200
1138 011124 001002
1139 011126 000137 011426
1140 011132 027627 000006 000100
1141 011140 001002
1142 011142 000137 011462
1143 011146 000137 011554
1144
1145
1146 011152 005737 002462
1147 011156 001050
1148 011160 023727 002464 000004
1149 011166 002405
1150 011170 042737 000010 002422

```

*****
* CKLPBK - THIS SUBROUTINE DETERMINES IF THE TEST CALLING IT CAN BE RUN. THE
* TEST PASSES THE DESIRED MODEM INTERFACE TYPE IN THE WORD FOLLOWING THE
* CALL, AND IF THE PROPER EXTERNAL LOOPBACK HAS BEEN PROVIDED BY THE
* OPERATOR FOR THAT INTERFACE, AND IF THE BAUD RATE IS CORRECT, A NORMAL
* RETURN IS MADE TO RUN THE TEST. IF NOT, A RETURN IS MADE TO THE TEST,
* AT THE ADDRESS IN RETADR (RETADR CONTAINS THE TEST EXIT ADDRESS, SO
* THE TEST GETS SKIPPED).
* IF BIT 15 IS SET IN THE WORD FOLLOWING THE CALL, THE TEST WILL NOT
* BE RUN UNLESS THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. IF THE
* TEST IS TO BE RUN, THE SUBROUTINE RETURNS THE MODEM SELECT BITS FOR
* AX3-15 IN MODINT. IF NECESSARY, THE SUBROUTINE WILL SET MAINT1, MAINT2,
* OR DTR.
* IF THE PROGRAM PASSES '0' IN THE WORD FOLLOWING THE CALL, THE SUBROUTINE
* WILL ATTEMPT TO RUN WHICHEVER MODEM INTERFACE IS SELECTED BY CABLE,
* SWITCH, OR TEST CONNECTOR. IF SUCCESSFUL, THE SELECTED INTERFACE WILL
* BE PASSED BACK TO THE TEST IN MODINT.
*****
CKLPBK: MOV     REGNUM, -(SP)      ;SAVE REG NO.
        MOV     AXNUM, -(SP)    ;SAVE AX BYTE NO.
        MOV     R2, -(SP)      ;SAVE R2
        MOV     6(SP), SUBRPC   ;
        SUB     #4, SUBRPC      ;GET PC OF SUBR CALL
        MOV     #1422!XYZ!V35!INTGRL!OP!TEST, MODINT ;INIT MODEM SELECT BITS
        BIT     #TCCHEK, @6(SP) ;SEE IF H3254,5 CHECK IS DESIRED
        BEQ     1$              ;BR IF NOT
        TST     TSTCON          ;SEE IF H3254,5 INSTALLED
        BNE     7$              ;BR IF NOT, TO SKIP TEST
        JMP     32$             ;BR TO RUN TEST
;IF NO EXTERNAL LPBK, SKIP TEST
1$:     CMP     TSTCON, #4      ;SEE IF NO LPBK
        BEQ     7$              ;BR IF NO LPBK, TO SKIP TEST
        BICB   #LULOOP, @BSEL1 ;CLEAR LULOOP
        MOV     #6, AXNUM       ;SET AX BYTE NO. FOR AX3-15
        JSR    PC, READAX      ;READ AX3-15
;*** SEE IF AN INTERFACE IS REQUESTED ***
        CMP     @6(SP), #INTGRL ;SEE IF INTEGRAL MODEM REQUESTED
        BEQ     4$              ;BR IF INTGRL MODEM REQUESTED
        CMP     @6(SP), #V35    ;SEE IF V.35 REQUESTED
        BEQ     10$             ;BR IF V.35 REQUESTED
        CMP     @6(SP), #1422   ;SEE IF 422 REQUESTED
        BNE     2$              ;BR IF 422 NOT REQUESTED
        JMP     14$             ;422 REQUESTED
2$:     CMP     @6(SP), #XYZ    ;SEE IF XYZ REQUESTED
        BNE     3$              ;BR IF XYZ NOT REQUESTED
        JMP     17$             ;XYZ REQUESTED
3$:     JMP     21$             ;NONE REQUESTED, FIND AN INTERFACE TO TEST
;SEE IF INTEGRAL MODEM CAN BE RUN
4$:     TST     TSTCON          ;SEE IF H3254 AND H3255 USED
        BNE     8$              ;BR IF NOT
5$:     CMP     BDRATE, #4      ;SEE IF BAUD RATE > OR = 56K
        BLT     6$              ;BR IF BAUD RATE TOO SLOW FOR INTGRL MODM
        BIC     #INTGRL, MODINT ;ASSERT INTEGRAL MODEM

```

```

1151 011176 000137 012074          JMP      32$          ;GO TO RUN TEST
1152          ;PRINT 'BAUD RATE INCORRECT'
1153 011202          6$:
1154 011202 023727 002402 000001    CMP      STARES,#1    ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
1155 011210 001026          BNE     40$          ;BR IF NOT, TO SKIP PRINTING
1156 011212          PRINTF #FMT25
1156 011212 012746 013127          MOV     #FMT25,-(SP)
1156 011216 012746 000001          MOV     #1,-(SP)
1156 011222 010600          MOV     SP,R0
1156 011224 104417          TRAP   C$PNTF
1156 011226 062706 000004          ADD     #4,SP
1157          ;PRINT 'TEST XX NOT RUN'
1158 011232          7$:
1159 011232 023727 002402 000001    CMP      STARES,#1    ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
1160 011240 001012          BNE     40$          ;BR IF NOT, TO SKIP PRINTING
1161 011242          PRINTF #FMT19,TSTNUM
1161 011242 013746 002434          MOV     TSTNUM,-(SP)
1161 011246 012746 013076          MOV     #FMT19,-(SP)
1161 011252 012746 000002          MOV     #2,-(SP)
1161 011256 010600          MOV     SP,R0
1161 011260 104417          TRAP   C$PNTF
1161 011262 062706 000006          ADD     #6,SP
1162 011266 013766 002346 000006 40$:  MOV     RETADR,6(SP)  ;SET TEST EXIT ADDRESS FOR ERRORS
1163 011274 000137 012102          JMP     33$          ;GO TO SKIP TEST
1164 011300 032737 000010 002354 8$:  BIT     #INTGRL,RAX15 ;SEE IF INTEGRAL MODEM IS SELECTED
1165 011306 001324          BNE     5$          ;BR IF YES, TO CHECK BAUD RATE
1166          ;PRINT 'MODEM INTERFACE NOT SELECTED'
1167 011310          9$:
1168 011310 023727 002402 000001    CMP      STARES,#1    ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
1169 011316 001363          BNE     40$          ;BR IF NOT, TO SKIP PRINTING
1170 011320          PRINTF #FMT26
1170 011320 012746 013157          MOV     #FMT26,-(SP)
1170 011324 012746 000001          MOV     #1,-(SP)
1170 011330 010600          MOV     SP,R0
1170 011332 104417          TRAP   C$PNTF
1170 011334 062706 000004          ADD     #4,SP
1171 011340 000734          BR      7$          ;GO TO PRINT 'TEST NOT RUN'
1172
1173          ;SEE IF V.35 CAN BE RUN
1174 011342 032737 000200 002354 10$: BIT     #I422,RAX15  ;SEE IF 422 IS SELECTED
1175 011350 001357          BNE     9$          ;BR IF YES, TO SKIP TEST
1176 011352 005737 002462          TST    TSTCON        ;SEE IF H3254 AND H3255 USED
1177 011356 001010          BNE     12$         ;BR IF H3254 AND H3255 NOT USED
1178 011360 012737 000004 002352 11$: MOV     #MAINT2,WRIBYT ;SET MAINT2 FOR MANUFACTURING TEST CONN.
1179 011366 042737 000020 002422    BIC     #V35,MODINT   ;ASSERT V.35
1180 011374 000137 011656          JMP     42$          ;GO SET DTR AND RUN THE TEST
1181 011400 032737 000020 002354 12$: BIT     #V35,RAX15  ;SEE IF V.35 IS SELECTED
1182 011406 001405          BEQ     13$         ;BR IF NOT
1183 011410 042737 000020 002422    BIC     #V35,MODINT   ;ASSERT V.35
1184 011416 000137 011652          JMP     27$         ;GO SET DTR AND RUN THE TEST
1185 011422 000137 011310          13$:  JMP     9$          ;WRONG INTRFCE, GO SKIP TEST
1186
1187          ;SEE IF 422 CAN BE RUN
1188 011426 005737 002462          14$:  TST    TSTCON        ;SEE IF H3254 AND H3255 USED
1189 011432 001005          BNE     16$         ;BR IF NOT
1190 011434 042737 000200 002422 15$:  BIC     #I422,MODINT  ;ASSERT 422
1191 011442 000137 011652          JMP     27$         ;GO TO RUN TEST

```

```

1192 011446 032737 000200 002354 16$: BIT #I422,RAX15 ;SEE IF 422 IS SELECTED
1193 011454 001367 BNE 15$ ;IF YES, GO ASSERT 422 AND RUN TEST
1194 011456 000137 011310 JMP 9$ ;WRONG INTRFCE, GO SKIP TEST
1195
1196 ;SEE IF XYZ CAN BE RUN
1197 011462 032737 000200 002354 17$: BIT #I422,RAX15 ;SEE IF 422 IS SELECTED
1198 011470 001402 BEQ 18$ ;BR IF NOT
1199 011472 000137 011310 JMP 9$ ;WRONG INTRFCE, GO SKIP TEST
1200 011476 032737 000100 002354 18$: BIT #XYZ,RAX15 ;SEE IF XYZ IS SELECTED
1201 011504 001002 BNE 19$ ;BR IF YES
1202 011506 000137 011310 JMP 9$ ;WRONG INTRFCE, GO SKIP TEST
1203 011512 023727 002464 000004 19$: CMP BDRATE,#4 ;SEE IF BAUD RATE < OR = 56K
1204 011520 003402 BLE 20$ ;BR IF YES
1205 011522 000137 011202 JMP 6$ ;BAUD RATE TOO FAST FOR XYZ
1206 011526 042737 000100 002422 20$: BIC #XYZ,MODINT ;ASSERT XYZ
1207 011534 005737 002462 TST TSTCON ;SEE IF H3254,5 BEING USED
1208 011540 001044 BNE 27$ ;BR IF NOT
1209 011542 012737 000004 002352 MOV #MAINT2,WRIBYT
1210 011550 000137 011656 JMP 42$ ;GO SET MAINT2 FOR MANUFACTURING TEST CONN.
1211
1212 ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
1213
1214 011554 032737 000010 002354 21$: BIT #INTGRL,RAX15 ;SEE IF INTEGRAL MODEM SELECTED
1215 011562 001402 BEQ 22$ ;BR IF NOT
1216 011564 000137 011160 JMP 5$ ;SEE IF INTEGRAL MODEM CAN BE RUN
1217 011570 032737 000020 002354 22$: BIT #V35,RAX15 ;SEE IF V.35 SELECTED
1218 011576 001402 BEQ 23$ ;BR IF NOT
1219 011600 000137 011342 JMP 10$ ;GO SEE IF V.35 CAN BE RUN
1220 011604 032737 000200 002354 23$: BIT #I422,RAX15 ;SEE IF 422 SELECTED
1221 011612 001402 BEQ 24$ ;BR IF NOT
1222 011614 000137 011434 JMP 15$ ;GO ASSERT AND RUN 422
1223 011620 005737 002462 24$: TST TSTCON ;SEE IF H3254 AND H3255 USED
1224 011624 001002 BNE 25$ ;BR IF NOT
1225 011626 000137 011360 JMP 11$ ;GO ASSERT AND RUN V.35 BY DEFAULT
1226 011632 032737 000100 002354 25$: BIT #XYZ,RAX15 ;SEE IF XYZ SELECTED
1227 011640 001402 BEQ 26$ ;BR IF NOT
1228 011642 000137 011512 JMP 19$ ;GO SEE IF XYZ CAN BE RUN
1229 011646 000137 011310 26$: JMP 9$ ;GO SKIP TEST
1230
1231 ;*** SET MAINT1 OR MAINT2 IF NEEDED, AND SET DTR ***
1232
1233 ;SET MAINT1 IF LOCAL LOOPBACK DESIRED
1234 011652 005037 002352 27$: CLR WRIBYT ;INIT DATA TO BE WRITTEN
1235 011656 012737 000013 002364 42$: MOV #13,REGNUM ;SET REG NO. = 13
1236 011664 023727 002462 000002 CMP TSTCON,#2 ;SEE IF MODEM LOCAL LPBK DESIRED
1237 011672 001032 BNE 29$ ;BR IF NOT, TO CHK REMOTE LPBK
1238 011674 112737 000010 002352 MOVB #MAINT1,WRIBYT
1239 011702 004737 003422 JSR PC,WRITLU ;SET MAINT1 IN REG 13
1240 011706 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1241 011714 012702 000000 MOV #0,R2 ;INIT TIMER
1242 011720 004737 003344 28$: JSR PC,READLU ;READ REG 17
1243 011724 132737 000004 002350 BITB #TESTMD,REDBYT ;SEE IF TEST MODE BIT SET IN REG 17 YET
1244 011732 001050 BNE 31$ ;BR IF YES, TO SET DTR
1245 011734 005202 INC R2 ;INCREMENT TIMER
1246 011736 001370 BNE 28$ ;BR IF NO TIME-OUT YET
1247 011740 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1248 ;REPORT TEST MODE NOT SET BY MAINT1

```

```
1249 011744 ERRDF 52,EM52,ERR4
      011744 104455
      011746 000064 TRAP C$ERDF
      011750 014443 .WORD 52
      011752 016350 .WORD EM52
      011754 000137 011232 .WORD ERR4
1250 011754 000137 011232 JMP 7$ ;BR TO SKIP TEST
1251 ;SET MAINT2 IF REMOTE LOOPBACK DESIRED
1252 011760 023727 002462 000003 29$: CMP TSTCON,#3 ;SEE IF MODEM REMOTE LPBK DESIRED
1253 011766 001032 BNE 31$ ;BR IF NOT, TO SET DTR
1254 011770 112737 000004 002352 MOVB #MAINT2,WRIBYT
1255 011776 004737 003422 JSR PC,WRITLU ;SET MAINT2 IN REG 13
1256 012002 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1257 012010 012702 000000 MOV #0,R2 ;INIT TIMER
1258 012014 004737 003344 30$: JSR PC,READLU ;READ REG 17
1259 012020 132737 000004 002350 BITB #TESTMD,REDBYT ;SEE IF TEST MODE BIT SET IN REG 17 YET
1260 012026 001012 BNE 31$ ;BR IF YES, TO SET DTR
1261 012030 005202 INC R2 ;INCREMENT TIMER
1262 012032 001370 BNE 30$ ;BR IF NO TIME-OUT YET
1263 012034 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1264 ;REPORT TEST MODE NOT SET BY MAINT2
1265 ERRDF 53,EM53,ERR4
      012040 104455 TRAP C$ERDF
      012042 000065 .WORD 53
      012044 014477 .WORD EM53
      012046 016350 .WORD ERR4
1266 012050 000137 011232 JMP 7$ ;BR TO SKIP TEST
1267 ;SET DTR
1268 012054 012737 000013 002364 31$: MOV #13,REGNUM ;SET REG NO. = 13
1269 012062 152737 000100 002352 BISB #DTR,WRIBYT ;SET DTR BIT TO BE WRITTEN
1270 012070 004737 003422 JSR PC,WRITLU ;LOAD REG 13
1271
1272
1273 012074 062766 000002 000006 32$: ADD #2,6(SP) ;INCREMENT RETURN ADRS
1274
1275
1276 012102 012602 33$: MOV (SP)+,R2 ;RESTORE R2
1277 012104 012637 002366 MOV (SP)+,AXNUM ;RESTORE AX BYTE NO.
1278 012110 012637 002364 MOV (SP)+,REGNUM ;RESTORE LU REG NO.
1279 012114 152777 000010 170316 BISB #LULoop,@BSEL1 ;SET LULoop
1280 012122 005037 002336 CLR SUBRPC ;CLEAR SUBROUTINE CALL PC
1281 012126 000207 RTS PC ;RETURN
1282
1283
1284
1285
1286
1287
1288 ;*****
1289 ;* CHKABT - THIS SUBROUTINE READS AX0-16 AND CHECKS FOR RAB, REOM SET. IF
1290 ;* EITHER IS NOT SET, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
1291 ;* CONTAINED IN RETADR.
1292 ;*****
1292 012130 013746 002366 CHKABT: MOV AXNUM,-(SP) ;SAVE AX BYTE NO.
1293 012134 016637 000002 002336 MOV 2(SP),SUBRPC
1294 012142 162737 000004 002336 SUB #4,SUBRPC ;GET PC OF SUBROUTINE CALL
1295 012150 005037 002366 CLR AXNUM ;SET AX0 ADDRESS
1296 012154 004737 003576 JSR PC,READAX ;READ REG AX0
1297 012160 032737 000004 002356 BIT #RABT,RAX16 ;CHK FOR RAB SET
```

```
1298 012166 001007          BNE      6$          ;BR IF RAB SET
1299 012170 004737 004200   JSR      PC,GETALL   ;GET REGS FOR PRINTOUT
1300          ;REPORT RAB NOT SET
1301 012174          ERRDF  40,EM40,ERR6
                                TRAP      C$ERDF
                                .WORD    40
                                .WORD    EM40
                                .WORD    ERR6
1302 012204 000412          BR       8$
1303 012206 032737 000002 002356 6$:   BIT      #REOM,RAX16 ;CHK FOR REOM SET
1304 012214 001015          BNE      9$          ;BR IF REOM SET
1305 012216 004737 004200   JSR      PC,GETALL   ;GET REGS FOR PRINTOUT
1306          ;REPORT REOM NOT SET
1307 012222          ERRDF  31,EM31,ERR6
                                TRAP      C$ERDF
                                .WORD    31
                                .WORD    EM31
                                .WORD    ERR6
1308 012232 011637 002366   8$:   MOV      (SP),AXNUM  ;RESTORE AX BYTE NO.
1309 012236 013706 002332   MOV      PSTACK,SP   ;RESTORE STACK POINTER TO BASE LEVEL
1310 012242 013746 002346   MOV      RETADR,-(SP) ;FIX ERROR RETURN PC
1311 012246 000402          BR       16$
1312 012250 012637 002366   9$:   MOV      (SP)+,AXNUM ;RESTORE AX BYTE NO.
1313 012254 000207          16$:   RTS      PC        ;RETURN
1314
1315
1316
1317
1318
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

.SBTTL GLOBAL ERROR REPORT SECTION

:/

:/ THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES

:/ THAT ARE USED IN MORE THAN ONE TEST.

:/

012256	045	124	045	FMT1: .ASCIZ /%T%06%N/
012261	117	066	045	
012264	116	000		
012266	045	116	045	FMT2: .ASCIZ /%N%AFAILING REG: /
012271	101	106	101	
012274	111	114	111	
012277	116	107	040	
012302	122	105	107	
012305	072	040	000	
012310	045	101	105	FMT3: .ASCIZ /%AEXPECTED: %03%S5%AACTUAL: %03%N/
012313	130	120	105	
012316	103	124	105	
012321	104	072	040	
012324	045	117	063	
012327	045	123	065	
012332	045	101	101	
012335	103	124	125	
012340	101	114	072	
012343	040	045	117	
012346	063	045	116	
012351	000			
012352	045	116	045	FMT4: .ASCIZ /%N%T%N%T%N/
012355	124	045	116	
012360	045	124	045	
012363	116	000		
012365	045	117	063	FMT5: .ASCIZ /%03%S5%03%S5%03%S5%03%N/
012370	045	123	065	
012373	045	117	063	
012376	045	123	065	
012401	045	117	063	
012404	045	123	065	
012407	045	117	063	
012412	045	116	000	
012415	045	123	064	FMT6: .ASCIZ /%S4%03%S5%03%S5%03%S5%03%N/
012420	045	117	063	
012423	045	123	065	
012426	045	117	063	
012431	045	123	065	
012434	045	117	063	
012437	045	123	065	
012442	045	117	063	
012445	045	116	000	
012450	045	124	045	FMT7: .ASCIZ /%T%02%N/
012453	117	062	045	
012456	116	000		
012460	045	101	105	FMT8: .ASCIZ /%AEXTENDED REG AX%01%A-%T%N/
012463	130	124	105	
012466	116	104	105	
012471	104	040	122	

	012474	105	107	040	
	012477	101	130	045	
	012502	117	061	045	
	012505	101	055	045	
	012510	124	045	116	
	012513	000			
17	012514	045	124	045	FMT9: .ASCIZ /%T%N/
	012517	116	000		
18	012521	045	101	120	FMT10: .ASCIZ /%APC OF SUBR CALL: %06%N/
	012524	103	040	117	
	012527	106	040	123	
	012532	125	102	122	
	012535	040	103	101	
	012540	114	114	072	
	012543	040	045	117	
	012546	066	045	116	
	012551	000			
19	012552	045	101	122	FMT11: .ASCIZ /%AREG %02%A LOADED WITH: %03%N/
	012555	105	107	040	
	012560	045	117	062	
	012563	045	101	040	
	012566	114	117	101	
	012571	104	105	104	
	012574	040	127	111	
	012577	124	110	072	
	012602	040	045	117	
	012605	063	045	116	
	012610	000			
20	012611	045	101	115	FMT12: .ASCIZ /%AM8203 SW PACK #1 (REG 11) = %03%N/
	012614	070	062	060	
	012617	063	040	123	
	012622	127	040	120	
	012625	101	103	113	
	012630	040	043	061	
	012633	040	050	122	
	012636	105	107	040	
	012641	061	061	051	
	012644	040	075	040	
	012647	045	117	063	
	012652	045	116	000	
21	012655	045	101	115	FMT13: .ASCIZ /%AM8203 SW PACK #2 (REG 15) = %03%N/
	012660	070	062	060	
	012663	063	040	123	
	012666	127	040	120	
	012671	101	103	113	
	012674	040	043	062	
	012677	040	050	122	
	012702	105	107	040	
	012705	061	065	051	
	012710	040	075	040	
	012713	045	117	063	
	012716	045	116	000	
22	012721	045	101	115	FMT14: .ASCIZ /%AM8203 SW PACK #3 (REG 16) = %03%N/
	012724	070	062	060	
	012727	063	040	123	
	012732	127	040	120	
	012735	101	103	113	

	012740	040	043	063	
	012743	040	050	122	
	012746	105	107	040	
	012751	061	066	051	
	012754	040	075	040	
	012757	045	117	063	
	012762	045	116	000	
23	012765	045	101	115	FMT15: .ASCIZ /%MODEM INTERFACE REG (AX3-15) = %03%N/
	012770	117	104	105	
	012773	115	040	111	
	012776	116	124	105	
	013001	122	106	101	
	013004	103	105	040	
	013007	122	105	107	
	013012	040	050	101	
	013015	130	063	055	
	013020	061	065	051	
	013023	040	075	040	
	013026	045	117	063	
	013031	045	116	000	
24	013034	045	116	045	FMT18: .ASCIZ /%N%AFOR DEVICE AT ADRS %06%A ,%N/
	013037	101	106	117	
	013042	122	040	104	
	013045	105	126	111	
	013050	103	105	040	
	013053	101	124	040	
	013056	101	104	122	
	013061	123	040	040	
	013064	045	117	066	
	013067	045	101	040	
	013072	054	045	116	
	013075	000			
25	013076	045	116	045	FMT19: .ASCIZ /%N%ATEST %D2%A NOT RUN%N/
	013101	101	124	105	
	013104	123	124	040	
	013107	045	104	062	
	013112	045	101	040	
	013115	116	117	124	
	013120	040	122	125	
	013123	116	045	116	
	013126	000			
26	013127	045	116	045	FMT25: .ASCIZ /%N%ABAUD RATE INCORRECT/
	013132	101	102	101	
	013135	125	104	040	
	013140	122	101	124	
	013143	105	040	111	
	013146	116	103	117	
	013151	122	122	105	
	013154	103	124	000	
27	013157	045	116	045	FMT26: .ASCIZ /%N%MODEM INTERFACE NOT SELECTED/
	013162	101	115	117	
	013165	104	105	115	
	013170	040	111	116	
	013173	124	105	122	
	013176	106	101	103	
	013201	105	040	116	
	013204	117	124	040	

	013207	123	105	114	
	013212	105	103	124	
	013215	105	104	000	
28					
29					
30					
31	013220	122	105	107	EM2: .ASCIZ /RFG NOT INITIALIZED BY MST CLR/
	013223	040	116	117	
	013226	124	040	111	
	013231	116	111	124	
	013234	111	101	114	
	013237	111	132	105	
	013242	104	040	102	
	013245	131	040	115	
	013250	123	124	040	
	013253	103	114	122	
	013256	000			
32	013257	122	105	107	EM3: .ASCIZ /REG MISCOMPARE/
	013262	040	115	111	
	013265	123	103	117	
	013270	115	120	101	
	013273	122	105	000	
33	013276	117	122	104	EM7: .ASCIZ /ORDY NOT SET/
	013301	131	040	116	
	013304	117	124	040	
	013307	123	105	124	
	013312	000			
34	013313	117	122	104	EM8: .ASCIZ /ORDY NOT CLEARED/
	013316	131	040	116	
	013321	117	124	040	
	013324	103	114	105	
	013327	101	122	105	
	013332	104	000		
35	013334	117	103	117	EM9: .ASCIZ /OCOR NOT SET/
	013337	122	040	116	
	013342	117	124	040	
	013345	123	105	124	
	013350	000			
36	013351	117	103	117	EM10: .ASCIZ /OCOR NOT CLEARED/
	013354	122	040	116	
	013357	117	124	040	
	013362	103	114	105	
	013365	101	122	105	
	013370	104	000		
37	013372	117	101	103	EM11: .ASCIZ /OACT NOT SET/
	013375	124	040	116	
	013400	117	124	040	
	013403	123	105	124	
	013406	000			
38	013407	117	101	103	EM12: .ASCIZ /OACT NOT CLEARED/
	013412	124	040	116	
	013415	117	124	040	
	013420	103	114	105	
	013423	101	122	105	
	013426	104	000		
39	013430	111	122	104	EM17: .ASCIZ /IRDY NOT SET/
	013433	131	040	116	

	013436	117	124	040	
	013441	123	105	124	
	013444	000			
40	013445	111	122	104	EM18: .ASCIZ /IRDY NOT CLEARED/
	013450	131	040	116	
	013453	117	124	040	
	013456	103	114	105	
	013461	101	122	105	
	013464	104	000		
41	013466	111	103	111	EM19: .ASCIZ /ICIR NOT SET/
	013471	122	040	116	
	013474	117	124	040	
	013477	123	105	124	
	013502	000			
42	013503	111	103	111	EM20: .ASCIZ /ICIR NOT CLEARED/
	013506	122	040	116	
	013511	117	124	040	
	013514	103	114	105	
	013517	101	122	105	
	013522	104	000		
43	013524	111	101	103	EM21: .ASCIZ /IACT NOT SET/
	013527	124	040	116	
	013532	117	124	040	
	013535	123	105	124	
	013540	000			
44	013541	111	101	103	EM22: .ASCIZ /IACT NOT CLEARED/
	013544	124	040	116	
	013547	117	124	040	
	013552	103	114	105	
	013555	101	122	105	
	013560	104	000		
45	013562	122	123	117	EM28: .ASCIZ /RSOM NOT CLEARED/
	013565	115	040	116	
	013570	117	124	040	
	013573	103	114	105	
	013576	101	122	105	
	013601	104	000		
46	013603	122	123	117	EM29: .ASCIZ /RSOM NOT SET/
	013606	115	040	116	
	013611	117	124	040	
	013614	123	105	124	
	013617	000			
47	013620	122	105	117	EM30: .ASCIZ /REOM NOT CLEARED/
	013623	115	040	116	
	013626	117	124	040	
	013631	103	114	105	
	013634	101	122	105	
	013637	104	000		
48	013641	122	105	117	EM31: .ASCIZ /REOM NOT SET/
	013644	115	040	116	
	013647	117	124	040	
	013652	123	105	124	
	013655	000			
49	013656	122	103	126	EM34: .ASCIZ /RCV'D DATA MISCOMPARE/
	013661	047	104	040	
	013664	104	101	124	
	013667	101	040	115	

	013672	111	123	103	
	013675	117	115	120	
	013700	101	122	105	
	013703	000			
50	013704	102	103	103	EM35: .ASCIZ /BCC NOT CLEARED/
	013707	040	116	117	
	013712	124	040	103	
	013715	114	105	101	
	013720	122	105	104	
	013723	000			
51	013724	102	103	103	EM36: .ASCIZ /BCC NOT SET/
	013727	040	116	117	
	013732	124	040	123	
	013735	105	124	000	
52	013740	105	102	114	EM37: .ASCIZ /EBLK NOT CLEARED/
	013743	113	040	116	
	013746	117	124	040	
	013751	103	114	105	
	013754	101	122	105	
	013757	104	000		
53	013761	105	102	114	EM38: .ASCIZ /EBLK NOT SET/
	013764	113	040	116	
	013767	117	124	040	
	013772	123	105	124	
	013775	000			
54	013776	122	101	102	EM39: .ASCIZ /RAB NOT CLEARED/
	014001	040	116	117	
	014004	124	040	103	
	014007	114	105	101	
	014012	122	105	104	
	014015	000			
55	014016	122	101	102	EM40: .ASCIZ /RAB NOT SET/
	014021	040	116	117	
	014024	124	040	123	
	014027	105	124	000	
56	014032	117	126	122	EM41: .ASCIZ /OVRR NOT CLEARED/
	014035	122	040	116	
	014040	117	124	040	
	014043	103	114	105	
	014046	101	122	105	
	014051	104	000		
57	014053	117	126	122	EM42: .ASCIZ /OVRR NOT SET/
	014056	122	040	116	
	014061	117	124	040	
	014064	123	105	124	
	014067	000			
58	014070	123	127	040	EM43: .ASCIZ /SW PACK #1 INCORRECT/
	014073	120	101	103	
	014076	113	040	043	
	014101	061	040	111	
	014104	116	103	117	
	014107	122	122	105	
	014112	103	124	000	
59	014115	123	127	040	EM44: .ASCIZ /SW PACK #2 INCORRECT/
	014120	120	101	103	
	014123	113	040	043	
	014126	062	040	111	

	014131	116	103	117	
	014134	122	122	105	
	014137	103	124	000	
60	014142	123	127	040	EM45: .ASCIZ /SW PACK #3 INCORRECT/
	014145	120	101	103	
	014150	113	040	043	
	014153	063	040	111	
	014156	116	103	117	
	014161	122	122	105	
61	014164	103	124	000	
	014167	122	103	126	EM46: .ASCIZ /RCV SILO NOT CLEARED BY IC/
	014172	040	123	111	
	014175	114	117	040	
	014200	116	117	124	
	014203	040	103	114	
	014206	105	101	122	
	014211	105	104	040	
	014214	102	131	040	
62	014217	111	103	000	
	014222	101	123	123	EM47: .ASCIZ /ASSEMB BIT COUNT INCORRECT/
	014225	105	115	102	
	014230	040	102	111	
	014233	124	040	103	
	014236	117	125	116	
	014241	124	040	111	
	014244	116	103	117	
	014247	122	122	105	
63	014252	103	124	000	
	014255	117	104	104	EM48: .ASCIZ /ODD VRC PARITY BIT NOT SET/
	014260	040	126	122	
	014263	103	040	120	
	014266	101	122	111	
	014271	124	131	040	
	014274	102	111	124	
	014277	040	116	117	
	014302	124	040	123	
64	014305	105	124	000	
	014310	117	104	104	EM49: .ASCIZ /ODD VRC PARITY BIT NOT CLEARED/
	014313	040	126	122	
	014316	103	040	120	
	014321	101	122	111	
	014324	124	131	040	
	014327	102	111	124	
	014332	040	116	117	
	014335	124	040	103	
	014340	114	105	101	
	014343	122	105	104	
	014346	000			
65	014347	105	126	105	EM50: .ASCIZ /EVEN VRC PARITY BIT NOT SET/
	014352	116	040	126	
	014355	122	103	040	
	014360	120	101	122	
	014363	111	124	131	
	014366	040	102	111	
	014371	124	040	116	
	014374	117	124	040	
	014377	123	105	124	

	014402	000			
66	014403	105	126	105	EM51: .ASCIZ /EVEN VRC PARITY BIT NOT CLEARED/
	014406	116	040	126	
	014411	122	103	040	
	014414	120	101	122	
	014417	111	124	131	
	014422	040	102	111	
	014425	124	040	116	
	014430	117	124	040	
	014433	103	114	105	
	014436	101	122	105	
	014441	104	000		
67	014443	124	105	123	EM52: .ASCIZ /TEST MODE NOT SET BY MAINT1/
	014446	124	040	115	
	014451	117	104	105	
	014454	040	116	117	
	014457	124	040	123	
	014462	105	124	040	
	014465	102	131	040	
	014470	115	101	111	
	014473	116	124	061	
	014476	000			
68	014477	124	105	123	EM53: .ASCIZ /TEST MODE NOT SET BY MAINT2/
	014502	124	040	115	
	014505	117	104	105	
	014510	040	116	117	
	014513	124	040	123	
	014516	105	124	040	
	014521	102	131	040	
	014524	115	101	111	
	014527	116	124	062	
	014532	000			
69	014533	124	130	040	EM54: .ASCIZ /TX UNDERRUN ERROR/
	014536	125	116	104	
	014541	105	122	122	
	014544	125	116	040	
	014547	105	122	122	
	014552	117	122	000	
70	014555	104	124	122	EM55: .ASCIZ /DTR NOT SET/
	014560	040	116	117	
	014563	124	040	123	
	014566	105	124	000	
71	014571	122	111	116	EM56: .ASCIZ /RING NOT SET/
	014574	107	040	116	
	014577	117	124	040	
	014602	123	105	124	
	014605	000			
72	014606	115	117	104	EM57: .ASCIZ /MODR NOT SET/
	014611	122	040	116	
	014614	117	124	040	
	014617	123	105	124	
	014622	000			
73	014623	110	104	130	EM58: .ASCIZ /HDX NOT SET/
	014626	040	116	117	
	014631	124	040	123	
	014634	105	124	000	
74	014637	123	124	102	EM59: .ASCIZ /STBY NOT SET/

	014642	131	040	116		
	014645	117	124	040		
	014650	123	105	124		
	014653	000				
75	014654	122	124	123	EM60:	.ASCIZ /RTS NOT SET/
	014657	040	116	117		
	014662	124	040	123		
	014665	105	124	000		
76	014670	103	123	040	EM61:	.ASCIZ /CS NOT SET/
	014673	116	117	124		
	014676	040	123	105		
	014701	124	000			
77	014703	103	101	122	EM62:	.ASCIZ /CARR NOT SET/
	014706	122	040	116		
	014711	117	124	040		
	014714	123	105	124		
	014717	000				
78	014720	123	111	107	EM63:	.ASCIZ /SIGQ NOT SET/
	014723	121	040	116		
	014726	117	124	040		
	014731	123	105	124		
	014734	000				
79	014735	123	111	107	EM64:	.ASCIZ /SIGR NOT SET/
	014740	122	040	116		
	014743	117	124	040		
	014746	123	105	124		
	014751	000				
80	014752	122	124	123	EM65:	.ASCIZ /RTS NOT CLEARED/
	014755	040	116	117		
	014760	124	040	103		
	014763	114	105	101		
	014766	122	105	104		
	014771	000				
81	014772	103	101	122	EM66:	.ASCIZ /CARR NOT CLEARED/
	014775	122	040	116		
	015000	117	124	040		
	015003	103	114	105		
	015006	101	122	105		
	015011	104	000			
82						
83						
84						
85	015013	111	116	102	DH1:	.ASCIZ &INBUS/OUTBUS REG &
	015016	125	123	057		
	015021	117	125	124		
	015024	102	125	123		
	015027	040	122	105		
	015032	107	040	000		
86	015035	114	111	116	DH2:	.ASCIZ /LINE UNIT INBUS REGS :/
	015040	105	040	125		
	015043	116	111	124		
	015046	040	111	116		
	015051	102	125	123		
	015054	040	122	105		
	015057	107	123	040		
	015062	072	000			
87	015064	122	105	107	DH3:	.ASCIZ /REG10 REG11 REG12 REG13/

	015067	061	060	040	
	015072	040	040	122	
	015075	105	107	061	
	015100	061	040	040	
	015103	040	122	105	
	015106	107	061	062	
	015111	040	040	040	
	015114	122	105	107	
	015117	061	063	000	
88	015122	040	040	040	DH4: .ASCIZ / REG14 REG15 REG16 REG17/
	015125	040	122	105	
	015130	107	061	064	
	015133	040	040	040	
	015136	122	105	107	
	015141	061	065	040	
	015144	040	040	122	
	015147	105	107	061	
	015152	066	040	040	
	015155	040	122	105	
	015160	107	061	067	
	015163	000			
89	015164	061	065	000	DH5: .ASCIZ /15/
90	015167	061	066	000	DH6: .ASCIZ /16/
91	015172	114	111	116	DH7: .ASCIZ /LINE UNIT EXTENDED REGS :/
	015175	105	040	125	
	015200	116	111	124	
	015203	040	105	130	
	015206	124	105	116	
	015211	104	105	104	
	015214	040	122	105	
	015217	107	123	040	
	015222	072	000		
92	015224	101	130	060	DH8: .ASCIZ /AX0-15 AX0-16 AX1-15 AX1-16/
	015227	055	061	065	
	015232	040	040	101	
	015235	130	060	055	
	015240	061	066	040	
	015243	040	101	130	
	015246	061	055	061	
	015251	065	040	040	
	015254	101	130	061	
	015257	055	061	066	
	015262	000			
93	015263	040	040	040	DH9: .ASCIZ / AX2-15 AX2-16 AX3-15 AX3-16/
	015266	040	101	130	
	015271	062	055	061	
	015274	065	040	040	
	015277	101	130	062	
	015302	055	061	066	
	015305	040	040	101	
	015310	130	063	055	
	015313	061	065	040	
	015316	040	101	130	
	015321	063	055	061	
	015324	066	000		

94
95

.EVEN

```

96
97
98
99
100
101 015326          BGNMSG  ERR1
    015326
102 015326          PRINTB  #FMT1,#ADDRESS,MPCSR
    015326 013746 002436
    015332 012746 036460
    015336 012746 012256
    015342 012746 000003
    015346 010600
    015350 104414
    015352 062706 000010
103 015356          ENDMSG
    015356
    015356 104423
104
105
106
107 015360          BGNMSG  ERR2
    015360
108 015360          PRINTB  #FMT1,#ADDRESS,MPCSR
    015360 013746 002436
    015364 012746 036460
    015370 012746 012256
    015374 012746 000003
    015400 010600
    015402 104414
    015404 062706 000010
109 015410          PRINTB  #FMT2
    015410 012746 012266
    015414 012746 000001
    015420 010600
    015422 104414
    015424 062706 000004
110 015430          PRINTB  #FMT7,#DH1,REGNUM
    015430 013746 002364
    015434 012746 015013
    015440 012746 012450
    015444 012746 000003
    015450 010600
    015452 104414
    015454 062706 000010
111 015460          PRINTB  #FMT3,GOODAT,BADDAT
    015460 013746 002372
    015464 013746 002370
    015470 012746 012310
    015474 012746 000003
    015500 010600
    015502 104414
    015504 062706 000010
112 015510          PRINTX  #FMT4,#DH2,#DH3
    015510 012746 015064
    015514 012746 015035
    015520 012746 012352
ERR1::
    MOV    MPCSR,-(SP)
    MOV    #ADDRESS,-(SP)
    MOV    #FMT1,-(SP)
    MOV    #3,-(SP)
    MOV    SP,R0
    TRAP   C$PNTB
    ADD    #10,SP
L10002:
    TRAP   C$MSG
ERR2::
    MOV    MPCSR,-(SP)
    MOV    #ADDRESS,-(SP)
    MOV    #FMT1,-(SP)
    MOV    #3,-(SP)
    MOV    SP,R0
    TRAP   C$PNTB
    ADD    #10,SP
    MOV    #FMT2,-(SP)
    MOV    #1,-(SP)
    MOV    SP,R0
    TRAP   C$PNTB
    ADD    #4,SP
    MOV    REGNUM,-(SP)
    MOV    #DH1,-(SP)
    MOV    #FMT7,-(SP)
    MOV    #3,-(SP)
    MOV    SP,R0
    TRAP   C$PNTB
    ADD    #10,SP
    MOV    BADDAT,-(SP)
    MOV    GOODAT,-(SP)
    MOV    #FMT3,-(SP)
    MOV    #3,-(SP)
    MOV    SP,R0
    TRAP   C$PNTB
    ADD    #10,SP
    MOV    #DH3,-(SP)
    MOV    #DH2,-(SP)
    MOV    #FMT4,-(SP)

```


	015524	012746	000003				MOV	#3,-(SP)
	015530	010600					MOV	SP,R0
	015532	104415					TRAP	C\$PNTX
	015534	062706	000010				ADD	#10,SP
113	015540			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13			
	015540	013746	002274				MOV	LUR13,-(SP)
	015544	013746	002272				MOV	LUR12,-(SP)
	015550	013746	002270				MOV	LUR11,-(SP)
	015554	013746	002266				MOV	LUR10,-(SP)
	015560	012746	012365				MOV	#FMT5,-(SP)
	015564	012746	000005				MOV	#5,-(SP)
	015570	010600					MOV	SP,R0
	015572	104415					TRAP	C\$PNTX
	015574	062706	000014				ADD	#14,SP
114	015600			PRINTX	#FMT9,#DH4			
	015600	012746	015122				MOV	#DH4,-(SP)
	015604	012746	012514				MOV	#FMT9,-(SP)
	015610	012746	000002				MOV	#2,-(SP)
	015614	010600					MOV	SP,R0
	015616	104415					TRAP	C\$PNTX
	015620	062706	000006				ADD	#6,SP
115	015624			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17			
	015624	013746	002304				MOV	LUR17,-(SP)
	015630	013746	002302				MOV	LUR16,-(SP)
	015634	013746	002300				MOV	LUR15,-(SP)
	015640	013746	002276				MOV	LUR14,-(SP)
	015644	012746	012415				MOV	#FMT6,-(SP)
	015650	012746	000005				MOV	#5,-(SP)
	015654	010600					MOV	SP,R0
	015656	104415					TRAP	C\$PNTX
	015660	062706	000014				ADD	#14,SP
116	015664			ENDMSG				
	015664					L10003:	TRAP	C\$MSG
	015664	104423						
117								
118								
119								
120								
121								
122	015666			BGNMSG	ERR3			
	015666					ERR3::		
123	015666			PRINTB	#FMT1,#ADDRES,MPCSR			
	015666	013746	002436				MOV	MPCSR,-(SP)
	015672	012746	036460				MOV	#ADDRES,-(SP)
	015676	012746	012256				MOV	#FMT1,-(SP)
	015702	012746	000003				MOV	#3,-(SP)
	015706	010600					MOV	SP,R0
	015710	104414					TRAP	C\$PNTB
	015712	062706	000010				ADD	#10,SP
124	015716			PRINTB	#FMT2			
	015716	012746	012266				MOV	#FMT2,-(SP)
	015722	012746	000001				MOV	#1,-(SP)
	015726	010600					MOV	SP,R0
	015730	104414					TRAP	C\$PNTB
	015732	062706	000004				ADD	#4,SP
125	015736			PRINTB	#FMT8,TMP1,TMP0			
	015736	013746	002516				MOV	TMP0,-(SP)

	015742	013746	002520		MOV	TMP1,-(SP)
	015746	012746	012460		MOV	#FMT8,-(SP)
	015752	012746	000003		MOV	#3,-(SP)
	015756	010600			MOV	SP,R0
	015760	104414			TRAP	C\$PNTB
	015762	062706	000010		ADD	#10,SP
126	015766			PRINTB	#FMT3,GOODAT,BADDAT	
	015766	013746	002372		MOV	BADDAT,-(SP)
	015772	013746	002370		MOV	GOODAT,-(SP)
	015776	012746	012310		MOV	#FMT3,-(SP)
	016002	012746	000003		MOV	#3,-(SP)
	016006	010600			MOV	SP,R0
	016010	104414			TRAP	C\$PNTB
	016012	062706	000010		ADD	#10,SP
127	016016			PRINTX	#FMT4,#DH2,#DH3	
	016016	012746	015064		MOV	#DH3,-(SP)
	016022	012746	015035		MOV	#DH2,-(SP)
	016026	012746	012352		MOV	#FMT4,-(SP)
	016032	012746	000003		MOV	#3,-(SP)
	016036	010600			MOV	SP,R0
	016040	104415			TRAP	C\$PNTX
	016042	062706	000010		ADD	#10,SP
128	016046			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	016046	013746	002274		MOV	LUR13,-(SP)
	016052	013746	002272		MOV	LUR12,-(SP)
	016056	013746	002270		MOV	LUR11,-(SP)
	016062	013746	002266		MOV	LUR10,-(SP)
	016066	012746	012365		MOV	#FMT5,-(SP)
	016072	012746	000005		MOV	#5,-(SP)
	016076	010600			MOV	SP,R0
	016100	104415			TRAP	C\$PNTX
	016102	062706	000014		ADD	#14,SP
129	016106			PRINTX	#FMT9,#DH4	
	016106	012746	015122		MOV	#DH4,-(SP)
	016112	012746	012514		MOV	#FMT9,-(SP)
	016116	012746	000002		MOV	#2,-(SP)
	016122	010600			MOV	SP,R0
	016124	104415			TRAP	C\$PNTX
	016126	062706	000006		ADD	#6,SP
130	016132			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	016132	013746	002304		MOV	LUR17,-(SP)
	016136	013746	002302		MOV	LUR16,-(SP)
	016142	013746	002300		MOV	LUR15,-(SP)
	016146	013746	002276		MOV	LUR14,-(SP)
	016152	012746	012415		MOV	#FMT6,-(SP)
	016156	012746	000005		MOV	#5,-(SP)
	016162	010600			MOV	SP,R0
	016164	104415			TRAP	C\$PNTX
	016166	062706	000014		ADD	#14,SP
131	016172			PRINTX	#FMT4,#DH7,#DH8	
	016172	012746	015224		MOV	#DH8,-(SP)
	016176	012746	015172		MOV	#DH7,-(SP)
	016202	012746	012352		MOV	#FMT4,-(SP)
	016206	012746	000003		MOV	#3,-(SP)
	016212	010600			MOV	SP,R0
	016214	104415			TRAP	C\$PNTX
	016216	062706	000010		ADD	#10,SP

132	016222			PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16			
	016222	013746	002314			MOV	AX1.16,-(SP)
	016226	013746	002312			MOV	AX1.15,-(SP)
	016232	013746	002310			MOV	AX0.16,-(SP)
	016236	013746	002306			MOV	AX0.15,-(SP)
	016242	012746	012365			MOV	#FMT5,-(SP)
	016246	012746	000005			MOV	#5,-(SP)
	016252	010600				MOV	SP,R0
	016254	104415				TRAP	C\$PNTX
	016256	062706	000014			ADD	#14,SP
133	016262			PRINTX #FMT9,#DH9			
	016262	012746	015263			MOV	#DH9,-(SP)
	016266	012746	012514			MOV	#FMT9,-(SP)
	016272	012746	000002			MOV	#2,-(SP)
	016276	010600				MOV	SP,R0
	016300	104415				TRAP	C\$PNTX
	016302	062706	000006			ADD	#6,SP
134	016306			PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16			
	016306	013746	002324			MOV	AX3.16,-(SP)
	016312	013746	002322			MOV	AX3.15,-(SP)
	016316	013746	002320			MOV	AX2.16,-(SP)
	016322	013746	002316			MOV	AX2.15,-(SP)
	016326	012746	012415			MOV	#FMT6,-(SP)
	016332	012746	000005			MOV	#5,-(SP)
	016336	010600				MOV	SP,R0
	016340	104415				TRAP	C\$PNTX
	016342	062706	000014			ADD	#14,SP
135	016346			ENDMSG			
	016346						
	016346	104423				L10004:	TRAP C\$MSG
136							
137							
138							
139							
140							
141	016350			BGNMSG ERR4			
	016350					ERR4::	
142	016350			PRINTB #FMT10,SUBRPC			
	016350	013746	002336			MOV	SUBRPC,-(SP)
	016354	012746	012521			MOV	#FMT10,-(SP)
	016360	012746	000002			MOV	#2,-(SP)
	016364	010600				MOV	SP,R0
	016366	104414				TRAP	C\$PNTB
	016370	062706	000006			ADD	#6,SP
143	016374			PRINTB #FMT1,#ADDRES,MPCSR			
	016374	013746	002436			MOV	MPCSR,-(SP)
	016400	012746	036460			MOV	#ADDRES,-(SP)
	016404	012746	012256			MOV	#FMT1,-(SP)
	016410	012746	000003			MOV	#3,-(SP)
	016414	010600				MOV	SP,R0
	016416	104414				TRAP	C\$PNTB
	016420	062706	000010			ADD	#10,SP
144	016424			PRINTB #FMT2			
	016424	012746	012266			MOV	#FMT2,-(SP)
	016430	012746	000001			MOV	#1,-(SP)
	016434	010600				MOV	SP,R0
	016436	104414				TRAP	C\$PNTB

145	016440	062706	000004		ADD	#4,SP	
	016444			PRINTB	#FMT7,#DH1,REGNUM		
	016450	012746	015013		MOV	REGNUM,-(SP)	
	016454	012746	012450		MOV	#DH1,-(SP)	
	016460	012746	000003		MOV	#FMT7,-(SP)	
	016464	010600			MOV	#3,-(SP)	
	016466	104414			MOV	SP,R0	
	016470	062706	000010		TRAP	C\$PNTB	
146	016474			PRINTX	#FMT4,#DH2,#DH3	ADD	#10,SP
	016474	012746	015064		MOV	#DH3,-(SP)	
	016500	012746	015035		MOV	#DH2,-(SP)	
	016504	012746	012352		MOV	#FMT4,-(SP)	
	016510	012746	000003		MOV	#3,-(SP)	
	016514	010600			MOV	SP,R0	
	016516	104415			TRAP	C\$PNTX	
	016520	062706	000010		ADD	#10,SP	
147	016524			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13		
	016524	013746	002274		MOV	LUR13,-(SP)	
	016530	013746	002272		MOV	LUR12,-(SP)	
	016534	013746	002270		MOV	LUR11,-(SP)	
	016540	013746	002266		MOV	LUR10,-(SP)	
	016544	012746	012365		MOV	#FMT5,-(SP)	
	016550	012746	000005		MOV	#5,-(SP)	
	016554	010600			MOV	SP,R0	
	016556	104415			TRAP	C\$PNTX	
	016560	062706	000014		ADD	#14,SP	
148	016564			PRINTX	#FMT9,#DH4		
	016564	012746	015122		MOV	#DH4,-(SP)	
	016570	012746	012514		MOV	#FMT9,-(SP)	
	016574	012746	000002		MOV	#2,-(SP)	
	016600	010600			MOV	SP,R0	
	016602	104415			TRAP	C\$PNTX	
	016604	062706	000006		ADD	#6,SP	
149	016610			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17		
	016610	013746	002304		MOV	LUR17,-(SP)	
	016614	013746	002302		MOV	LUR16,-(SP)	
	016620	013746	002300		MOV	LUR15,-(SP)	
	016624	013746	002276		MOV	LUR14,-(SP)	
	016630	012746	012415		MOV	#FMT6,-(SP)	
	016634	012746	000005		MOV	#5,-(SP)	
	016640	010600			MOV	SP,R0	
	016642	104415			TRAP	C\$PNTX	
	016644	062706	000014		ADD	#14,SP	
150	016650			PRINTX	#FMT4,#DH7,#DH8		
	016650	012746	015224		MOV	#DH8,-(SP)	
	016654	012746	015172		MOV	#DH7,-(SP)	
	016660	012746	012352		MOV	#FMT4,-(SP)	
	016664	012746	000003		MOV	#3,-(SP)	
	016670	010600			MOV	SP,R0	
	016672	104415			TRAP	C\$PNTX	
	016674	062706	000010		ADD	#10,SP	
151	016700			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16		
	016700	013746	002314		MOV	AX1.16,-(SP)	
	016704	013746	002312		MOV	AX1.15,-(SP)	
	016710	013746	002310		MOV	AX0.16,-(SP)	
	016714	013746	002306		MOV	AX0.15,-(SP)	

	016720	012746	012365				MOV	#FMT5,-(SP)
	016724	012746	000005				MOV	#5,-(SP)
	016730	010600					MOV	SP,R0
	016732	104415					TRAP	C\$PNTX
	016734	062706	000014				ADD	#14,SP
152	016740			PRINTX	#FMT9,#DH9			
	016740	012746	015263				MOV	#DH9,-(SP)
	016744	012746	012514				MOV	#FMT9,-(SP)
	016750	012746	000002				MOV	#2,-(SP)
	016754	010600					MOV	SP,R0
	016756	104415					TRAP	C\$PNTX
	016760	062706	000006				ADD	#6,SP
153	016764			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16			
	016764	013746	002324				MOV	AX3.16,-(SP)
	016770	013746	002322				MOV	AX3.15,-(SP)
	016774	013746	002320				MOV	AX2.16,-(SP)
	017000	013746	002316				MOV	AX2.15,-(SP)
	017004	012746	012415				MOV	#FMT6,-(SP)
	017010	012746	000005				MOV	#5,-(SP)
	017014	010600					MOV	SP,R0
	017016	104415					TRAP	C\$PNTX
	017020	062706	000014				ADD	#14,SP
154	017024			ENDMSG				
	017024							
	017024	104423					L10005:	TRAP
								C\$MSG
155								
156								
157								
158								
159								
160	017026			BGNMSG	ERR5			
	017026							
161	017026			PRINTB	#FMT1,#ADDRES,MPCSR		ERR5::	
	017026	013746	002436				MOV	MPCSR,-(SP)
	017032	012746	036460				MOV	#ADDRES,-(SP)
	017036	012746	012256				MOV	#FMT1,-(SP)
	017042	012746	000003				MOV	#3,-(SP)
	017046	010600					MOV	SP,R0
	017050	104414					TRAP	C\$PNTB
	017052	062706	000010				ADD	#10,SP
162	017056			PRINTB	#FMT11,REGNUM,LOADAT			
	017056	013746	002374				MOV	LOADAT,-(SP)
	017062	013746	002364				MOV	REGNUM,-(SP)
	017066	012746	012552				MOV	#FMT11,-(SP)
	017072	012746	000003				MOV	#3,-(SP)
	017076	010600					MOV	SP,R0
	017100	104414					TRAP	C\$PNTB
	017102	062706	000010				ADD	#10,SP
163	017106			PRINTB	#FMT2			
	017106	012746	012266				MOV	#FMT2,-(SP)
	017112	012746	000001				MOV	#1,-(SP)
	017116	010600					MOV	SP,R0
	017120	104414					TRAP	C\$PNTB
	017122	062706	000004				ADD	#4,SP
164	017126			PRINTB	#FMT8,TMP1,TMP0			
	017126	013746	002516				MOV	TMP0,-(SP)
	017132	013746	002520				MOV	TMP1,-(SP)

	017136	012746	012460		MOV	#FMT8,-(SP)
	017142	012746	000003		MOV	#3,-(SP)
	017146	010600			MOV	SP,R0
	017150	104414			TRAP	C\$PNTB
	017152	062706	000010		ADD	#10,SP
165	017156			PRINTB	#FMT3,GOODAT,BADDAT	
	017156	013746	002372		MOV	BADDAT,-(SP)
	017162	013746	002370		MOV	GOODAT,-(SP)
	017166	012746	012310		MOV	#FMT3,-(SP)
	017172	012746	000003		MOV	#3,-(SP)
	017176	010600			MOV	SP,R0
	017200	104414			TRAP	C\$PNTB
	017202	062706	000010		ADD	#10,SP
166	017206			PRINTX	#FMT4,#DH2,#DH3	
	017206	012746	015064		MOV	#DH3,-(SP)
	017212	012746	015035		MOV	#DH2,-(SP)
	017216	012746	012352		MOV	#FMT4,-(SP)
	017222	012746	000003		MOV	#3,-(SP)
	017226	010600			MOV	SP,R0
	017230	104415			TRAP	C\$PNTX
	017232	062706	000010		ADD	#10,SP
167	017236			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	017236	013746	002274		MOV	LUR13,-(SP)
	017242	013746	002272		MOV	LUR12,-(SP)
	017246	013746	002270		MOV	LUR11,-(SP)
	017252	013746	002266		MOV	LUR10,-(SP)
	017256	012746	012365		MOV	#FMT5,-(SP)
	017262	012746	000005		MOV	#5,-(SP)
	017266	010600			MOV	SP,R0
	017270	104415			TRAP	C\$PNTX
	017272	062706	000014		ADD	#14,SP
168	017276			PRINTX	#FMT9,#DH4	
	017276	012746	015122		MOV	#DH4,-(SP)
	017302	012746	012514		MOV	#FMT9,-(SP)
	017306	012746	000002		MOV	#2,-(SP)
	017312	010600			MOV	SP,R0
	017314	104415			TRAP	C\$PNTX
	017316	062706	000006		ADD	#6,SP
169	017322			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	017322	013746	002304		MOV	LUR17,-(SP)
	017326	013746	002302		MOV	LUR16,-(SP)
	017332	013746	002300		MOV	LUR15,-(SP)
	017336	013746	002276		MOV	LUR14,-(SP)
	017342	012746	012415		MOV	#FMT6,-(SP)
	017346	012746	000005		MOV	#5,-(SP)
	017352	010600			MOV	SP,R0
	017354	104415			TRAP	C\$PNTX
	017356	062706	000014		ADD	#14,SP
170	017362			PRINTX	#FMT4,#DH7,#DH8	
	017362	012746	015224		MOV	#DH8,-(SP)
	017366	012746	015172		MOV	#DH7,-(SP)
	017372	012746	012352		MOV	#FMT4,-(SP)
	017376	012746	000003		MOV	#3,-(SP)
	017402	010600			MOV	SP,R0
	017404	104415			TRAP	C\$PNTX
	017406	062706	000010		ADD	#10,SP
171	017412			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16	

	017412	013746	002314		MOV	AX1.16,-(SP)
	017416	013746	002312		MOV	AX1.15,-(SP)
	017422	013746	002310		MOV	AX0.16,-(SP)
	017426	013746	002306		MOV	AX0.15,-(SP)
	017432	012746	012365		MOV	#FMT5,-(SP)
	017436	012746	000005		MOV	#5,-(SP)
	017442	010600			MOV	SP,R0
	017444	104415			TRAP	C\$PNTX
	017446	062706	000014		ADD	#14,SP
172	017452			PRINTX	#FMT9,#DH9	
	017452	012746	015263		MOV	#DH9,-(SP)
	017456	012746	012514		MOV	#FMT9,-(SP)
	017462	012746	000002		MOV	#2,-(SP)
	017466	010600			MOV	SP,R0
	017470	104415			TRAP	C\$PNTX
	017472	062706	000006		ADD	#6,SP
173	017476			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16	
	017476	013746	002324		MOV	AX3.16,-(SP)
	017502	013746	002322		MOV	AX3.15,-(SP)
	017506	013746	002320		MOV	AX2.16,-(SP)
	017512	013746	002316		MOV	AX2.15,-(SP)
	017516	012746	012415		MOV	#FMT6,-(SP)
	017522	012746	000005		MOV	#5,-(SP)
	017526	010600			MOV	SP,R0
	017530	104415			TRAP	C\$PNTX
	017532	062706	000014		ADD	#14,SP
174	017536			ENDMSG		
	017536					
	017536	104423				
175						
176						
177						
178						
179						
180	017540			BGNMSG	ERR6	
	017540					
181	017540			PRINTB	#FMT10,SUBRPC	ERR6::
	017540	013746	002336		MOV	SUBRPC,-(SP)
	017544	012746	012521		MOV	#FMT10,-(SP)
	017550	012746	000002		MOV	#2,-(SP)
	017554	010600			MOV	SP,R0
	017556	104414			TRAP	C\$PNTB
	017560	062706	000006		ADD	#6,SP
182	017564			PRINTB	#FMT1,#ADDRES,MPCSR	
	017564	013746	002436		MOV	MPCSR,-(SP)
	017570	012746	036460		MOV	#ADDRES,-(SP)
	017574	012746	012256		MOV	#FMT1,-(SP)
	017600	012746	000003		MOV	#3,-(SP)
	017604	010600			MOV	SP,R0
	017606	104414			TRAP	C\$PNTB
	017610	062706	000010		ADD	#10,SP
183	017614			PRINTB	#FMT2	
	017614	012746	012266		MOV	#FMT2,-(SP)
	017620	012746	000001		MOV	#1,-(SP)
	017624	010600			MOV	SP,R0
	017626	104414			TRAP	C\$PNTB
	017630	062706	000004		ADD	#4,SP

L10006:

TRAP C\$MSG

ERR6::

184	017634		PRINTB #FMT8,TMP1,TMP0		
	017634	013746		MOV	TMP0,-(SP)
	017640	013746		MOV	TMP1,-(SP)
	017644	012746		MOV	#FMT8,-(SP)
	017650	012746		MOV	#3,-(SP)
	017654	010600		MOV	SP,R0
	017656	104414		TRAP	C\$PNTB
	017660	062706		ADD	#10,SP
185	017664		PRINTX #FMT4,#DH2,#DH3		
	017664	012746		MOV	#DH3,-(SP)
	017670	012746		MOV	#DH2,-(SP)
	017674	012746		MOV	#FMT4,-(SP)
	017700	012746		MOV	#3,-(SP)
	017704	010600		MOV	SP,R0
	017706	104415		TRAP	C\$PNTX
	017710	062706		ADD	#10,SP
186	017714		PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13		
	017714	013746		MOV	LUR13,-(SP)
	017720	013746		MOV	LUR12,-(SP)
	017724	013746		MOV	LUR11,-(SP)
	017730	013746		MOV	LUR10,-(SP)
	017734	012746		MOV	#FMT5,-(SP)
	017740	012746		MOV	#5,-(SP)
	017744	010600		MOV	SP,R0
	017746	104415		TRAP	C\$PNTX
	017750	062706		ADD	#14,SP
187	017754		PRINTX #FMT9,#DH4		
	017754	012746		MOV	#DH4,-(SP)
	017760	012746		MOV	#FMT9,-(SP)
	017764	012746		MOV	#2,-(SP)
	017770	010600		MOV	SP,R0
	017772	104415		TRAP	C\$PNTX
	017774	062706		ADD	#6,SP
188	020000		PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17		
	020000	013746		MOV	LUR17,-(SP)
	020004	013746		MOV	LUR16,-(SP)
	020010	013746		MOV	LUR15,-(SP)
	020014	013746		MOV	LUR14,-(SP)
	020020	012746		MOV	#FMT6,-(SP)
	020024	012746		MOV	#5,-(SP)
	020030	010600		MOV	SP,R0
	020032	104415		TRAP	C\$PNTX
	020034	062706		ADD	#14,SP
189	020040		PRINTX #FMT4,#DH7,#DH8		
	020040	012746		MOV	#DH8,-(SP)
	020044	012746		MOV	#DH7,-(SP)
	020050	012746		MOV	#FMT4,-(SP)
	020054	012746		MOV	#3,-(SP)
	020060	010600		MOV	SP,R0
	020062	104415		TRAP	C\$PNTX
	020064	062706		ADD	#10,SP
190	020070		PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16		
	020070	013746		MOV	AX1.16,-(SP)
	020074	013746		MOV	AX1.15,-(SP)
	020100	013746		MOV	AX0.16,-(SP)
	020104	013746		MOV	AX0.15,-(SP)
	020110	012746		MOV	#FMT5,-(SP)

	020114	012746	000005			MOV	#5,-(SP)
	020120	010600				MOV	SP,R0
	020122	104415				TRAP	C\$PNTX
	020124	062706	000014			ADD	#14,SP
191	020130			PRINTX	#FMT9,#DH9		
	020130	012746	015263			MOV	#DH9,-(SP)
	020134	012746	012514			MOV	#FMT9,-(SP)
	020140	012746	000002			MOV	#2,-(SP)
	020144	010600				MOV	SP,R0
	020146	104415				TRAP	C\$PNTX
	020150	062706	000006			ADD	#6,SP
192	020154			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16		
	020154	013746	002324			MOV	AX3.16,-(SP)
	020160	013746	002322			MOV	AX3.15,-(SP)
	020164	013746	002320			MOV	AX2.16,-(SP)
	020170	013746	002316			MOV	AX2.15,-(SP)
	020174	012746	012415			MOV	#FMT6,-(SP)
	020200	012746	000005			MOV	#5,-(SP)
	020204	010600				MOV	SP,R0
	020206	104415				TRAP	C\$PNTX
	020210	062706	000014			ADD	#14,SP
193	020214			ENDMSG			
	020214						
	020214	104423				L10007:	TRAP C\$MSG
194							
195							
196							
197							
198							
199	020216			BGNMSG	ERR7		
	020216						
200	020216			PRINTB	#FMT1,#ADDRES,MPCSR		ERR7::
	020216	013746	002436			MOV	MPCSR,-(SP)
	020222	012746	036460			MOV	#ADDRES,-(SP)
	020226	012746	012256			MOV	#FMT1,-(SP)
	020232	012746	000003			MOV	#3,-(SP)
	020236	010600				MOV	SP,R0
	020240	104414				TRAP	C\$PNTB
	020242	062706	000010			ADD	#10,SP
201	020246			PRINTB	#FMT2		
	020246	012746	012266			MOV	#FMT2,-(SP)
	020252	012746	000001			MOV	#1,-(SP)
	020256	010600				MOV	SP,R0
	020260	104414				TRAP	C\$PNTB
	020262	062706	000004			ADD	#4,SP
202	020266			PRINTB	#FMT7,#DH1,REGNUM		
	020266	013746	002364			MOV	REGNUM,-(SP)
	020272	012746	015013			MOV	#DH1,-(SP)
	020276	012746	012450			MOV	#FMT7,-(SP)
	020302	012746	000003			MOV	#3,-(SP)
	020306	010600				MOV	SP,R0
	020310	104414				TRAP	C\$PNTB
	020312	062706	000010			ADD	#10,SP
203	020316			PRINTX	#FMT4,#DH2,#DH3		
	020316	012746	015064			MOV	#DH3,-(SP)
	020322	012746	015035			MOV	#DH2,-(SP)
	020326	012746	012352			MOV	#FMT4,-(SP)

	020332	012746	000003		MOV	#3,-(SP)
	020336	010600			MOV	SP,R0
	020340	104415			TRAP	C\$PNTX
	020342	062706	000010		ADD	#10,SP
204	020346			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	020346	013746	002274		MOV	LUR13,-(SP)
	020352	013746	002272		MOV	LUR12,-(SP)
	020356	013746	002270		MOV	LUR11,-(SP)
	020362	013746	002266		MOV	LUR10,-(SP)
	020366	012746	012365		MOV	#FMT5,-(SP)
	020372	012746	000005		MOV	#5,-(SP)
	020376	010600			MOV	SP,R0
	020400	104415			TRAP	C\$PNTX
	020402	062706	000014		ADD	#14,SP
205	020406			PRINTX	#FMT9,#DH4	
	020406	012746	015122		MOV	#DH4,-(SP)
	020412	012746	012514		MOV	#FMT9,-(SP)
	020416	012746	000002		MOV	#2,-(SP)
	020422	010600			MOV	SP,R0
	020424	104415			TRAP	C\$PNTX
	020426	062706	000006		ADD	#6,SP
206	020432			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	020432	013746	002304		MOV	LUR17,-(SP)
	020436	013746	002302		MOV	LUR16,-(SP)
	020442	013746	002300		MOV	LUR15,-(SP)
	020446	013746	002276		MOV	LUR14,-(SP)
	020452	012746	012415		MOV	#FMT6,-(SP)
	020456	012746	000005		MOV	#5,-(SP)
	020462	010600			MOV	SP,R0
	020464	104415			TRAP	C\$PNTX
	020466	062706	000014		ADD	#14,SP
207	020472			PRINTX	#FMT4,#DH7,#DH8	
	020472	012746	015224		MOV	#DH8,-(SP)
	020476	012746	015172		MOV	#DH7,-(SP)
	020502	012746	012352		MOV	#FMT4,-(SP)
	020506	012746	000003		MOV	#3,-(SP)
	020512	010600			MOV	SP,R0
	020514	104415			TRAP	C\$PNTX
	020516	062706	000010		ADD	#10,SP
208	020522			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16	
	020522	013746	002314		MOV	AX1.16,-(SP)
	020526	013746	002312		MOV	AX1.15,-(SP)
	020532	013746	002310		MOV	AX0.16,-(SP)
	020536	013746	002306		MOV	AX0.15,-(SP)
	020542	012746	012365		MOV	#FMT5,-(SP)
	020546	012746	000005		MOV	#5,-(SP)
	020552	010600			MOV	SP,R0
	020554	104415			TRAP	C\$PNTX
	020556	062706	000014		ADD	#14,SP
209	020562			PRINTX	#FMT9,#DH9	
	020562	012746	015263		MOV	#DH9,-(SP)
	020566	012746	012514		MOV	#FMT9,-(SP)
	020572	012746	000002		MOV	#2,-(SP)
	020576	010600			MOV	SP,R0
	020600	104415			TRAP	C\$PNTX
	020602	062706	000006		ADD	#6,SP
210	020606			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16	

	020606	013746	002324		MOV	AX3.16,-(SP)
	020612	013746	002322		MOV	AX3.15,-(SP)
	020616	013746	002320		MOV	AX2.16,-(SP)
	020622	013746	002316		MOV	AX2.15,-(SP)
	020626	012746	012415		MOV	#FMT6,-(SP)
	020632	012746	000005		MOV	#5,-(SP)
	020636	010600			MOV	SP,R0
	020640	104415			TRAP	C\$PNTX
	020642	062706	000014		ADD	#14,SP
211	020646			ENDMSG		
	020646				L10010:	
	020646	104423			TRAP	C\$MSG
212						
213						
214						
215						
216						
217	020650			BGNMSG ERR8		
	020650				ERR8::	
218	020650			PRINTB #FMT10,SUBRPC		
	020650	013746	002336		MOV	SUBRPC,-(SP)
	020654	012746	012521		MOV	#FMT10,-(SP)
	020660	012746	000002		MOV	#2,-(SP)
	020664	010600			MOV	SP,R0
	020666	104414			TRAP	C\$PNTB
	020670	062706	000006		ADD	#6,SP
219	020674			PRINTB #FMT1,#ADDRES,MPCSR		
	020674	013746	002436		MOV	MPCSR,-(SP)
	020700	012746	036460		MOV	#ADDRES,-(SP)
	020704	012746	012256		MOV	#FMT1,-(SP)
	020710	012746	000003		MOV	#3,-(SP)
	020714	010600			MOV	SP,R0
	020716	104414			TRAP	C\$PNTB
	020720	062706	000010		ADD	#10,SP
220	020724			PRINTB #FMT2		
	020724	012746	012266		MOV	#FMT2,-(SP)
	020730	012746	000001		MOV	#1,-(SP)
	020734	010600			MOV	SP,R0
	020736	104414			TRAP	C\$PNTB
	020740	062706	000004		ADD	#4,SP
221	020744			PRINTB #FMT7,#DH1,REGNUM		
	020744	013746	002364		MOV	REGNUM,-(SP)
	020750	012746	015013		MOV	#DH1,-(SP)
	020754	012746	012450		MOV	#FMT7,-(SP)
	020760	012746	000003		MOV	#3,-(SP)
	020764	010600			MOV	SP,R0
	020766	104414			TRAP	C\$PNTB
	020770	062706	000010		ADD	#10,SP
222	020774			PRINTB #FMT3,GOODAT,BADDAT		
	020774	013746	002372		MOV	BADDAT,-(SP)
	021000	013746	002370		MOV	GOODAT,-(SP)
	021004	012746	012310		MOV	#FMT3,-(SP)
	021010	012746	000003		MOV	#3,-(SP)
	021014	010600			MOV	SP,R0
	021016	104414			TRAP	C\$PNTB
	021020	062706	000010		ADD	#10,SP
223	021024			PRINTX #FMT4,#DH2,#DH3		

	021024	012746	015064		MOV	#DH3,-(SP)
	021030	012746	015035		MOV	#DH2,-(SP)
	021034	012746	012352		MOV	#FMT4,-(SP)
	021040	012746	000003		MOV	#3,-(SP)
	021044	010600			MOV	SP,R0
	021046	104415			TRAP	C\$PNTX
	021050	062706	000010		ADD	#10,SP
224	021054			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	021054	013746	002274		MOV	LUR13,-(SP)
	021060	013746	002272		MOV	LUR12,-(SP)
	021064	013746	002270		MOV	LUR11,-(SP)
	021070	013746	002266		MOV	LUR10,-(SP)
	021074	012746	012365		MOV	#FMT5,-(SP)
	021100	012746	000005		MOV	#5,-(SP)
	021104	010600			MOV	SP,R0
	021106	104415			TRAP	C\$PNTX
	021110	062706	000014		ADD	#14,SP
225	021114			PRINTX	#FMT9,#DH4	
	021114	012746	015122		MOV	#DH4,-(SP)
	021120	012746	012514		MOV	#FMT9,-(SP)
	021124	012746	000002		MOV	#2,-(SP)
	021130	010600			MOV	SP,R0
	021132	104415			TRAP	C\$PNTX
	021134	062706	000006		ADD	#6,SP
226	021140			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	021140	013746	002304		MOV	LUR17,-(SP)
	021144	013746	002302		MOV	LUR16,-(SP)
	021150	013746	002300		MOV	LUR15,-(SP)
	021154	013746	002276		MOV	LUR14,-(SP)
	021160	012746	012415		MOV	#FMT6,-(SP)
	021164	012746	000005		MOV	#5,-(SP)
	021170	010600			MOV	SP,R0
	021172	104415			TRAP	C\$PNTX
	021174	062706	000014		ADD	#14,SP
227	021200			PRINTX	#FMT4,#DH7,#DH8	
	021200	012746	015224		MOV	#DH8,-(SP)
	021204	012746	015172		MOV	#DH7,-(SP)
	021210	012746	012352		MOV	#FMT4,-(SP)
	021214	012746	000003		MOV	#3,-(SP)
	021220	010600			MOV	SP,R0
	021222	104415			TRAP	C\$PNTX
	021224	062706	000010		ADD	#10,SP
228	021230			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16	
	021230	013746	002314		MOV	AX1.16,-(SP)
	021234	013746	002312		MOV	AX1.15,-(SP)
	021240	013746	002310		MOV	AX0.16,-(SP)
	021244	013746	002306		MOV	AX0.15,-(SP)
	021250	012746	012365		MOV	#FMT5,-(SP)
	021254	012746	000005		MOV	#5,-(SP)
	021260	010600			MOV	SP,R0
	021262	104415			TRAP	C\$PNTX
	021264	062706	000014		ADD	#14,SP
229	021270			PRINTX	#FMT9,#DH9	
	021270	012746	015263		MOV	#DH9,-(SP)
	021274	012746	012514		MOV	#FMT9,-(SP)
	021300	012746	000002		MOV	#2,-(SP)
	021304	010600			MOV	SP,R0

	021306	104415			TRAP	C\$PNTX
	021310	062706	000006		ADD	#6,SP
230	021314			PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16		
	021314	013746	002324		MOV	AX3.16,-(SP)
	021320	013746	002322		MOV	AX3.15,-(SP)
	021324	013746	002320		MOV	AX2.16,-(SP)
	021330	013746	002316		MOV	AX2.15,-(SP)
	021334	012746	012415		MOV	#FMT6,-(SP)
	021340	012746	000005		MOV	#5,-(SP)
	021344	010600			MOV	SP,R0
	021346	104415			TRAP	C\$PNTX
	021350	062706	000014		ADD	#14,SP
231	021354			ENDMSG		
	021354				L10011:	
	021354	104423			TRAP	C\$MSG
232						
233						
234						
235						
236						
237	021356			BGNMSG ERR10		
	021356				ERR10::	
238	021356			PRINTB #FMT1,#ADDRES,MPCSR		
	021356	013746	002436		MOV	MPCSR,-(SP)
	021362	012746	036460		MOV	#ADDRES,-(SP)
	021366	012746	012256		MOV	#FMT1,-(SP)
	021372	012746	000003		MOV	#3,-(SP)
	021376	010600			MOV	SP,R0
	021400	104414			TRAP	C\$PNTB
	021402	062706	000010		ADD	#10,SP
239	021406			PRINTB #FMT2		
	021406	012746	012266		MOV	#FMT2,-(SP)
	021412	012746	000001		MOV	#1,-(SP)
	021416	010600			MOV	SP,R0
	021420	104414			TRAP	C\$PNTB
	021422	062706	000004		ADD	#4,SP
240	021426			PRINTB #FMT8,TMP1,TMP0		
	021426	013746	002516		MOV	TMP0,-(SP)
	021432	013746	002520		MOV	TMP1,-(SP)
	021436	012746	012460		MOV	#FMT8,-(SP)
	021442	012746	000003		MOV	#3,-(SP)
	021446	010600			MOV	SP,R0
	021450	104414			TRAP	C\$PNTB
	021452	062706	000010		ADD	#10,SP
241	021456			PRINTX #FMT4,#DH2,#DH3		
	021456	012746	015064		MOV	#DH3,-(SP)
	021462	012746	015035		MOV	#DH2,-(SP)
	021466	012746	012352		MOV	#FMT4,-(SP)
	021472	012746	000003		MOV	#3,-(SP)
	021476	010600			MOV	SP,R0
	021500	104415			TRAP	C\$PNTX
	021502	062706	000010		ADD	#10,SP
242	021506			PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13		
	021506	013746	002274		MOV	LUR13,-(SP)
	021512	013746	002272		MOV	LUR12,-(SP)
	021516	013746	002270		MOV	LUR11,-(SP)
	021522	013746	002266		MOV	LUR10,-(SP)

	021526	012746	012365		MOV	#FMT5,-(SP)
	021532	012746	000005		MOV	#5,-(SP)
	021536	010600			MOV	SP,R0
	021540	104415			TRAP	C\$PNTX
	021542	062706	000014		ADD	#14,SP
243	021546			PRINTX	#FMT9,#DH4	
	021546	012746	015122		MOV	#DH4,-(SP)
	021552	012746	012514		MOV	#FMT9,-(SP)
	021556	012746	000002		MOV	#2,-(SP)
	021562	010600			MOV	SP,R0
	021564	104415			TRAP	C\$PNTX
	021566	062706	000006		ADD	#6,SP
244	021572			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	021572	013746	002304		MOV	LUR17,-(SP)
	021576	013746	002302		MOV	LUR16,-(SP)
	021602	013746	002300		MOV	LUR15,-(SP)
	021606	013746	002276		MOV	LUR14,-(SP)
	021612	012746	012415		MOV	#FMT6,-(SP)
	021616	012746	000005		MOV	#5,-(SP)
	021622	010600			MOV	SP,R0
	021624	104415			TRAP	C\$PNTX
	021626	062706	000014		ADD	#14,SP
245	021632			PRINTX	#FMT4,#DH7,#DH8	
	021632	012746	015224		MOV	#DH8,-(SP)
	021636	012746	015172		MOV	#DH7,-(SP)
	021642	012746	012352		MOV	#FMT4,-(SP)
	021646	012746	000003		MOV	#3,-(SP)
	021652	010600			MOV	SP,R0
	021654	104415			TRAP	C\$PNTX
	021656	062706	000010		ADD	#10,SP
246	021662			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16	
	021662	013746	002314		MOV	AX1.16,-(SP)
	021666	013746	002312		MOV	AX1.15,-(SP)
	021672	013746	002310		MOV	AX0.16,-(SP)
	021676	013746	002306		MOV	AX0.15,-(SP)
	021702	012746	012365		MOV	#FMT5,-(SP)
	021706	012746	000005		MOV	#5,-(SP)
	021712	010600			MOV	SP,R0
	021714	104415			TRAP	C\$PNTX
	021716	062706	000014		ADD	#14,SP
247	021722			PRINTX	#FMT9,#DH9	
	021722	012746	015263		MOV	#DH9,-(SP)
	021726	012746	012514		MOV	#FMT9,-(SP)
	021732	012746	000002		MOV	#2,-(SP)
	021736	010600			MOV	SP,R0
	021740	104415			TRAP	C\$PNTX
	021742	062706	000006		ADD	#6,SP
248	021746			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16	
	021746	013746	002324		MOV	AX3.16,-(SP)
	021752	013746	002322		MOV	AX3.15,-(SP)
	021756	013746	002320		MOV	AX2.16,-(SP)
	021762	013746	002316		MOV	AX2.15,-(SP)
	021766	012746	012415		MOV	#FMT6,-(SP)
	021772	012746	000005		MOV	#5,-(SP)
	021776	010600			MOV	SP,R0
	022000	104415			TRAP	C\$PNTX
	022002	062706	000014		ADD	#14,SP

249 022006
022006
022006 104423

ENDMSG

L10012: TRAP CSMSG

250
251
252
253
254

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

.SBTTL REPORT CODING SECTION

:///
:// THE REPORT CODING SECTION CONTAINS THE
:// 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
:///

022010
022010

BGNRPT

LSRPT::

022010
022010
022010

ENDRPT

L10013:

104425

TRAP CSRPT

.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```
.SBTTL INITIALIZE SECTION

:////////////////////
:/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
:/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
:////////////////////

      BGNINIT

      L$INIT::

MOV    SP,PSTACK      ;SAVE BASE-LEVEL STACK POINTER
CLR    SUBRPC         ;CLEAR SUBR CALL PC
CLR    DISILO         ;CLEAR CURRENT STATE OF DISSI
CLR    CHPTYP        ;CLEAR USYRT CHIP TYPE INDICATOR
CLR    ERROR1        ;CLEAR ERROR FLAGS
CLR    SAVLEN        ;CLEAR CHAR LENGTH FROM SETUP
TST    FRSTIM        ;SEE IF FIRST TIME THROUGH AFTER LOAD
BNE    6$            ;BR IF NOT
MOV    @#4,SAVE4      ;SAVE ERROR TRAP VECTOR
MOV    @#6,SAVE6
BR     9$
6$:   MOV    SAVE4,@#4      ;RESTORE ERROR TRAP VECTOR
      MOV    SAVE6,@#6
9$:   MOV    #1,FRSTIM     ;MARK FLAG FOR NEXT TIME THROUGH
:SEE IF PROGRAM JUST STARTED, BR IF YES
      READEF #EF.START

      MOV    #EF.START,RO
      TRAP  CSREFG

      BCOMPLETE      STARST
      BCS    STARST

:SEE IF PROGRAM JUST RESTARTED, BR IF YES
      READEF #EF.RESTART

      MOV    #EF.RESTART,RO
      TRAP  CSREFG

      BCOMPLETE      STARST
      BCS    STARST

:SEE IF THIS IS A NEW PASS, BR IF YES
      READEF #EF.NEW

      MOV    #EF.NEW,RO
      TRAP  CSREFG

      BCOMPLETE      NEWST
      BCS    NEWST

:SEE IF PROGRAM WAS JUST CONTINUED
      READEF #EF.CONTINUE

      MOV    #EF.CONTINUE,RO
      TRAP  CSREFG

      BCOMPLETE      ENDIT
      BCS    ENDIT

      BR     GETPRM

STARST:
      CLR    STARES      ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
:CLEAR DEVICE MAP
      CLR    DEVMAP

NEWST:
      MOV    #-1,LOGDEV  ;RESET LOGICAL DEVICE TO -1
      INC    FRSPAS     ;INCREMENT NO. OF PASSES AFTER LOAD
      INC    STARES     ;INCREMENT NO. OF PASSES SINCE STA OR RES
```

```

45 022206 012737 000001 002430      MOV    #BIT0,DEVPTX      ;INIT DEVICE MAP BIT POINTER
46                                     ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
47                                     ; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
48 022214                                     GETPRM:
49 022214 005237 002330 002430      INC    LOGDEV            ;INCREMENT LOGICAL DEVICE NUMBER
50 022220 023737 002330 002012      CMP    LOGDEV,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
51 022226 002360                                     BGE    NEWST            ;BR IF YES
52 022230                                     GPHARD LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
    022230 013700 002330                                     MOV    LOGDEV,R0
    022234 104442                                     TRAP  C$GPHRD
    022236 010001                                     MOV    R0,R1
53 022240                                     BCOMPLETE 10$         ;BR IF DEVICE AVAILABLE
    022240 103403                                     BCS    10$
54 022242 006337 002430      ASL    DEVPTX            ;SHIFT DEVICE MAP BIT POINTER
55 022246 000762      BR    GETPRM            ;SKIP THIS DEVICE
56 022250 053737 002430 002426 10$:  BIS    DEVPTX,DEVMAP    ;SET BIT FOR THIS DEVICE IN DEVICE MAP
57 022256 006337 002430      ASL    DEVPTX            ;SHIFT DEVICE MAP BIT POINTER
58 022262 062701 000002      ADD    #2,R1            ;INCREMENT R1 PAST MICROPROCESSOR TYPE
59 022266 011137 002436      MOV    (R1),MPCSR       ;STORE POINTER TO MICROPROCESSOR CSR'S
60 022272 011137 002440      MOV    (R1),BSEL1
61 022276 005237 002440      INC    BSEL1            ;GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
62 022302 011137 002442      MOV    (R1),SEL4
63 022306 062737 000004 002442      ADD    #4,SEL4          ;GET POINTER TO SEL4
64 022314 012137 002444      MOV    (R1)+,SEL6
65 022320 062737 000006 002444      ADD    #6,SEL6          ;STORE POINTER TO SEL6
66 022326 011137 002446      MOV    (R1),MPIVEC      ;GET MICROPROCESSOR INPUT INTRPT VECTOR
67 022332 012137 002450      MOV    (R1)+,MPOVEC
68 022336 062737 000004 002450      ADD    #4,MPOVEC        ;GET MICROPROCESSOR OUTPUT INTRPT VECTOR
69 022344 012137 002452      MOV    (R1)+,MPRIOR     ;GET MICROPROCESSOR DEVICE PRIORITY
70 022350 062701 000002      ADD    #2,R1            ;INCREMENT R1 PAST LU TYPE
71 022354 012137 002454      MOV    (R1)+,LUSWI1     ;GET LU SWITCH PACK #1
72 022360 012137 002456      MOV    (R1)+,LUSWI2     ;GET LU SWITCH PACK #2
73 022364 012137 002460      MOV    (R1)+,LUSWI3     ;GET LU SWITCH PACK #3
74 022370 012137 002462      MOV    (R1)+,TSTCON     ;GET TEST CONNECTOR INDICATOR
75 022374 011137 002464      MOV    (R1),BDRATE      ;GET BAUD RATE
76                                     ;SEE IF MANUAL INTERVENTION DESIRED BETWEEN UNITS FOR INSTALLATION OR REMOVAL
77                                     ; OF TEST CONNECTORS, BR IF NOT
78 022400 005737 002256      TST    MIFLAG
79 022404 001442      BEQ    22$
80                                     ;SEE IF MANUAL INTERVENTION ALLOWED BY SUPERVISOR
81 022406      MANUAL
    022406 104450      TRAP  C$MANI
82                                     ;BR IF ALLOWED
83 022410      BCOMPLETE 18$
    022410 103412      BCS    18$
84                                     ;PRINT MSG THAT OPERATOR INTERVENTION IS NOT ALLOWED
85 022412      PRINTF #FMT16
    022412 012746 022514      MOV    #FMT16,-(SP)
    022416 012746 000001      MOV    #1,-(SP)
    022422 010600      MOV    SP,R0
    022424 104417      TRAP  C$PNTF
    022426 062706 000004      ADD    #4,SP
86 022432 16$:  BREAK          ;HANG UNTIL ^C TYPED
    022432 104422      TRAP  C$BRK
87 022434 000776      BR    16$
88 022436 18$:
89                                     ;TYPE "INSTALL TEST CONNECTOR(S) ON UNIT AT ADRS XXXXXX"

```

```
90 022436          PRINTF #FMT17,MPCSR
   022436 013746 002436
   022442 012746 022636
   022446 012746 000002
   022452 010600
   022454 104417
   022456 062706 000006
91 022462 005037 002502
92 022466          CLR      REG2
93                20$:
94                ;ASK OPERATOR TO 'TYPE <Y> <CR> WHEN READY TO PROCEED''
   022466 104443          GMANIL TYPEY,REG2,1,NO
   022470 000404
   022472 002502
   022474 000120
   022476 022725
   022500 000001
   022502
95 022502 023727 002502 000001
96 022510 001366
97 022512
98 022512
99 022512
   022512
   022512 104411
100
101 022514 045 116 045 FMT16: .ASCII /%N%AMANUAL INTERVENTION IS NOT ALLOWED!%N/
   022517 101 115 101
   022522 116 125 101
   022525 114 040 111
   022530 116 124 105
   022533 122 126 105
   022536 116 124 111
   022541 117 116 040
   022544 111 123 040
   022547 116 117 124
   022552 040 101 114
   022555 114 117 127
   022560 105 104 041
   022563 045 116
102 022565 045 101 124 .ASCIZ /%ATYPE CONTROL-C (^C) <CR> TO PROCEED:%N/
   022570 131 120 105
   022573 040 103 117
   022576 116 124 122
   022601 117 114 055
   022604 103 040 050
   022607 136 103 051
   022612 040 074 103
   022615 122 076 040
   022620 124 117 040
   022623 120 122 117
   022626 103 105 105
   022631 104 072 045
   022634 116 000
103 022636 045 116 045 FMT17: .ASCIZ /%N%AINSTALL TEST CONNECTOR(S) ON UNIT AT ADRS : %06%N/
   022641 101 111 116
   022644 123 124 101
```

```
MOV MPCSR, -(SP)
MOV #FMT17, -(SP)
MOV #2, -(SP)
MOV SP, R0
TRAP C$PNTF
ADD #6, SP
```

```
CLR REG2
20$:
;ASK OPERATOR TO 'TYPE <Y> <CR> WHEN READY TO PROCEED''
GMANIL TYPEY,REG2,1,NO
```

```
TRAP C$GMAN
BR 10000$
.WORD REG2
.WORD T$CODE
.WORD TYPEY
.WORD 1
```

10000\$:

```
CMP REG2,#1
BNE 20$
22$:
ENDIT:
ENDINIT
```

L10015:

```
TRAP C$INIT
```

	022647	114	114	040	
	022652	124	105	123	
	022655	124	040	103	
	022660	117	116	116	
	022663	105	103	124	
	022666	117	122	050	
	022671	123	051	040	
	022674	117	116	040	
	022677	125	116	111	
	022702	124	040	101	
	022705	124	040	101	
	022710	104	122	123	
	022713	040	072	040	
	022716	040	045	117	
	022721	066	045	116	
	022724	000			
104	022725	124	131	120	TYPEY: .ASCIZ /TYPE <Y><CR> WHEN READY TO PROCEED /
	022730	105	040	074	
	022733	131	076	074	
	022736	103	122	076	
	022741	040	127	110	
	022744	105	116	040	
	022747	122	105	101	
	022752	104	131	040	
	022755	124	117	040	
	022760	120	122	117	
	022763	103	105	105	
	022766	104	040	000	

105 .EVEN
106
107
108
109
110

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

.SBTTL AUTO DROP UNIT SECTION

:/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
:/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.

BGNAUTO

L\$AUTO::

;ESTABLISH PRIORITY = 7
SETPRI #PRI07

MOV #PRI07,R0
TRAP C\$SPRI

MOV #6\$,@#4 ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
MOV #PRI07,@#6
TST @MPCSR ;ADDRESS SELO
BR 9\$;TAKE THIS BRANCH IF DEVICE RESPONDS

;COME HERE IF DEVICE CSR IS NON-EXISTENT
6\$: ADD #4,SP ;CLEAN UP THE STACK POINTER
DODU LOGDEV ;DROP THIS UNIT FROM TESTING

MOV LOGDEV,R0
TRAP C\$DODU

9\$: MOV SAVE4,@#4 ;RESTORE ERROR TRAP VECTOR
MOV SAVE6,@#6
ENDAUTO

L10016:

TRAP C\$AUTO

022772 012700 000340
022772 104441
022776 104441
023000 012737 023022 000004
023006 012737 000340 000006
023014 005777 157416
023020 000405
023022 062706 000004
023026 013700 002330
023032 104451
023034 013737 002404 000004
023042 013737 002406 000006
023050
023050 104461

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

.SBTTL CLEANUP CODING SECTION

:///
:// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
:// AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
:///

023052
023052

BGNCLN

L\$CLEAN::

023052
023052
023052 104412

ENDCLN

L10017: TRAP C\$CLEAN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

.SBTTL DROP UNIT SECTION

:/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:/ TO NO LONGER BE TESTED.

BGNDU

L\$DU::

;ISSUE UNIBUS RESET TO CLEAN UP
BRESET

TRAP C\$RESET

;PRINT 'UNIT XX DROPPED'
PRINTF #FMT27,LOGDEV

MOV LOGDEV,-(SP)
MOV #FMT27,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PRINTF
ADD #6,SP

ENDDU

L10020:

TRAP C\$DU

FMT27: .ASCIZ /%N%AUNIT %D2%A DROPPED%N/

.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

.SBTTL ADD UNIT SECTION

:/://////
:/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:/ TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
:/ 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
:/://////

023136
023136
023136
023136 104452

BGNAU
ENDAU

LSAU::
L10021: TRAP CS AU

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

.SBTTL HARDWARE TESTS

```
*****  
:SBTTL TEST 1 - BIT STUFFING TEST  
:  
:* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS  
:* INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,  
:* FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN  
:* CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS  
:* WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH  
:* ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED  
:* BY THE RECEIVER.  
:* PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,  
:* 177,376,377  
:*****
```

BGNTST

```
T1::  
MOV #24$,RETADR ;SET TEST EXIT ADRS FOR ERRORS  
JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S  
000  
CRC2!CRC1 ;BIT MODE, NO ERR DETECTION  
JSR PC,LDBYTS ;LOAD PAT S INTO TX SILO  
PATS  
16.  
MOV #TXEOM,TXWORD ;LOAD 2 EOM'S INTO TX SILO  
JSR PC,LDTXSI  
JSR PC,LDTXSI ;CLK MORE THAN ENTIRE MSG  
JSR PC,STPLU  
192.  
MOV #PATS,R1 ;INIT PAT S POINTER  
6$: MOVB (R1)+,8$  
JSR PC,CKDATA ;CHK A RCV'D CHAR  
8$: .WORD 0  
CMP R1,#PATS+15. ;SEE IF 15 CHARS CHECKED YET  
BLO 6$ ;BR IF NOT YET  
MOVB (R1),12$  
BIS #RXEBL,12$ ;GET SET TO CHK EBLK = 1  
JSR PC,CKDATA ;CHK LAST CHAR AND EBLK = 1  
12$: .WORD 0  
24$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP  
ENDTST  
L1002: TRAP C$ETST
```

```
*****  
:SBTTL TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST  
:  
:
```

```

55                    : * IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
56                    : * IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
57                    : * RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
58                    : * OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
59                    : * CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
60                    : * THEN, IC IS SET TO CLEAR THE ERROR, AND THIS IS VERIFIED.
61                    : *
62                    : * IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, AND A MASTER CLEAR
63                    : * IS ISSUED TO CLEAR THE ERROR, AND THIS IS VERIFIED.
64                    : *
65                    : *****
  
```

```

65 023270
   023270
66 023270 012737 023636 002346                    MOV    #24$,RETADR                    ;SET TEST EXIT ADRS FOR ERRORS
67                    :-----
68                    : CAUSE OVRR, SET IC TO CLEAR IT
69                    :-----
70 023276 004737 005212                    JSR    PC,INITRN                    ;MST CLR, LOAD 2 SOM'S
71 023302 000226                    SYNCH
72 023304 000311                    CRC2!CRC1!STRIP!DDCMP                ;DDCMP, NO ERR DET
73 023306 004737 010720                    JSR    PC,LODSIL                    ;LOAD 64 001 CHARS INTO TX SILO
74 023312 000001                    001
75 023314 000100                    64.
76 023316 004737 006752                    JSR    PC,RCV1ST                    ;CLOCK UNTIL FIRST DATA CHAR RCV'D
77 023322 000030                    24.
78 023324 004737 004726                    JSR    PC,STPLU                    ;CLOCK UNTIL 59 MORE RCV'D
79 023330 000730                    472.
80 023332 004737 010720                    JSR    PC,LODSIL                    ;LOAD 60 MORE INTO TX SILO
81 023336 000001                    001
82 023340 000074                    60.
83 023342 004737 004726                    JSR    PC,STPLU                    ;CLK 8 MORE TIMES TO FORCE UNDERRUN
84 023346 000100                    64.
85 023350 012701 000100                    MOV    #64.,R1                    ;READ AND CHK 64 CHARS FROM RCV SILO
86 023354 004737 007266                    6$: JSR    PC,CKDATA
87 023360 000001                    001
88 023362 000000                    0
89 023364 005301                    DEC    R1
90 023366 001372                    BNE    6$
91 023370 004737 007266                    JSR    PC,CKDATA                    ;READ CHAR, CHK OVRR = 1
92 023374 004001                    4001
93 023376 000010                    8.
94 023400 004737 007266                    JSR    PC,CKDATA                    ;READ CHAR, CHK OVRR STILL = 1
95 023404 004001                    4001
96 023406 000010                    8.
97 023410 012737 000012 002364                    MOV    #12,REGNUM                    ;SET REG NO. = 12
98 023416 012737 000200 002352                    MOV    #IC,WRIBYT
99 023424 004737 003422                    JSR    PC,WRITLU                    ;SET IC TO CLEAR RCVR
100 023430 004737 006672                    JSR    PC,RDRXSI                    ;READ RCV SILO
101 023434 132737 000010 002415                    BITB    #OVRR,RXWORD+1                ;CHK FOR OVRR CLEARED
102 023442 001407                    BEQ    8$                    ;BR IF OVRR CLEARED
103 023444 004737 004200                    JSR    PC,GETALL                    ;GET REGS FOR PRINTOUT
104                    :REPORT OVRR NOT CLEARED
105 023450                    ERRDF    41,EM41,ERR7
   023450 104455                    TRAP    C$ERDF
   023452 000051                    .WORD    41
   023454 014032                    .WORD    EM41
   023456 020216                    .WORD    ERR7
106 023460 000466                    BR      24$
  
```

```

107 023462
108
109
110
111 023462 004737 005212      8$:
112 023466 000226              -----
113 023470 000311              : CAUSE OVRR, SET MST CLR TO CLEAR IT
114 023472 004737 010720      -----
115 023476 000001              JSR      PC,INITRN      ;MST CLR, LOAD 2 SOM'S
116 023500 000100              SYNCH
117 023502 004737 006752      CRC2!CRC1!STRIP!DDCMP      ;DDCMP, NO ERR DET
118 023506 000030              JSR      PC,LODSIL      ;LOAD 64 001 CHARS INTO TX SILO
119 023510 004737 004726      001
120 023514 000730              64.
121 023516 004737 010720      JSR      PC,RCV1ST      ;CLOCK UNTIL FIRST DATA CHAR RCV'D
122 023522 000001              24.
123 023524 000074              JSR      PC,STPLU      ;CLOCK UNTIL 59 MORE RCV'D
124 023526 004737 004726      472.
125 023532 000100              JSR      PC,LODSIL      ;LOAD 60 MORE INTO TX SILO
126 023534 012701 000100      001
127 023540 004737 007266      60.
128 023544 000001              JSR      PC,STPLU      ;CLK 8 MORE TIMES TO FORCE UNDERRUN
129 023546 000000              64.
130 023550 005301              MOV      #64.,R1      ;READ AND CHK 64 CHARS FROM RCV SILO
131 023552 001372              JSR      PC,CKDATA
132 023554 004737 007266      0
133 023560 004001              DEC      R1
134 023562 000010              BNE      9$
135 023564 004737 007266      JSR      PC,CKDATA      ;READ CHAR, CHK OVRR = 1
136 023570 004001              4001
137 023572 000010              8.
138 023574 012737 000012 002364  JSR      PC,CKDATA      ;READ CHAR, CHK OVRR STILL = 1
139 023602 004737 003276      4001
140 023606 004737 006672      8.
141 023612 132737 000010 002415  MOV      #12,REGNUM      ;SET REG NO. = 12
142 023620 001406              JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
143 023622 004737 004200      JSR      PC,RDRXSI      ;READ RCV SILO
144
145 023626              BITB     #OVRR,RXWORD+1 ;CHK FOR OVRR CLEARED
146 023626 104455              BEQ     24$             ;BR IF OVRR CLEARED
147 023630 000051              JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
148 023632 014032              ;REPORT OVRR NOT CLEARED
149 023634 020216              ERRDF   41,EM41,ERR7
150 023636 004737 003276      24$:
151 023642              JSR      PC,MSTCLR      ;ISSUE CLEAN UP MST CLR
152 023642              ENDTST
153
154
155
156
157

```

```

*****
:SBTTL      TEST 3 - ABORT SEQUENCE TEST
:
:* SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
:* TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).

```

158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

```

: * AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
: * SET THE ABORT BIT (REG 11).
: * ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
: * AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
: * ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
: * IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
: * RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
: *
: * REPEAT THE ABOVE SEQUENCE, SET IC, AND CHECK THAT
: * THIS CLEARS RAB.
: *
: * REPEAT THE ABOVE SEQUENCE, ISSUE MASTER CLEAR, CHECK THAT THIS
: * CLEARS RAB.
: *
: *****

```

173 023644
023644
174 023644 012737 024174 002346

```

: *****
: BGNTST

```

```

MOV #24$,RETADR ;SET TEST EXIT ADRS FOR ERRORS
T3::
-----

```

```

: CAUSE ABORT, START NEW MSG TO CLEAR IT
-----

```

178 023652 004737 005212
179 023656 000000
180 023660 000000
181 023662 004737 010470
182 023666 002732
183 023670 000014
184 023672 004737 006752
185 023676 000060
186 023700 004737 007266
187 023704 000001
188 023706 000010
189 023710 004737 007266
190 023714 003001
191 023716 000000
192 023720 004737 006232
193 023724 000000
194 023726 004737 006752
195 023732 000060
196 023734 004737 007266
197 023740 000003
198 023742 000000
199

```

JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
000
000 ;BIT MODE, CRC
JSR PC,LODMSG ;LOAD MSG INTO TX SILO
MSG3
12.
JSR PC,RCV1ST ;CLK AND RCV FIRST DATA CHAR
48.
JSR PC,CKDATA ;CHK CHR = 001, CLK ABORT CHAR
001
8.
JSR PC,CKDATA ;CHK FOR RAB, EBLK, AND 001 CHAR
RXABT!RXEBL!001
0
JSR PC,IACTIV ;CHK FOR IACT = 0
0
JSR PC,RCV1ST ;CLK AND RCV NEW MSG
48.
JSR PC,CKDATA ;CHK CHAR = 003
003
0
-----

```

```

: CAUSE ABORT, SET IC TO CLEAR IT
-----

```

200
201
202 023744 004737 005212
203 023750 000000
204 023752 000000
205 023754 004737 010470
206 023760 002732
207 023762 000014
208 023764 004737 006752
209 023770 000060
210 023772 004737 007266
211 023776 000001
212 024000 000010
213 024002 004737 007266

```

JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
000
000 ;BIT MODE, CRC
JSR PC,LODMSG ;LOAD MSG INTO TX SILO
MSG3
12.
JSR PC,RCV1ST ;CLK AND RCV FIRST DATA CHAR
48.
JSR PC,CKDATA ;CHK CHR = 001, CLK ABORT CHAR
001
8.
JSR PC,CKDATA ;CHK FOR RAB, EBLK, AND 001 CHAR

```

```
214 024006 003001 RXABT!RXEBL!001
215 024010 000000 0
216 024012 012737 000012 002364 MOV #12,REGNUM ;SET REG NO. = 12
217 024020 012737 000200 002352 MOV #IC,WRIBYT
218 024026 004737 003422 JSR PC,WRITLU ;SET IC TO CLEAR RCVR
219 024032 004737 006672 JSR PC,RDRXSI ;READ RCV SILO
220 024036 132737 000004 002415 BITB #RAB,RXWORD+1 ;CHK FOR RAB CLEARED
221 024044 001407 BEQ 8$ ;BR IF RAB CLEARED
222 024046 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
223 ;REPORT RAB NOT CLEARED
224 024052 ERRDF 39,EM39,ERR7
024052 104455 TRAP C$ERDF
024054 000047 .WORD 39
024056 013776 .WORD EM39
024060 020216 .WORD ERR7
225 024062 000444 BR 24$
226 024064 8$:
-----
228 ;CAUSE ABORT, ISSUE MASTER CLEAR TO CLEAR IT
-----
230 024064 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
231 024070 000000 000
232 024072 000000 000 ;BIT MODE, CRC
233 024074 004737 010470 JSR PC,LODMSG ;LOAD MSG INTO TX SILO
234 024100 002732 MSG3
235 024102 000014 12.
236 024104 004737 006752 JSR PC,RCV1ST ;CLK AND RCV FIRST DATA CHAR
237 024110 000060 48.
238 024112 004737 007266 JSR PC,CKDATA ;CHK CHR = 001, CLK ABORT CHAR
239 024116 000001 001
240 024120 000010 8.
241 024122 004737 007266 JSR PC,CKDATA ;CHK FOR RAB, EBLK, AND 001 CHAR
242 024126 003001 RXABT!RXEBL!001
243 024130 000000 0
244 024132 012737 000012 002364 MOV #12,REGNUM ;SET REG NO. = 12
245 024140 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
246 024144 004737 006672 JSR PC,RDRXSI ;READ RCV SILO
247 024150 132737 000004 002415 BITB #RAB,RXWORD+1 ;CLK FOR RAB CLEARED
248 024156 001406 BEQ 24$ ;BR IF RAB CLEARED
249 024160 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
250 ;REPORT RAB NOT CLEARED
251 024164 ERRDF 39,EM39,ERR7
024164 104455 TRAP C$ERDF
024166 000047 .WORD 39
024170 013776 .WORD EM39
024172 020216 .WORD ERR7
252 024174 004737 003276 24$: JSR PC,MSTCLR ;ISSUE MST CLR TO CLEAN UP
253 024200 ENDTST
024200 L10024:
024200 104401 TRAP C$ETST
254
255
256
257
258
259
260 ;*****
;SBTTL TEST 4 - ABORT AND IDLE FLAGS TEST
```

```

261
262
263
264
265
266
267 024202
    024202
268 024202 012737 024256 002346      MOV    #24$,RETADR    ;SET TEST EXIT ADRS FOR ERRORS
269 024210 004737 005212              JSR    PC,INITRN     ;MST CLR, LOAD 2 SOM'S
270 024214 000000                      000
271 024216 000040                      IDLE    ;BIT MODE, NO ERROR DET.
272 024220 004737 010470              JSR    PC,LODMSG     ;LOAD MSG INTO TX SILO
273 024224 002732                      MSG3
274 024226 000005                      5
275 024230 004737 006752              JSR    PC,RCV1ST    ;CLK AND RCV FIRST DATA CHAR
276 024234 000060                      48.
277 024236 004737 007266              JSR    PC,CKDATA    ;CHK CHR = 001, CLK FLAG CHAR
278 024242 000001                      001
279 024244 000010                      8.
280 024246 004737 007266              JSR    PC,CKDATA    ;CHK RAB = 0, EBLK = 1
281 024252 001001                      RXEBL!001
282 024254 000000                      0
283 024256 004737 003276      24$: JSR    PC,MSTCLR    ;ISSUE MASTER CLEAR
284 024262
    024262
    024262 104401
                                L10025: TRAP C$ETST
    
```

285
286
287
288
289
290
291
292
293
294
295
296
297
298

```

:*****
:SBTTL      TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
:
:* A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
:* IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
:* UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
:* CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
:* THE DEVICE IDLES ABORT CHARACTERS.
:*****
BGNTST
    
```

```

299 024264
    024264
300 024264 012737 024346 002346      MOV    #24$,RETADR    ;SET TEST EXIT ADRS FOR ERRORS
301 024272 004737 005212              JSR    PC,INITRN     ;MST CLR, LOAD 2 SOM'S
302 024276 000000                      000
303 024300 000000                      000
304 024302 012737 000100 002416      MOV    #TXEN,DISILO  ;SET TX ENB TO KEEP RTS HIGH
305 024310 004737 010720              JSR    PC,LODSIL     ;LOAD 4 001 CHARS INTO TX SILO
306 024314 000001                      001
307 024316 000004                      4
308 024320 004737 006752              JSR    PC,RCV1ST    ;CLK AND RCV FIRST CHAR
309 024324 000060                      48.
310 024326 004737 007266              JSR    PC,CKDATA    ;CHK DATA = 001, CLOCK ABORT CHAR
311 024332 000001                      001
312 024334 000011                      9.
313 024336 004737 007266              JSR    PC,CKDATA    ;CHK FOR RAB, EBLK, AND 001 CHAR
    
```

```

314 024342 003001          RXABT!RXEBL!001
315 024344 000000          0
316 024346 004737 003276  24$: JSR    PC,MSTCLR    ;ISSUE MASTER CLEAR
317 024352          ENDTST
    024352
    024352 104401          L10026: TRAP  C$ETST
    
```

318
319
320
321
322
323
324
325
326
327
328
329
330
331

```

:*****
:SBTTL      TEST 6 - RECEIVER DISABLE TEST
:*
:* TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
:* ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
:* TO BE RECEIVED, IC IS SET.
:* THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
:* THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
:*****
BGNTST
    
```

```

332 024354
    024354
333 024354 012737 024542 002346      MOV    #24$,RETADR    ;SET TEST EXIT ADRS FOR ERRORS T6::
334 024362 004737 005212          JSR    PC,INITRN     ;MST CLR, LOAD 2 SOM'S
335 024366 000000          000
336 024370 000000          000          ;BIT MODE, CRC
337 024372 004737 010720          JSR    PC,LODSIL     ;LOAD 5 252 CHARS
338 024376 000252          252
339 024400 000005          5
340 024402 004737 006752          JSR    PC,RCV1ST    ;CLK AND RCV FIRST DATA CHAR
341 024406 000060          48.
342 024410 004737 004726          JSR    PC,STPLU     ;CLK TO MIDDLE OF 2ND CHAR
343 024414 000004          4
344 024416 012737 000012 002364      MOV    #12,REGNUM    ;SET REG NO. = 12
345 024424 012737 000200 002352      MOV    #IC,WRIBYT
346 024432 004737 003422          JSR    PC,WRITLU    ;SET IC IN REG 12
347 024436 004737 006232          JSR    PC,IACTIV    ;CHK IACT = 0
348 024442 000000          0
349 024444 004737 005746          JSR    PC,ISIRDY    ;CHK ICIR = 1, IRDY = 0
350 024450 000001          1
351 024452 005037 002370          CLR    GOODAT      ;SET EXPECTED DATA = 0
352 024456 005037 002372          CLR    BADDAT
353 024462 004737 006672          JSR    PC,RDRXSI    ;READ RCV SILO
354 024466 105737 002414          TSTB   RXWORD      ;SEE IF SILO BITS 0-7 = 000
355 024472 001404          9$ BEQ    9$          ;BR IF YES
356 024474 012737 000010 002364      MOV    #10,REGNUM   ;SET REG NO. = 10
357 024502 000406          BR     12$
358 024504 105737 002415          9$: TSTB   RXWORD+1   ;SEE IF SILO BITS 8-11 = 000
359 024510 001414          BEQ    24$          ;BR IF YES
360 024512 012737 000012 002364      MOV    #12,REGNUM   ;SET REG NO. = 12
361 024520 113737 002414 002372 12$: MOVB   RXWORD,BADDAT ;GET ACTUAL DATA
362 024526 004737 004200          JSR    PC,GETALL    ;GET REGS FOR PRINTOUT
363          ;REPORT RCV SILO NOT CLEARED BY IC
364 024532          ERRDF 46,EM46,ERR2
    024532 104455          TRAP  C$ERDF
    024534 000056          .WORD 46
    024536 014167          .WORD EM46
    
```


365 024540 015360
366 024542 004737 003276
024546
024546 104401

24\$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
ENDTST

L10027:
TRAP C\$ETST

367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383

```
*****
:SBTTL      TEST 7 - ASSEMBLED BIT COUNT TEST
:*
:* THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A
:* DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
:* BITS :
:* A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
:* 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
:* TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
:* IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
:* THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
*****
```

384 024550
024550
385 024550 012737 025306 002346
386 024556 004737 005212
387 024562 000000
388 024564 000000
389 024566 012701 000100
390 024572 004737 003276
391 024576 004737 010276
392 024602 000000
393 024604 000300
394 024606 000000
395 024610 000000
396 024612 012737 000014 002364
397 024620 012737 000140 002352
398 024626 004737 003422
399 024632 012737 000140 002416
400 024640 012737 000012 002364
401 024646 112737 000040 002352
402 024654 004737 003422
403 024660 012737 000002 002366
404 024666 105037 002360
405 024672 112737 000001 002362
406 024700 004737 003764
407 024704 005004
408 024706 012737 000011 002364
409 024714 004737 004726
410 024720 000001
411 024722 004737 003344
412 024726 132737 000100 002350
413 024734 001014
414 024736 005204
415 024740 020427 000004
416 024744 002763
417 024746 004737 004200

```
BGNTST
T7::
MOV #24$,RETADR ;SET TEST EXIT ADRS FOR ERRORS
JSR PC,INITRN ;FIND OUT WHICH USYRT CHIP
0
0
MOV #TXLEN1,R1 ;SET INITIAL TX LENGTH TO 2 BITS
6$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR
JSR PC,SETUP ;PROGRAM THE USYRT
000
CRC2!CRC1
000
000
MOV #14,REGNUM ;SET REG NO. = 14
MOV #TXEN!DISSI,WRIBYT
JSR PC,WRITLU ;SET TXEN AND DISSI IN REG 14
MOV #TXEN!DISSI,DISILO ;SET DISABLE SILO FLAG
MOV #12,REGNUM ;SET LU REG NO. = 12
MOVB #LULP,WRIBYT
JSR PC,WRITLU ;SET LULP IN REG 12
MOV #2,AXNUM ;SET AX BYTE NO. = 2
CLRB WAX15
MOVB #TSOM,WAX16
JSR PC,WRITAX ;LOAD SOM CHAR
CLR R4 ;INIT COUNTER
MOV #11,REGNUM ;SET REG NO. = 11
7$: JSR PC,STPLU ;CLOCK LU FOR A CYCLE
1
JSR PC,READLU ;READ REG 11
BITB #OACT,REDBYT ;SEE IF OACT SET YET
BNE 10$ ;BR IF OACT SET
INC R4 ;INCR COUNTER
CMP R4,#4 ;SEE IF COUNT TOO BIG
BLT 7$ ;BR IF NOT
JSR PC,GETALL ;GET REGS FOR PRINTOUT
```

```

418
419 024752      ;REPORT OACT NOT SET
      024752 104455 ERRDF 11,EM11,ERR7
      024754 000013
      024756 013372
      024760 020216
420 024762 000137 025306
421 024766 004737 003764
422 024772 004737 004726
423 024776 000010
424 025000 105037 002362
425 025004 004737 003764
426 025010 004737 004726
427 025014 000010
428 025016 004737 003764
429 025022 004737 004726
430 025026 000010
431 025030 012737 000006 002366
432 025036 010137 002362
433 025042 004737 003764
434 025046 012737 000002 002366
435 025054 105037 002362
436 025060 005737 002420
437 025064 001403
438 025066 112737 000002 002362
439 025074 004737 003764
440 025100 004737 004726
441 025104 000010
442 025106 005737 002420
443 025112 001005
444 025114 112737 000002 002362
445 025122 004737 003764
446 025126 012737 000001 002366
447 025134 005003
448 025136 004737 003576
449 025142 132737 000002 002356
450 025150 001016
451 025152 004737 004726
452 025156 000001
453 025160 005203
454 025162 020327 000023
455 025166 002763
456 025170 004737 004200
457
458 025174      ;REPORT REOM NOT SET
      025174 104455 ERRDF 31,EM31,ERR10
      025176 000037
      025200 013641
      025202 021356
459 025204 000440
460 025206 013702 002356
461 025212 042702 000217
462 025216 006102
463 025220 120201
464 025222 001421
465 025224 010137 002370
466 025230 006237 002370
    
```

```

TRAP C$ERDF
.WORD 11
.WORD EM11
.WORD ERR7
    
```

```

TRAP C$ERDF
.WORD 31
.WORD EM31
.WORD ERR10
    
```

```

10$:
    JMP 24$
    JSR PC,WRITAX ;LOAD ANOTHER SOM CHAR
    JSR PC,STPLU ;CLK FIRST FLAG
    8.
    CLRB WAX16
    JSR PC,WRITAX ;LOAD FIRST 000 CHAR
    JSR PC,STPLU ;CLK SECOND FLAG
    8.
    JSR PC,WRITAX ;LOAD SECOND 000 CHAR
    JSR PC,STPLU ;CLK FIRST 000 CHAR
    8.
    MOV #6,AXNUM ;SET AX BYTE NO. FOR AX 3
    MOV R1,WAX16 ;GET TX CHAR LENGTH
    JSR PC,WRITAX ;SET TX CHAR LENGTH
    MOV #2,AXNUM ;SET AX BYTE NO. = 2
    CLRB WAX16
    TST CHPTYP ;SEE IF SIG USYRT
    BEQ 5$ ;BR IF YES
    MOVB #TEOM,WAX16 ;SET TEOM WITH LAST DATA CHAR
    JSR PC,WRITAX ;LOAD 3RD 000 CHAR
    JSR PC,STPLU ;CLK 2ND 000 CHAR
    8.
    TST CHPTYP ;SEE IF SIG USYRT
    BNE 16$ ;BR IF NOT
    MOVB #TEOM,WAX16
    JSR PC,WRITAX ;LOAD AN EOM CHAR
    MOV #1,AXNUM ;SET AX BYTE NO. = 1
    CLR R3
    JSR PC,READAX ;READ AX0
    BITB #REOM,RAX16 ;CHK FOR REOM = 1
    BNE 14$ ;BR IF YES
    JSR PC,STPLU ;CLOCK LU FOR A CYCLE
    1
    INC R3 ;INCR COUNT
    CMP R3,#19. ;SEE IF COUNT TOO BIG
    BLT 12$ ;BR IF NOT
    JSR PC,GETALL ;GET REGS FOR PRINTOUT
    
```

```

14$:
    BR 24$
    MOV RAX16,R2 ;GET AX0-16 CONTENTS
    BIC #217,R2 ;MASK OFF ALL BUT ASSEMB BIT COUNT
    ROL R2
    CMPB R2,R1 ;CHK FOR CORRECT ASSEMB BIT COUNT
    BEQ 9$ ;BR IF MATCH
    MOV R1,GOODAT ;SET EXPECTED DATA
    ASR GOODAT
    
```

467 025234 152737 000002 002370
 468 025242 013737 002356 002372
 469 025250 004737 004200
 470
 471 025254
 025254 104455
 025256 000057
 025260 014222
 025262 015666
 472 025264 000410
 473 025266 005701
 474 025270 001406
 475 025272 062701 000040
 476 025276 042701 000400
 477 025302 000137 024572
 478 025306 005037 002416
 479 025312 004737 003276
 480 025316
 025316
 025316 104401

```

BISB #REOM,GOODAT
MOV RAX16,BADDAT ;SET ACTUAL DATA
JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT ASSEMB BIT COUNT INCORRECT
ERRDF 47,EM47,ERR3

BR 24$
9$: TST R1 ;SEE IF ALL DONE YET
BEQ 24$ ;BR IF YES
ADD #TXLENO,R1 ;INCR TX LENGTH
BIC #400,R1 ;MASK OFF OVERFLOW IF 8 BITS
JMP 6$ ;PROCEED
24$: CLR DISILO
JSR PC,MSTCLR ;ISSUE MASTER CLR TO CLEAN UP
  
```

TRAP C\$ERDF
 .WORD 47
 .WORD EM47
 .WORD ERR3

L10030:

TRAP C\$ETST

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

```

*****
:SBTTL TEST 8 - SECONDARY STATION ADDRESS BIT TEST
:*
:* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
:* BIT MODE, AND THE SECA BIT (REG 17) IS SET.
:* 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
:* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
:* RECEIVED.
:*
:* NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
:* WITH THE FIRST WORD OF DATA PATTERN T. 2 FLAGS ARE SENT,
:* FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
:* AND A TERMINATING FLAG.
:* THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
:* ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
:*
:* THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
:* DATA PATTERN T.
:* PATTERN T = 000,125,252,176,177
*****
  
```

BGNTST

T8::

 : SEND MSG WITH INVALID SEC STA ADRS

BGNSUB

T8.1:

TRAP C\$BSUB

506 025320
 025320
 507
 508
 509
 510 025320
 025320
 025320 104402
 511 025322 012737 025420 002346
 512 025330 004737 005212
 513 025334 000000
 514 025336 000020

```

MOV #3$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
000 ;SEC ADRS = 000
SECA ;BIT MODE, CRC, SEC ADRS MODE
  
```

```

515 025340 004737 010720 JSR PC,LODSIL ;LOAD 252 INTO TX SILO
516 025344 000252 252
517 025346 000001 1
518 025350 004737 010720 JSR PC,LODSIL ;LOAD 000 DATA INTO TX SILO
519 025354 000000 000
520 025356 000001 1
521 025360 004737 010720 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
522 025364 001000 TXEOM
523 025366 000002 2
524 025370 004737 004726 JSR PC,STPLU ;TRANSMIT THE MSG
525 025374 000060 48.
526 025376 004737 006232 JSR PC,IACTIV ;CHK IACT = 0
527 025402 000000 0
528 025404 004737 004726 JSR PC,STPLU ;CLOCK 8 MORE CYCLES
529 025410 000010 8.
530 025412 004737 006232 JSR PC,IACTIV ;CHK IACT = 0
531 025416 000000 0
532 025420 3$:
533 025420 ENDSUB
    025420 104403 L10032: TRAP C$ESUB
534
535 :-----: SEND MSG'S WITH VALID SEC ADRS'S FROM PAT T
536 :-----:
537 025422 012701 002662 MOV #PATT,R1 ;INIT DATA PATTERN POINTER
538 025426 A11:
539 025426 BGNSUB
    025426 104402 T8.2: TRAP C$BSUB
540 025430 012737 025540 002346 MOV #24$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
541 025436 111137 025456 MOVB (R1),5$ ;SET SEC ADRS
542 025442 111137 025466 MOVB (R1),6$ ;SET FIRST DATA CHAR
543 025446 111137 025524 MOVB (R1),9$ ;SET EXPECTED DATA CHAR
544 025452 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
545 025456 000000 5$: .WORD 0
546 025460 000020 SECA ;BIT MODE, CRC, SEC ADRS MODE
547 025462 004737 010720 JSR PC,LODSIL ;LOAD 1ST DATA CHAR INTO TX SILO
548 025466 000000 6$: .WORD 0
549 025470 000001 1
550 025472 004737 010720 JSR PC,LODSIL ;LOAD A 000 CHAR INTO TX SILO
551 025476 000000 000
552 025500 000001 1
553 025502 004737 010720 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
554 025506 001000 TXEOM
555 025510 000002 2
556 025512 004737 006752 JSR PC,RCV1ST ;CLOCK AND RCV FIRST DATA CHAR
557 025516 000060 48.
558 025520 004737 007266 JSR PC,CKDATA ;CHK FOR CORRECT RCV'D SEC STA ADRS
559 025524 000000 9$: .WORD 0
560 025526 000011 9.
561 025530 004737 007266 JSR PC,CKDATA ;READ AND CHK 000 CHAR, EBLK=1, BCC=0
562 025534 101000 CRCCHK!RXEBL!000
563 025536 000000 0
564 025540 24$:
565 025540 ENDSUB
    025540 104403 L10033: TRAP C$ESUB
    
```

```

566 025542 005201          INC      R1          ;INCR PATTERN POINTER
567 025544 020127 002667  CMP      R1,#ENDPAT ;SEE IF ALL DONE YET
568 025550 103726          BLO     A11         ;BR IF NO
569 025552 004737 003276  JSR     PC,MSTCLR   ;ISSUE MASTER CLEAR
570 025556          ENDTST
    025556
    025556 104401          L10031: TRAP  C$ETST
    
```

571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588

```

:*****
:SBTTL      TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST
:*
:* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
:* BIT MODE, AND THE SECA BIT IS SET.
:* 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
:* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
:* RECEIVED.
:* NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
:* ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
:* THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
:* IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
:*****
:BGNTST
    
```

589 025560
025560

T9::

```

:-----
: SET SEC ADR = 000, SEND ADR = 377, WITH RDALL = 0
:-----
    
```

590
591
592

```

593 025560          BGNSUB
    025560
    025560 104402
594 025562 012737 025660 002346  MOV     #3$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
595 025570 004737 005212  JSR     PC,INITRN  ;MST CLR, LOAD 2 SOM'S
596 025574 000000          000          ;SEC ADRS = 000
597 025576 000020          SECA
598 025600 004737 010720  JSR     PC,LODSIL  ;BIT MODE, CRC, SEC ADRS MODE
599 025604 000377          377          ;LOAD 377 INTO TX SILO
600 025606 000001          1
601 025610 004737 010720  JSR     PC,LODSIL  ;LOAD 125 DATA INTO TX SILO
602 025614 000125          125
603 025616 000001          1
604 025620 004737 010720  JSR     PC,LODSIL  ;LOAD 2 EOM'S INTO TX SILO
605 025624 001000          TXEOM
606 025626 000002          2
607 025630 004737 004726  JSR     PC,STPLU   ;TRANSMIT THE MSG
608 025634 000060          48.
609 025636 004737 006232  JSR     PC,IACTIV  ;CHK IACT = 0
610 025642 000000          0
611 025644 004737 004726  JSR     PC,STPLU   ;CLOCK 8 MORE CYCLES
612 025650 000010          8.
613 025652 004737 006232  JSR     PC,IACTIV  ;CHK IACT = 0
614 025656 000000          0
615 025660
616 025660
    
```

```

:-----
:BGNSUB
    T9.1: TRAP  C$BSUB
    MOV     #3$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
    JSR     PC,INITRN  ;MST CLR, LOAD 2 SOM'S
    000          ;SEC ADRS = 000
    SECA
    JSR     PC,LODSIL  ;BIT MODE, CRC, SEC ADRS MODE
    377          ;LOAD 377 INTO TX SILO
    1
    JSR     PC,LODSIL  ;LOAD 125 DATA INTO TX SILO
    125
    1
    JSR     PC,LODSIL  ;LOAD 2 EOM'S INTO TX SILO
    TXEOM
    2
    JSR     PC,STPLU   ;TRANSMIT THE MSG
    48.
    JSR     PC,IACTIV  ;CHK IACT = 0
    0
    JSR     PC,STPLU   ;CLOCK 8 MORE CYCLES
    8.
    JSR     PC,IACTIV  ;CHK IACT = 0
    0
3$:
    ENDSUB
    L10035:
    
```

```

025660 104403 TRAP C$ESUB
617
618
619
620 025662 BGNSUB
025662 T9.2:
025662 104402 TRAP C$BSUB
621 025664 012737 025760 002346 MOV #24$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
622 025672 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
623 025676 000000 000 ;SEC ADRS = 000
624 025700 000024 SECA!RDALL ;BIT MODE, CRC, SEC ADRS MODE, RDALL
625 025702 004737 010720 JSR PC,LODSIL ;LOAD 1ST DATA CHAR INTO TX SILO
626 025706 000377 377
627 025710 000001 1
628 025712 004737 010720 JSR PC,LODSIL ;LOAD A 125 CHAR INTO TX SILO
629 025716 000125 125
630 025720 000001 1
631 025722 004737 010720 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
632 025726 001000 TXEOM
633 025730 000002 2
634 025732 004737 006752 JSR PC,RCV1ST ;CLOCK AND RCV FIRST DATA CHAR
635 025736 000060 48.
636 025740 004737 007266 JSR PC,CKDATA ;CHK FOR 377 CHAR RCV'D
637 025744 000377 377
638 025746 000010 8.
639 025750 004737 007266 JSR PC,CKDATA ;READ AND CHK 125 CHAR, EBLK=1, BCC=0
640 025754 101125 CRCCHK!RXEBL!125
641 025756 000000 0
642 025760 24$:
643 025760 ENDSUB
025760 L10036:
025760 104403 TRAP C$ESUB
644 025762 ENDTST
025762 L10034:
025762 104401 TRAP C$ETST
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663 025764
025764
664 025764 012737 026112 002346 MOV #15$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
665 025772 004737 005212 JSR PC,INITRN ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
T10::

```

```

*****
SBTTL TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
* SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
* TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
* CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
* THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
* AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
* 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
* AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
* THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
*****
BGNTST

```

```

666 025776 000226          SYNCH
667 026000 000011          STRIP!DDCMP
668 026002 004737 010470  JSR    PC,LODMSG      ;LOAD MSG INTO TX SILO
669 026006 002722          MSG2+4
670 026010 000004          4
671 026012 004737 004726  JSR    PC,STPLU       ;CLOCK LU UNTIL 2ND BIT OF 000 CHAR
672 026016 100021          CHPCHK!17.
673 026020 004737 007164  JSR    PC,STPERR     ;SET IERR 1 CYCLE
674 026024 000011          STRIP!DDCMP
675 026026 000001          1
676 026030 004737 004726  JSR    PC,STPLU       ;CLOCK LU UNTIL 4TH BIT OF 000 CHAR
677 026034 000001          1
678 026036 004737 007164  JSR    PC,STPERR     ;SET IERR FOR 2 CYCLES
679 026042 000011          STRIP!DDCMP
680 026044 000002          2
681 026046 004737 004726  JSR    PC,STPLU       ;CLOCK LU UNTIL 1ST BIT OF 377 CHAR
682 026052 000003          3
683 026054 004737 007164  JSR    PC,STPERR     ;SET IERR FOR 4 CYCLES
684 026060 000011          STRIP!DDCMP
685 026062 000004          4
686 026064 004737 006752  JSR    PC,RCV1ST     ;CLOCK AND RCV 1ST CHAR
687 026070 000014          12.
688 026072 004737 007266  JSR    PC,CKDATA     ;READ AND COMPARE 1ST CHAR TO 032
689 026076 000032          032
690 026100 000010          8.
691 026102 004737 007266  JSR    PC,CKDATA     ;READ AND COMPARE 2ND CHAR TO 377
692 026106 000377          377
693 026110 000000          0
694 026112 004737 003276  15$: JSR    PC,MSTCLR  ;ISSUE MASTER CLEAR TO CLEAN UP
695 026116 026116          ENDTST
                                L10037: TRAP    C$ETST
026116 104401

```

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

```

:*****
:SBTTL      TEST 11 - SWITCH PACK PRINTOUT AND TEST
:*
:* - READ AND PRINT SWITCH PACK #1 :
:* THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
:* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
:* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
:* SWITCHES ARE IN BITS 1,2,3,5.
:*
:* - READ AND PRINT SWITCH PACK #2 :
:* THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
:* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
:* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
:* SWITCHES ARE IN BITS 0-7.
:*
:* - READ AND PRINT SWITCH PACK #3 :
:* THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
:* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
:* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
:* SWITCHES ARE IN BITS 0-7.

```

```

721
722 026120
026120
723
724
725
726 026120
026120
026120 104402
727 026122 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
728 026126 012737 000011 002364 MOV #11,REGNUM ;SET LU REG NO. = 11
729 026134 004737 003344 JSR PC,READLU ;READ LU REG 11
730 026140 142737 000321 002350 BICB #321,REDBYT ;MASK OFF NON-SWITCH BITS
731 026146 023727 002400 000001 CMP FRSPAS,#1 ;SEE IF IN FIRST PASS AFTER LOAD
732 026154 001403 BEQ 3$ ;BR IF YES
733 026156 005737 002260 TST PRNFLG ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
734 026162 001424 BEQ 4$ ;BR IF NOT
735 026164
736
737 026164 013746 002436
026170 012746 013034
026174 012746 000002
026200 010600
026202 104417
026204 062706 000006
738
739 026210
026210 013746 002350
026214 012746 012611
026220 012746 000002
026224 010600
026226 104417
026230 062706 000006
740 026234 005737 002262
741 026240 001420
742 026242 123737 002350 002454
743 026250 001414
744 026252 013737 002454 002370
745 026260 013737 002350 002372
746 026266 004737 004200
747
748 026272
026272 104455
026274 000053
026276 014070
026300 015360
749 026302
750 026302
026302
026302 104403
751
752
753
754 026304
026304
026304 104402
    ;*****
    BGNTST
    T11::
    -----
    ; READ AND PRINT SWITCH PACK #1, IF DESIRED
    -----
    BGNSUB
    T11.1: TRAP C$BSUB
    JSR PC,MSTCLR ;ISSUE MASTER CLEAR
    MOV #11,REGNUM ;SET LU REG NO. = 11
    JSR PC,READLU ;READ LU REG 11
    BICB #321,REDBYT ;MASK OFF NON-SWITCH BITS
    CMP FRSPAS,#1 ;SEE IF IN FIRST PASS AFTER LOAD
    BEQ 3$ ;BR IF YES
    TST PRNFLG ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
    BEQ 4$ ;BR IF NOT
    3$:
    ;PRINT DEVICE ADDRESS
    PRINTF #FMT18,MPCSR
    MOV MPCSR,-(SP)
    MOV #FMT18,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTF
    ADD #6,SP
    ;PRINT SWITCH PACK #1
    PRINTF #FMT12,REDBYT
    MOV REDBYT,-(SP)
    MOV #FMT12,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTF
    ADD #6,SP
    4$: TST SWIFLG ;SEE IF TEST IS ALLOWED
    BEQ 6$ ;BR IF NOT
    CMPB REDBYT,LUSWI1 ;COMPARE SWITCHES TO EXPECTED
    BEQ 6$ ;BR IF MATCH
    MOV LUSWI1,GOODAT ;SET EXPECTED DATA
    MOV REDBYT,BADDAT ;SET ACTUAL DATA
    JSR PC,GETALL ;GET REGS FOR PRINTOUT
    ;REPORT SWITCH PACK #1 INCORRECT
    ERRDF 43,EM43,ERR2
    TRAP C$ERDF
    .WORD 43
    .WORD EM43
    .WORD ERR2
    6$:
    ENDSUB
    L10041: TRAP C$ESUB
    -----
    ; READ AND PRINT SWITCH PACK #2, IF DESIRED
    -----
    BGNSUB
    T11.2: TRAP C$BSUB
    
```


TEST 11 - SWITCH PACK PRINTOUT AND TEST

```

755 026306 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
756 026312 012737 000015 002364 MOV #15,REGNUM ;SET LU REG NO. = 15
757 026320 004737 003344 JSR PC,READLU ;READ LU REG 15
758 026324 023727 002400 000001 CMP FRSPAS,#1 ;SEE IF IN FIRST PASS AFTER LOAD
759 026332 001403 BEQ 3$ ;BR IF YES
760 026334 005737 002260 TST PRNFLG ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
761 026340 001412 BEQ 4$ ;BR IF NOT
762 026342 3$: PRINTF #FMT13,REDBYT
763 026342 013746 002350 MOV REDBYT,-(SP)
026346 012746 012655 MOV #FMT13,-(SP)
026352 012746 000002 MOV #2,-(SP)
026356 010600 MOV SP,R0
026360 104417 TRAP C$PNTF
026362 062706 000006 ADD #6,SP
764 026366 005737 002262 4$: TST SWIFLG ;SEE IF TEST IS ALLOWED
765 026372 001420 BEQ 6$ ;BR IF NOT
766 026374 123737 002350 002456 CMPB REDBYT,LUSWI2 ;COMPARE SWITCHES TO EXPECTED
767 026402 001414 BEQ 6$ ;BR IF MATCH
768 026404 013737 002456 002370 MOV LUSWI2,GOODAT ;SET EXPECTED DATA
769 026412 013737 002350 002372 MOV REDBYT,BADDAT ;SET ACTUAL DATA
770 026420 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
771 ;REPORT SWITCH PACK #2 INCORRECT
772 026424 ERRDF 44,EM44,ERR2
773 026424 104455 TRAP C$ERDF
774 026426 000054 .WORD 44
026430 014115 .WORD EM44
026432 015360 .WORD ERR2
775 026434 6$: ENDSUB
776 026434 L10042: TRAP C$ESUB
777 026434 104403
778 026436 ;-----
026436 ; READ AND PRINT SWITCH PACK #3, IF DESIRED
026436 ;-----
026436 BGNSUB
026436 104402 T11.3: TRAP C$BSUB
779 026440 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
780 026444 012737 000016 002364 MOV #16,REGNUM ;SET LU REG NO. = 16
781 026452 004737 003344 JSR PC,READLU ;READ LU REG 16
782 026456 023727 002400 000001 CMP FRSPAS,#1 ;SEE IF IN FIRST PASS AFTER LOAD
783 026464 001403 BEQ 3$ ;BR IF YES
784 026466 005737 002260 TST PRNFLG ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
785 026472 001412 BEQ 4$ ;BR IF NOT
786 026474 3$: PRINTF #FMT14,REDBYT
787 026474 013746 002350 MOV REDBYT,-(SP)
026474 012746 012721 MOV #FMT14,-(SP)
026500 012746 000002 MOV #2,-(SP)
026504 010600 MOV SP,R0
026510 104417 TRAP C$PNTF
026512 062706 000006 ADD #6,SP
788 026520 005737 002262 4$: TST SWIFLG ;SEE IF TEST IS ALLOWED
789 026524 001420 BEQ 6$ ;BR IF NOT
790 026526 123737 002350 002460 CMPB REDBYT,LUSWI3 ;COMPARE SWITCHES TO EXPECTED
791 026534 001414 BEQ 6$ ;BR IF MATCH

```

```

792 026536 013737 002460 002370      MOV    LUSWI3,GOODAT ;SET EXPECTED DATA
793 026544 013737 002350 002372      MOV    REDBYT,BADDAT ;SET ACTUAL DATA
794 026552 004737 004200              JSR    PC,GETALL     ;GET REGS FOR PRINTOUT
795                                     ;REPORT SWITCH PACK #3 INCORRECT
796 026556                                ERRDF  45,EM45,ERR2
                                TRAP    C$ERDF
                                .WORD   45
                                .WORD   EM45
                                .WORD   ERR2
797 026566                                6$:
798 026566                                ENDSUB
                                L10043:
                                TRAP    C$ESUB
799 026570 104403                                ENDTST
                                L10040:
                                TRAP    C$ETST
800
801
802
803
804
805
806
807
808
809
810
811
812 026572                                ;*****
813 026572 004737 003276              SBTTL  TEST 12 - REG AX3-15 PRINTOUT
814 026576 142777 000010 153634      ;*
815 026604 012737 000006 002366      ;* IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
816 026612 004737 003576              ;* THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
817 026616 023727 002400 000001      ;* PRINT THE REG.
818 026624 001403                                ;*****
819 026626 005737 002260              BGNTST
820 026632 001424                                T12::
821 026634                                JSR    PC,MSTCLR    ;ISSUE MASTER CLEAR
822                                BICB  #LULoop,@BSEL1 ;CLEAR LULoop
823 026634 013746 002436              MOV    #6,AXNUM    ;SET AX BYTE NO. FOR AX3-15
824 026640 012746 013034              JSR    PC,READAX   ;READ AX3-15,AX3-16
825 026644 012746 000002              CMP    FRSPAS,#1   ;SEE IF FIRST PASS AFTER LOAD
826 026650 010600              BEQ    3$          ;BR IF NOT
827 026652 104417              TST   PRNFLG      ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
828 026654 062706 000006              BEQ    4$          ;BR IF NOT
                                3$:
                                ;PRINT DEVICE ADDRESS
                                PRINTF #FMT18,MPCSR
                                MOV    MPCSR,-(SP)
                                MOV    #FMT18,-(SP)
                                MOV    #2,-(SP)
                                MOV    SP,R0
                                TRAP  C$PNTF
                                ADD   #6,SP
829 026660                                ;PRINT AX3-15
830 026660 013746 002354              PRINTF #FMT15,RAX15
831 026664 012746 012765              MOV    RAX15,-(SP)
832 026670 012746 000002              MOV    #FMT15,-(SP)
833 026674 010600              MOV    #2,-(SP)
834 026676 104417              MOV    SP,R0
835 026700 062706 000006              TRAP  C$PNTF
836 026704                                ADD   #6,SP
837 026704                                4$:
                                ENDTST
    
```

026704
026704 104401

L10044:
TRAP C\$ETST

828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849

```

:*****
:SBTTL      TEST 13 - CRC GENERATION TEST
:*
:* - CRC-16, CHAR MODE:
:* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
:* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
:* TO CLOCK THE DATA.  AT THE END OF THE MESSAGE THE
:* PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
:*
:* - CRC-CCITT - 1'S PRESET:
:* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED.  AT
:* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
:*
:* - CRC-CCITT - 0'S PRESET:
:* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED.  AT
:* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
:*****
BGNTST

```

850 026706
026706

T13::

851
852
853

: CRC 16, CHAR MODE

854 026706 012737 027236 002346
855 026714 004737 003276
856 026720 004737 010276
857 026724 000226
858 026726 000011
859 026730 000000
860 026732 000000
861 026734 004737 010470
862 026740 002670
863 026742 000011
864 026744 004737 004726
865 026750 000136
866 026752 004737 007266
867 026756 000000
868 026760 000000
869 026762 004737 007266
870 026766 000125
871 026770 000000
872 026772 004737 007266
873 026776 000252
874 027000 000000
875 027002 004737 007266
876 027006 000377
877 027010 000000
878 027012 004737 007266
879 027016 100400
880 027020 000000
881

```

MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
JSR      PC,MSTCLR        ;ISSUE MASTER CLEAR
JSR      PC,SETUP         ;PROGRAM THE USYRT
SYNCH
STRIP!DDCMP
000
000
JSR      PC,LODMSG        ;LOAD MSG INTO TX SILO
MSG1
9.
JSR      PC,STPLU         ;CLOCK THE MSG
94.
JSR      PC,CKDATA        ;READ AND COMPARE CHAR TO 000
000
0
JSR      PC,CKDATA        ;READ AND COMPARE CHAR TO 125
125
0
JSR      PC,CKDATA        ;READ AND COMPARE CHAR TO 252
252
0
JSR      PC,CKDATA        ;READ AND COMPARE CHAR TO 377
377
0
JSR      PC,CKDATA        ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
CRCCHK!400
0

```

```

882
883
884
885 027022 004737 003276
886 027026 004737 010276
887 027032 000000
888 027034 000000
889 027036 000000
890 027040 000000
891 027042 004737 010470
892 027046 002670
893 027050 000011
894 027052 004737 004726
895 027056 000146
896 027060 004737 007266
897 027064 000000
898 027066 000000
899 027070 004737 007266
900 027074 000125
901 027076 000000
902 027100 004737 007266
903 027104 000252
904 027106 000000
905 027110 004737 007266
906 027114 000377
907 027116 000000
908 027120 004737 007266
909 027124 101000
910 027126 000000
911
912
913
914
915
916 027130 004737 003276
917 027134 004737 010276
918 027140 000000
919 027142 000100
920 027144 000000
921 027146 000000
922 027150 004737 010470
923 027154 002670
924 027156 000011
925 027160 004737 004726
926 027164 000146
927 027166 004737 007266
928 027172 000000
929 027174 000000
930 027176 004737 007266
931 027202 000125
932 027204 000000
933 027206 004737 007266
934 027212 000252
935 027214 000000
936 027216 004737 007266
937 027222 000377
938 027224 000000

```

```

: CRC-CCITT-1'S PRESET, BIT MODE
-----
      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
      JSR      PC,SETUP      ;PROGRAM THE USYRT
      000
      000
      000
      000
      JSR      PC,LODMSG      ;LOAD MSG INTO TX SILO
      MSG1
      9.
      JSR      PC,STPLU      ;CLOCK THE MSG
      102.
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 000
      000
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 125
      125
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 252
      252
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 377
      377
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
      CRCCHK!1000
      0

```

```

: CRC-CCITT-0'S PRESET, BIT MODE
-----
      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
      JSR      PC,SETUP      ;PROGRAM THE USYRT
      000
      CRC1
      000
      000
      JSR      PC,LODMSG      ;LOAD MSG INTO TX SILO
      MSG1
      9.
      JSR      PC,STPLU      ;CLOCK THE MSG
      102.
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 000
      000
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 125
      125
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 252
      252
      0
      JSR      PC,CKDATA      ;READ AND COMPARE CHAR TO 377
      377
      0

```

```

939 027226 004737 007266      JSR    PC,CKDATA      ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
940 027232 101000              CRCCHK!1000
941 027234 000000              0
942
943 027236 004737 003276      24$: JSR    PC,MSTCLR   ;ISSUE MASTER CLEAR TO CLEAN UP
944 027242 000000              ENDTST
                                L10045:
                                TRAP   C$ETST
027242 104401

```

945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968

```

:*****
:SBTTL      TEST 14 - CRC ERROR DETECTION TEST
:
:* - CRC-16, CHAR MODE :
:* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
:* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
:* TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
:* THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
:* ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
:* AN ERROR.
:
:* - CRC-CCITT - 1'S PRESET :
:* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
:* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
:
:* - CRC-CCITT - 0'S PRESET :
:* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
:* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
:*****
BGNTST

```

969 027244
027244

T14::

970
971
972

: CRC 16, CHAR MODE

```

973 027244 012737 027646 002346  MOV    #24$,RETADR   ;SET TEST EXIT ADRS FOR ERRORS
974 027252 004737 005212              JSR    PC,INITRN    ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
975 027256 000226              SYNCH
976 027260 000011              STRIP!DDCMP
977 027262 004737 005574              JSR    PC,TXCHAR    ;LOAD 000 CHAR, TX 1ST SYNCH
978 027266 000000              000
979 027270 100010              CHPCHK!8.
980 027272 004737 010470              JSR    PC,LODMSG    ;LOAD MSG INTO TX SILO
981 027276 002676              MSG1+6
982 027300 000006              6
983 027302 004737 004726              JSR    PC,STPLU     ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
984 027306 000010              8.
985 027310 004737 007164              JSR    PC,STPERR    ;MAKE 1ST BIT = 1 INSTEAD OF 0
986 027314 000011              STRIP!DDCMP
987 027316 000001              1
988 027320 004737 004726              JSR    PC,STPLU     ;CLOCK REST OF MESSAGE
989 027324 000122              82.
990 027326 004737 007266              JSR    PC,CKDATA    ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
991 027332 000001              001
992 027334 000000              0

```

```

993 027336 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 125
994 027342 000125 125
995 027344 000000 0
996 027346 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 252
997 027352 000252 252
998 027354 000000 0
999 027356 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 377
1000 027362 000377 377
1001 027364 000000 0
1002 027366 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
1003 027372 100000 CRCCHK!000
1004 027374 000000 0
1005
1006
1007
1008
    
```

 : CRC-CCITT-1'S PRESET, BIT MODE

```

1009 027376 004737 005212 JSR PC,INITRN ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
1010 027402 000000 000
1011 027404 000000 000
1012 027406 004737 005574 JSR PC,TXCHAR ;LOAD 000 CHAR, TX 1ST FLAG
1013 027412 000000 000
1014 027414 100010 CHPCHK!8.
1015 027416 004737 010470 JSR PC,LODMSG ;LOAD MSG INTO TX SILO
1016 027422 002676 MSG1+6
1017 027424 000006 6
1018 027426 004737 004726 JSR PC,STPLU ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
1019 027432 000010 8.
1020 027434 004737 007164 JSR PC,STPERR ;MAKE 1ST BIT = 1 INSTEAD OF 0
1021 027440 000000 000
1022 027442 000001 1
1023 027444 004737 004726 JSR PC,STPLU ;CLOCK REST OF MESSAGE
1024 027450 000122 82.
1025 027452 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
1026 027456 000001 001
1027 027460 000000 0
1028 027462 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 125
1029 027466 000125 125
1030 027470 000000 0
1031 027472 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 252
1032 027476 000252 252
1033 027500 000000 0
1034 027502 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 377
1035 027506 000377 377
1036 027510 000000 0
1037 027512 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
1038 027516 101400 CRCCHK!1400
1039 027520 000000 0
1040
1041
1042
1043
    
```

 : CRC-CCITT-0'S PRESET, BIT MODE

```

1044 027522 004737 005212 JSR PC,INITRN ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
1045 027526 000000 000
1046 027530 000100 CRC1
1047 027532 004737 005574 JSR PC,TXCHAR ;LOAD 000 CHAR, TX 1ST FLAG
1048 027536 000000 000
1049 027540 100010 CHPCHK!8.
    
```

```
1050 027542 004737 010470 JSR PC,LODMSG ;LOAD MSG INTO TX SILO
1051 027546 002676 MSG1+6
1052 027550 000006 6
1053 027552 004737 004726 JSR PC,STPLU ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
1054 027556 000010 8.
1055 027560 004737 007164 JSR PC,STPERR ;MAKE 1ST BIT = 1 INSTEAD OF 0
1056 027564 000100 CRC1
1057 027566 000001 1
1058 027570 004737 004726 JSR PC,STPLU ;CLOCK REST OF MESSAGE
1059 027574 000122 82.
1060 027576 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
1061 027602 000001 001
1062 027604 000000 0
1063 027606 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 125
1064 027612 000125 125
1065 027614 000000 0
1066 027616 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 252
1067 027622 000252 252
1068 027624 000000 0
1069 027626 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 377
1070 027632 000377 377
1071 027634 000000 0
1072 027636 004737 007266 JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
1073 027642 101400 CRCCHK!1400
1074 027644 000000 0
1075
1076 027646 004737 003276 24$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
1077 027652 ENDTST
027652 L10046: TRAP C$ETST
027652 104401
```

1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102 027654
027654
1103

```
*****
:SBTTL TEST 15 - VRC PARITY GENERATION TEST
:*
:* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
:* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
:* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
:* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
:* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
:* LAST 4 CHARS IT SHOULD = 0.
:*
:* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
:* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
:* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
:* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
:* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE
:* LAST 4 CHARS IT SHOULD = 1.
:*
:* DATA PATTERN Q = 000,120,125,137,040,052,057,177
:*****
:BGNTST
:-----
: T15::
```

```

1104 : TEST ODD VRC GENERATION
1105 -----
1106 027654 012737 000006 002366 MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
1107 027662 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1108 027670 BGNSUB
                                T15.1:
                                TRAP C$BSUB
1109 027670 104402
1109 027672 012737 030074 002346 MOV #8$,RETADR ;SET SUBTEST EXIT ADDRESS FOR ERRORS
1110 027700 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
1111 027704 000026 026
1112 027706 000111 CRC1!STRIP!DDCMP ;CHAR MODE, ODD VRC
1113 027710 004737 010546 JSR PC,LDBYTS ;LOAD DATA INTO TX SILO
1114 027714 002613 PATQ
1115 027716 000010 8.
1116 027720 004737 010720 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
1117 027724 001000 TXEOM
1118 027726 000002 2
1119 027730 005037 002360 CLR WAX15
1120 027734 012737 000347 002362 MOV #TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO,WAX16
1121 027742 004737 003764 JSR PC,WRITAX ;SET TX AND RCV LENGTHS = 7
1122 027746 004737 004726 JSR PC,STPLU ;CLOCK FIRST SYNCH
1123 027752 000010 8.
1124 027754 004737 004726 JSR PC,STPLU ;CLOCK 2ND SYNCH
1125 027760 000010 8.
1126 027762 005001 CLR R1 ;INIT CHAR COUNT
1127 027764 004737 004726 2$: JSR PC,STPLU ;CLOCK A CHAR
1128 027770 000010 8.
1129 027772 004737 003344 JSR PC,READLU ;READ REG 17
1130 027776 005201 INC R1 ;INCR CHAR COUNT
1131 030000 020127 000004 CMP R1,#4 ;SEE IF 4 CHARS CLKD YET
1132 030004 003014 BGT 4$ ;BR IF YES
1133 030006 132737 000040 002350 BITB #TXDATA,REDBYT ;SEE IF PARITY BIT IS SET
1134 030014 001024 BNE 6$ ;BR IF YES
1135 030016 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1136 ;REPORT ODD VRC PARITY BIT NOT SET
1137 ERRDF 48,EM48,ERR7
                                TRAP C$ERDF
                                .WORD 48
                                .WORD EM48
                                .WORD ERR7
1138 030032 ESCAPE SUB
                                TRAP C$ESCAPE
                                .WORD L10050-.
1139 030036 132737 000040 002350 4$: BITB #TXDATA,REDBYT ;SEE IF PARITY BIT IS CLEARED
1140 030044 001410 BEQ 6$ ;BR IF YES
1141 030046 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1142 ;REPORT ODD VRC PARITY BIT NOT CLEARED
1143 ERRDF 49,EM49,ERR7
                                TRAP C$ERDF
                                .WORD 49
                                .WORD EM49
                                .WORD ERR7
1144 030062 ESCAPE SUB
                                TRAP C$ESCAPE
                                .WORD L10050-.
1145 030066 020127 000010 6$: CMP R1,#8. ;SEE IF ALL CHARS TESTED YET
1146 030072 002734 BLT 2$ ;BR IF NOT
  
```



```

1147 030074
1148 030074
      030074
      030074 104403
1149
1150
1151
1152 030076 012737 000006 002366
1153 030104 012737 000017 002364
1154 030112
      030112
      030112 104402
1155 030114 012737 030316 002346
1156 030122 004737 005212
1157 030126 000026
1158 030130 000211
1159 030132 004737 010546
1160 030136 002613
1161 030140 000010
1162 030142 004737 010720
1163 030146 001000
1164 030150 000002
1165 030152 005037 002360
1166 030156 012737 000347 002362
1167 030164 004737 003764
1168 030170 004737 004726
1169 030174 000010
1170 030176 004737 004726
1171 030202 000010
1172 030204 005001
1173 030206 004737 004726
1174 030212 000010
1175 030214 004737 003344
1176 030220 005201
1177 030222 020127 000004
1178 030226 003014
1179 030230 132737 000040 002350
1180 030236 001424
1181 030240 004737 004200
1182
1183 030244
      030244 104455
      030246 000063
      030250 014403
      030252 020216
1184 030254
      030254 104410
      030256 000040
1185 030260 132737 000040 002350
1186 030266 001010
1187 030270 004737 004200
1188
1189 030274
      030274 104455
      030276 000062
      030300 014347
      030302 020216
8$:
      ENDSUB
      L10050: TRAP C$ESUB
-----
: TEST EVEN VRC GENERATION
-----
      MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
      MOV #17,REGNUM ;SET REG NO. = 17
      BGNSUB
      T15.2: TRAP C$BSUB
      MOV #18$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
      JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
      026
      CRC2!STRIP!DDCMP ;CHAR MODE, EVEN VRC
      JSR PC,LDBYTS ;LOAD DATA INTO TX SILO
      PATQ
      8.
      JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
      TXEOM
      2
      CLR WAX15
      MOV #TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO,WAX16
      JSR PC,WRITAX ;SET TX AND RCV LENGTHS = 7
      JSR PC,STPLU ;CLOCK FIRST SYNCH
      8.
      JSR PC,STPLU ;CLOCK 2ND SYNCH
      8.
      CLR R1 ;INIT CHAR COUNT
      JSR PC,STPLU ;CLOCK A CHAR
12$:
      8.
      JSR PC,READLU ;READ REG 17
      INC R1 ;INCR CHAR COUNT
      CMP R1,#4 ;SEE IF 4 CHARS CLKD YET
      BGT 14$ ;BR IF YES
      BITB #TXDATA,REDBYT ;SEE IF PARITY BIT IS CLEARED
      BEQ 16$ ;BR IF YES
      JSR PC,GETALL ;GET REGS FOR PRINTOUT
      ;REPORT EVEN VRC PARITY BIT NOT CLEARED
      ERRDF 51,EM51,ERR7
      TRAP C$ERDF
      .WORD 51
      .WORD EM51
      .WORD ERR7
      ESCAPE SUB
      TRAP C$ESCAPE
      .WORD L10051-.
14$:
      BITB #TXDATA,REDBYT ;SEE IF PARITY BIT IS SET
      BNE 16$ ;BR IF YES
      JSR PC,GETALL ;GET REGS FOR PRINTOUT
      ;REPORT EVEN VRC PARITY BIT NOT SET
      ERRDF 50,EM50,ERR7
      TRAP C$ERDF
      .WORD 50
      .WORD EM50
      .WORD ERR7
  
```

```

1190 030304          ESCAPE SUB
      030304 104410
      030306 000010
1191 030310 020127 000010 16$: CMP R1,#8. ;SEE IF ALL CHARS TESTED YET
1192 030314 002734          BLT 12$ ;BR IF NOT
1193 030316          18$:
1194 030316          ENDSUB
      030316          L10051:
      030316 104403          TRAP C$ESUB
1195 030320 004737 003276          JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
1196 030324          ENDTST
      030324          L10047:
      030324 104401          TRAP C$ETST
    
```

1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223

```

:*****
:SBTTL      TEST 16 - VRC ERROR DETECTION TEST
:
:SUBTEST 1 - FORCING OF BCC USING ODD VRC
:* THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.
:* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
:* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
:* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
:* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
:* RECEIVED (INDICATING AN ERROR).
:
:SUBTEST 2 - FORCING OF BCC USING EVEN VRC
:* THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.
:* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
:* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
:* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
:* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
:* RECEIVED (INDICATING AN ERROR).
:
:* DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
:*                   172,176.
:*****
    
```

```

1224 030326
      030326
1225
1226
1227
1228 030326 012737 000006 002366
1229 030334 012737 000012 002364
1230 030342
      030342
      030342 104402
1231 030344 012737 030562 002346
1232 030352 004737 005212
1233 030356 000026
1234 030360 000111
1235 030362 004737 010720
1236 030366 000400
1237 030370 000001
    
```

```

BGNTST
:-----
:TEST ODD VRC ERROR DETECTION
:-----
      MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
      MOV #12,REGNUM ;SET REG NO.
      BGNSUB
:
:
      T16.1:
      TRAP C$BSUB
      MOV #10$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
      JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
      O26
      CRC1!STRIP!DDCMP ;CHAR MODE, ODD VRC
      JSR PC,LODSIL ;LOAD A THIRD SOM INTO TX SILO
      TXSOM
      1
    
```

```

1238 030372 004737 010546 JSR PC,LDBYTS ;LOAD DATA INTO TX BUFFER
1239 030376 002623 PATR
1240 030400 000017 15.
1241 030402 004737 010720 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
1242 030406 001000 TXEOM
1243 030410 000002 2
1244 030412 005037 002360 CLR WAX15
1245 030416 012737 000347 002362 MOV #TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO,WAX16
1246 030424 013737 002362 002424 MOV WAX16,SAVLEN ;STORE LENGTH 7
1247 030432 004737 003764 JSR PC,WRITAX ;SET TX AND RCV LENGTHS = 7
1248 030436 004737 004726 JSR PC,STPLU ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
1249 030442 000130 88.
1250 030444 012701 000007 MOV #7,R1 ;INIT COUNTER FOR LAST 7 CHARS
1251 030450 004737 007164 3$: JSR PC,STPERR ;ASSERT IERR BIT FOR 1 TIME
1252 030454 000111 CRC1!STRIP!DDCMP
1253 030456 000001 1
1254 030460 004737 004726 JSR PC,STPLU ;CLOCK REST OF CHAR
1255 030464 000007 7
1256 030466 005301 DEC R1 ;DECR COUNTER
1257 030470 001367 BNE 3$ ;BR IF NOT DONE TRANSMITTING YET
1258 030472 004737 004726 JSR PC,STPLU ;CLOCK 2 TERMINATING SYNCHS
1259 030476 000020 16.
1260 030500 012701 000010 MOV #8,R1 ;INIT COUNTER FOR ERROR-FREE CHARS
1261 030504 012703 002623 MOV #PATR,R3 ;INIT DATA PATTERN POINTER
1262 030510 112337 030520 5$: MOV (R3)+,6$ ;GET AN EXPECTED DATA CHAR
1263 030514 004737 007266 JSR PC,CKDATA ;GO CHECK CHAR, CHK BCC=0
1264 030520 100000 6$: BCCCHK!000
1265 030522 000000 0
1266 030524 005301 DEC R1 ;DECR COUNTER
1267 030526 001370 BNE 5$ ;BR IF NOT DONE YET
1268 030530 012701 000007 MOV #7,R1 ;INIT COUNTER FOR ERROR CHARS
1269 030534 112337 030552 8$: MOV (R3)+,9$ ;GET EXPECTED DATA CHAR
1270 030540 052737 000001 030552 BIS #BIT0,9$ ;EXPECT ERROR BIT 0 SET
1271 030546 004737 007266 JSR PC,CKDATA ;CHECK DATA, CHK BCC=1
1272 030552 100400 9$: BCCCHK!RXBCC!000
1273 030554 000000 0
1274 030556 005301 DEC R1 ;DECR COUNTER
1275 030560 001365 BNE 8$ ;BR IF NOT DONE YET
1276 030562 10$:
1277 030562 ENDSUB
030562 L10053: TRAP C$ESUB
030562 104403
-----
: TEST EVEN VRC ERROR DETECTION
-----
1281 030564 012737 000006 002366 MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
1282 030572 012737 000012 002364 MOV #12,REGNUM ;SET REG NO.
1283 030600 BGNSUB
030600 T16.2: TRAP C$BSUB
030600 104402
1284 030602 012737 031020 002346 MOV #30$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
1285 030610 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
1286 030614 000026 026
1287 030616 000211 CRC2!STRIP!DDCMP ;CHAR MODE, EVEN VRC
1288 030620 004737 010720 JSR PC,LODSIL ;LOAD A THIRD SOM INTO TX SILO
1289 030624 000400 TXSOM
1290 030626 000001 1
    
```

```

1291 030630 004737 010546 JSR PC,LDBYTS ;LOAD DATA INTO TX BUFFER
1292 030634 002623 PATR
1293 030636 000017 15.
1294 030640 004737 010720 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
1295 030644 001000 TXEOM
1296 030646 000002 2
1297 030650 005037 002360 CLR WAX15
1298 030654 012737 000347 002362 MOV #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
1299 030662 013737 002362 002424 MOV WAX16,SAVLEN ;STORE LENGTH 7
1300 030670 004737 003764 JSR PC,WRITAX ;SET TX AND RCV LENGTHS = 7
1301 030674 004737 004726 JSR PC,STPLU ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
1302 030700 000130 88.
1303 030702 012701 000007 MOV #7,R1 ;INIT COUNTER FOR LAST 7 CHARS
1304 030706 004737 007164 23$: JSR PC,STPERR ;ASSERT IERR BIT FOR 1 TIME
1305 030712 000211 CRC2!STRIP!DDCMP
1306 030714 000001 1
1307 030716 004737 004726 JSR PC,STPLU ;CLOCK REST OF CHAR
1308 030722 000007 7
1309 030724 005301 DEC R1 ;DECR COUNTER
1310 030726 001367 BNE 23$ ;BR IF NOT DONE TRANSMITTING YET
1311 030730 004737 004726 JSR PC,STPLU ;CLOCK 2 TERMINATING SYNCHS
1312 030734 000020 16.
1313 030736 012701 000010 MOV #8,R1 ;INIT COUNTER FOR ERROR-FREE CHARS
1314 030742 012703 002623 MOV #PATR,R3 ;INIT DATA PATTERN POINTER
1315 030746 112337 030756 25$: MOVB (R3)+,26$ ;GET EXPECTED DATA CHAR
1316 030752 004737 007266 JSR PC,CKDATA ;CHK DATA, CHECK BCC=0
1317 030756 100000 26$: BCCCHK!000
1318 030760 000000 0
1319 030762 005301 DEC R1 ;DECR COUNTER
1320 030764 001370 BNE 25$ ;BR IF NOT DONE YET
1321 030766 012701 000007 MOV #7,R1 ;INIT COUNTER FOR ERROR CHARS
1322 030772 112337 031010 28$: MOVB (R3)+,29$ ;GET EXPECTED DATA CHAR
1323 030776 052737 000001 031010 BIS #BIT0,29$ ;SET EXPECTED ERROR BIT 0
1324 031004 004737 007266 JSR PC,CKDATA ;CHK DATA, CHK BCC=1
1325 031010 100400 29$: BCCCHK!RXBCC!000
1326 031012 000000 0
1327 031014 005301 DEC R1 ;DECR COUNTER
1328 031016 001365 BNE 28$ ;BR IF NOT DONE YET
1329 031020 30$:
1330 031020 ENDSUB
1331 031022 104403 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP TRAP C$ESUB
1332 031026 031026 031026 104401 30$: ENDTST TRAP C$SETST
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343

```

```

:*****
:SBTTL TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC
:*
:* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A
:* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
:* ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF

```

```

1344      : * 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
1345      : * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
1346      : * SKIPPED FOR THAT UNIT.
1347      : *****
1348      BGNTST
1349      031030      012737      000021      002434      MOV      #17, TSTNUM      ;SET TEST NO.
1350      031030      012737      031310      002346      MOV      #24$, RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
1351      031044      004737      003276      JSR      PC, MSTCLR      ;ISSUE MASTER CLEAR
1352      031050      004737      010770      JSR      PC, CKLPBK      ;CHECK LOOPBACK -
1353      031054      000010      INTGRL      ; SEE IF TEST SHOULD BE RUN
1354      031056      012737      000323      031074      MOV      #I422!XYZ!V35!OP!TEST, 6$ ;SET UP TO SELECT INTEGRAL MODEM
1355      031064      004737      010276      JSR      PC, SETUP      ;PROGRAM THE USYRT
1356      031070      000226      SYNCH
1357      031072      000011      STRIP!DDCMP
1358      031074      000000      6$:      .WORD      0
1359      031076      000000      000
1360      031100      004737      010630      JSR      PC, LDMSG1      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1361      031104      142777      000010      151326      BICB     #LULOOK, @BSEL1   ;CLEAR LULOOK, CLOCK MSG
1362      031112      012737      000012      002364      MOV      #12, REGNUM     ;SET LU REG NO. = 12
1363      031120      012703      002762      MOV      #RCVBUF, R3     ;GET POINTER TO RCV MSG BUF
1364      031124      013702      002264      9$:      MOV      TCOUNT, R2     ;INIT TIMER
1365      031130      004737      003344      10$:     JSR      PC, READLU      ;READ REG 12
1366      031134      132737      000020      002350      BITB     #IRDY, REDBYT   ;SEE IF IRDY IS SET YET
1367      031142      001011      BNE      12$            ;BR IF YES
1368      031144      005202      INC      R2              ;INCREMENT TIMER
1369      031146      001370      BNE      10$            ;BR IF NO TIME-OUT YET
1370      031150      004737      004200      JSR      PC, GETALL     ;GET REGS FOR PRINTOUT
1371      ;REPORT IRDY NOT SET
1372      031154      104455      ERRDF    17, EM17, ERR7
1373      031154      000021      TRAP     C$ERRDF
1374      031156      013430      .WORD    17
1375      031160      020216      .WORD    EM17
1376      031162      000451      .WORD    ERR7
1377      031164      012337      031176      12$:     BR      24$            ;ESCAPE TO END OF TEST
1378      031166      004737      007266      MOV      (R3)+, 16$
1379      031172      000000      JSR      PC, CKDATA     ;COMPARE RCV'D DATA CHAR TO EXPECTED
1380      031176      000000      16$:     0
1381      031200      000000      0
1382      031202      020327      003000      CMP      R3, #RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1383      031206      103746      BLO      9$             ;BR IF NOT YET
1384      031210      004737      004620      JSR      PC, WAIT50     ;STALL FOR 50 MICRO-SEC
1385      031214      004737      005024      JSR      PC, OACTIV     ;CHECK OACT = 0
1386      031220      000000      0
1387      031222      004737      006232      JSR      PC, IACTIV     ;CHECK IACT STILL = 1
1388      031226      000001      1
1389      031230      012737      000013      002364      MOV      #13, REGNUM     ;SET REG NO. = 13
1390      031236      004737      003344      JSR      PC, READLU     ;READ REG 13
1391      031242      042737      000232      002350      BIC      #RING!HDX!MODR!STBY, REDBYT ;CLR UNUSED BITS
1392      031250      023727      002350      000000      CMP      REDBYT, #0      ;CHECK REG 13 FOR 000 (MODEM SIGNALS SHOULD BE CLEARED)
1393      031256      001414      BEQ      24$            ;BR IF CLEARED
1394      031260      012737      000000      002370      MOV      #0, GOODAT     ;SET EXPECTED DATA = 0
1395      031266      013737      002350      002372      MOV      REDBYT, BADDAT ;SET ACTUAL DATA
1396      031274      004737      004200      JSR      PC, GETALL     ;GET REGS FOR PRINTOUT
1397      ;REPORT REG MISCMPARE
1398      031300      104455      ERRDF    3, EM3, ERR3
1399      031300      TRAP     C$ERRDF
    
```

031302 000003
 031304 013257
 031306 015666
 1395 031310
 1396 031310
 031310
 031310 104401

.WORD 3
 .WORD EM3
 .WORD ERR3

24\$:
 ENDTST

L10055:
 TRAP C\$ETST

1397
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412

```

:*****
:SBTTL      TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC
:
:* THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
:* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
:* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
:* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
:* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
:* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
:* SKIPPED FOR THAT UNIT.
:*****
    
```

1413 031312

BGNTST

T18::

031312
 1414 031312 012737 000022 002434
 1415 031320 012737 031562 002346
 1416 031326 004737 003276
 1417 031332 004737 010770
 1418 031336 000020
 1419 031340 012737 000313 031356
 1420 031346 004737 010276
 1421 031352 000226
 1422 031354 000011
 1423 031356 000000
 1424 031360 000000
 1425 031362 142777 000010 151050
 1426 031370 012737 000013 002364
 1427 031376 004737 003344
 1428 031402 132737 000001 002350
 1429 031410 001415
 1430 031412 012737 000000 002370
 1431 031420 013737 002350 002372
 1432 031426 004737 004200

```

MOV #18, TSTNUM ;SET TEST NO.
MOV #24$, RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
JSR PC, MSTCLR ;ISSUE MASTER CLEAR
JSR PC, CKLPBK ;CHECK LOOPBACK -
V35 ; SEE IF TEST SHOULD BE RUN
MOV #I422!XYZ!INTGRL!OP!TEST, 6$ ;SET UP TO SELECT V35
JSR PC, SETUP ;PROGRAM THE USYRT
SYNCH
STRIP!DDCMP
6$: .WORD 0
000
BICB #LULOOP, @BSEL1 ;CLEAR LULOOP
MOV #13, REGNUM ;SET LU REG NO. = 13
JSR PC, READLU ;READ REG 13
BITB #CARR, REDBYT ;CHECK FOR CARRIER FALSELY SET
BEQ 8$ ;BR IF NOT SET
MOV #000, GOODAT ;SET EXPECTED DATA
MOV REDBYT, BADDAT ;SET ACTUAL DATA
JSR PC, GETALL ;GET REGS FOR PRINTOUT
:REPORT CARRIER NOT CLEARED
ERRDF 66, EM66, ERR7
    
```

TRAP C\$ERDF
 .WORD 66
 .WORD EM66
 .WORD ERR7

1433
 1434 031432
 031432 104455
 031434 000102
 031436 014772
 031440 020216
 1435 031442 000447
 1436 031444 152777 000010 150766
 1437 031452 004737 010630
 1438 031456 142777 000010 150754
 1439 031464 012737 000012 002364
 1440 031472 012703 002762
 1441 031476 013702 002264

```

8$: BR 24$
BISB #LULOOP, @BSEL1 ;SET LULOOP AGAIN
JSR PC, LDMSG1 ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
BICB #LULOOP, @BSEL1 ;CLEAR LULOOP, CLOCK MSG
MOV #12, REGNUM ;SET LU REG NO. = 12
MOV #RCVBUF, R3 ;GET POINTER TO RCV MSG BUF
9$: MOV TCOUNT, R2 ;INIT TIMER
    
```

```

1442 031502 004737 003344 10$: JSR PC,READLU ;READ REG 12
1443 031506 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1444 031514 001011 BNE 12$ ;BR IF YES
1445 031516 005202 INC R2 ;INCREMENT TIMER
1446 031520 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1447 031522 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1448 ;REPORT IRDY NOT SET
1449 031526 ERRDF 17,EM17,ERR7
031526 104455 TRAP C$ERDF
031530 000021 .WORD 17
031532 013430 .WORD EM17
031534 020216 .WORD ERR7
1450 031536 000411 BR 24$ ;ESCAPE TO END OF TEST
1451 031540 012337 031550 12$: MOV (R3)+,16$
1452 031544 004737 007266 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1453 031550 000000 16$: 0
1454 031552 000000 0
1455 031554 020327 003000 CMP R3,#RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1456 031560 103746 BLO 9$ ;BR IF NOT YET
1457 031562 24$:
1458 031562 ENDTST
031562
031562 104401 L10056: TRAP C$ETST
    
```

1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474

```

:*****
:SBTTL TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC
:*
:* THE RS232C & RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
:* AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
:* AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
:* OR A MODEM TEST MODE. THE MESSAGE CONSISTS
:* OF 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
:* P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
:* PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
:*****
    
```

```

1475 031564 BGNTST
031564 T19::
1476 031564 012737 000023 002434 MOV #19,,TSTNUM ;SET TEST NO.
1477 031572 012737 031744 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1478 031600 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1479 031604 004737 010770 JSR PC,CKLPBK ;CHECK LOOPBACK -
1480 031610 000100 XYZ ; SEE IF TEST SHOULD BE RUN
1481 031612 012737 000233 031630 MOV #I422!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT XYZ
1482 031620 004737 010276 JSR PC,SETUP ;PROGRAM THE USYRT
1483 031624 000226 SYNCH
1484 031626 000011 STRIP!DDCMP
1485 031630 000000 6$: .WORD 0
1486 031632 000000 000
1487 031634 004737 010630 JSR PC,LDMSG1 ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1488 031640 142777 000010 150572 BICB #LULOOP,@BSEL1 ;CLEAR LULOOP, CLOCK MSG
1489 031646 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
1490 031654 012703 002762 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
1491 031660 013702 002264 9$: MOV TCOUNT,R2 ;INIT TIMER
    
```

```

1492 031664 004737 003344 10$: JSR PC,READLU ;READ REG 12
1493 031670 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1494 031676 001011 BNE 12$ ;BR IF YES
1495 031700 005202 INC R2 ;INCREMENT TIMER
1496 031702 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1497 031704 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1498 ;REPORT IRDY NOT SET
1499 031710 ERRDF 17,EM17,ERR7
031710 104455 TRAP C$ERDF
031712 000021 .WORD 17
031714 013430 .WORD EM17
031716 020216 .WORD ERR7
1500 031720 000411 BR 24$ ;ESCAPE TO END OF TEST
1501 031722 012337 031732 12$: MOV (R3)+,16$
1502 031726 004737 007266 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1503 031732 000000 16$: 0
1504 031734 000000 0
1505 031736 020327 003000 CMP R3,#RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1506 031742 103746 BLO 9$ ;BR IF NOT YET
1507 031744 24$:
1508 031744 ENDTST
031744 L10057:
031744 104401 TRAP C$ETST
    
```

1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524

```

:*****
:SBTTL TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC
:*
:* THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
:* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
:* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
:* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
:* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
:* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
:* SKIPPED FOR THAT UNIT.
:*****
BGNTST
    
```

```

1525 031746 T20::
031746
1526 031746 012737 000024 002434 MOV #20,,TSTNUM ;SET TEST NO.
1527 031754 012737 032126 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1528 031762 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1529 031766 004737 010770 JSR PC,CKLPBK ;CHECK LOOPBACK -
1530 031772 000200 I422 ; SEE IF TEST SHOULD BE RUN
1531 031774 012737 000133 032012 MOV #XYZ!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT 422
1532 032002 004737 010276 JSR PC,SETUP ;PROGRAM THE USYRT
1533 032006 000226 SYNCH
1534 032010 000011 STRIP!DDCMP
1535 032012 000000 6$: .WORD 0
1536 032014 000000 000
1537 032016 004737 010630 JSR PC,LDMSG1 ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1538 032022 142777 000010 150410 BICB #LULOOP,@BSEL1 ;CLEAR LULOOP, CLOCK MSG
1539 032030 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
1540 032036 012703 002762 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
1541 032042 013702 002264 9$: MOV TCOUNT,R2 ;INIT TIMER
    
```



```

1542 032046 004737 003344 10$: JSR PC,READLU ;READ REG 12
1543 032052 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1544 032060 001011 BNE 12$ ;BR IF YES
1545 032062 005202 INC R2 ;INCREMENT TIMER
1546 032064 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1547 032066 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1548 ;REPORT IRDY NOT SET
1549 032072 ERRDF 17,EM17,ERR7
                                TRAP C$ERDF
                                .WORD 17
                                .WORD EM17
                                .WORD ERR7
1550 032102 000411 BR 24$ ;ESCAPE TO END OF TEST
1551 032104 012337 032114 12$: MOV (R3)+,16$
1552 032110 004737 007266 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1553 032114 000000 16$: 0
1554 032116 000000 0
1555 032120 020327 003000 CMP R3,#RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1556 032124 103746 BLO 9$ ;BR IF NOT YET
1557 032126 24$:
1558 032126 ENDTST
                                L10060:
                                TRAP C$ETST
1559 032126 104401
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576 032130
                                *****
                                .SBTTL TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST
                                *
                                * THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 DOES NOT INHIBIT
                                * LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
                                * A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
                                * BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
                                * CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
                                * THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
                                * BE UNLOADED (ALL CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
                                * IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.
                                *****
                                BGNTST
1577 032130 012737 032220 002346 MOV #24$,RETADR ;SET TEST EXIT ADRS FOR ERRORS
1578 032136 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1579 032144 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
1580 032150 000226 SYNCH
1581 032152 000011 STRIP!DDCMP
1582 032154 112737 000020 002352 MOVB #HDX,WRIBYT
1583 032162 004737 003422 JSR PC,WRITLU ;SET HDX BIT IN REG 13
1584 032166 004737 010470 JSR PC,LODMSG ;LOAD MSG INTO TX SILO
1585 032172 002674 MSG1+4
1586 032174 000007 7
1587 032176 004737 004726 JSR PC,STPLU ;CLK MORE THAN ENTIRE MSG
1588 032202 000136 94.
1589 032204 004737 004334 JSR PC,OSIRDY ;CHK ORDY = 1, OCOR = 0
1590 032210 000001 1
1591 032212 004737 005746 JSR PC,ISIRDY ;CHK ICIR = 1, IRDY = 0
  
```



```

032540 104410
032542 000170
1694 032544 142777 000010 147666 6$: BICB #LULOO, @BSEL1 ;CLEAR LULOO
1695 032552 004737 003344 JSR PC, READLU ;READ REG 13
1696 032556 023727 002350 000000 CMP REDBYT, #0 ;CHECK FOR INITIALIZED STATE
1697 032564 001416 BEQ 8$ ;BR IF OK
1698 032566 012737 000000 002370 MOV #0, GOODAT ;GET EXPECTED DATA
1699 032574 013737 002350 002372 MOV REDBYT, BADDAT ;GET ACTUAL DATA
1700 032602 004737 004200 JSR PC, GETALL ;GET REGS FOR PRINTOUT
1701 ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
1702 ERRDF 2, EM2, ERR2
032606 104455
032610 000002
032612 013220
032614 015360
1703 032616 ESCAPE SUB
032616 104410
032620 000112
1704 032622 005037 002352 8$: CLR WRIBYT ;SET DATA = 0 TO BE WRITTEN
1705 032626 004737 003422 JSR PC, WRITLU ;LOAD 0'S INTO REG 13
1706 032632 004737 003344 JSR PC, READLU ;READ REG 13
1707 032636 023727 002350 000000 CMP REDBYT, #000 ;CHECK FOR REG 13 CLEARED
1708 032644 001407 BEQ 9$ ;BR IF CLEARED
1709 032646 012737 000000 002370 MOV #000, GOODAT ;SET EXPECTED DATA
1710 032654 013737 002350 002372 MOV REDBYT, BADDAT ;SET ACTUAL DATA
1711 032662 000720 BR 3$ ;GO PRINT ERROR
1712 032664 012737 000017 002364 9$: MOV #17, REGNUM ;SET REG NO. = 17
1713 032672 004737 003344 JSR PC, READLU ;READ REG 17
1714 032676 042737 000002 002350 BIC #MCLK, REDBYT ;IGNORE MCLK BIT
1715 032704 123727 002350 000051 CMPB REDBYT, #TXDATA!ICIR!DDCMP ;CHK REG 17 FOR INIT'D STATE
1716 032712 001407 BEQ 10$ ;BR IF REG 17 INITIALIZED
1717 032714 012737 000051 002370 MOV #TXDATA!ICIR!DDCMP, GOODAT ;SET EXPECTED DATA
1718 032722 013737 002350 002372 MOV REDBYT, BADDAT ;SET ACTUAL DATA
1719 032730 000675 BR 3$ ;GO REPORT ERROR
1720 032732
1721 032732
032732 104403
1722
1723
1724
1725
1726 032734
032734
032734 104402
1727 032736 004737 003276 JSR PC, MSTCLR ;ISSUE MASTER CLEAR
1728 032742 105077 147472 CLRB @BSEL1 ;CLEAR LULOO
1729 032746 112777 000200 147464 MOVB #RUN, @BSEL1 ;SET RUN BIT IN BSEL1
1730 032754 112777 000010 147456 MOVB #LULOO, @BSEL1 ;CLEAR RUN, SET LULOO
1731 032762 012737 000013 002364 MOV #13, REGNUM ;SET REG NO. = 13
1732 032770 004737 003344 JSR PC, READLU ;READ REG 13
1733 032774 132737 000200 002350 BITB #RING, REDBYT ;SEE IF RING = 1
1734 033002 001010 BNE 9$ ;BR IF RING = 1
1735 033004 004737 004200 JSR PC, GETALL ;GET REGS FOR PRINTOUT
1736 ;REPORT RING NOT SET
1737 ERRDF 56, EM56, ERR7
033010 104455
    
```

TRAP C\$ESCAPE
.WORD L10064-

TRAP C\$ERDF
.WORD 2
.WORD EM2
.WORD ERR2

TRAP C\$ESCAPE
.WORD L10064-

L10064: TRAP C\$ESUB

 : SET RUN IN BSEL1, CHECK FOR RING SET IN REG 13

T23.2: TRAP C\$BSUB

TRAP C\$ERDF

```

033012 000070 .WORD 56
033014 014571 .WORD EM56
033016 020216 .WORD ERR7
1738 033020 ESCAPE SUB
033020 104410 TRAP C$ESCAPE
033022 000002 .WORD L10065-.
1739 033024 9$:
1740 033024 ENDSUB
033024 L10065:
033024 104403 TRAP C$ESUB
1741
1742
1743
1744
1745 033026
033026 BGNSUB
033026 104402 T23.3:
1746 033030 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR TRAP C$BSUB
1747 033034 112737 000200 002352 MOVB #POLL,WRIBYT
1748 033042 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1749 033050 004737 003422 JSR PC,WRITLU ;SET POLL IN REG 13
1750 033054 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1751 033062 004737 003344 JSR PC,READLU ;READ REG 17
1752 033066 132737 000100 002350 BITB #SIGQ,REDBYT ;SEE IF SIGQ = 1
1753 033074 001006 BNE 6$ ;BR IF SIGQ = 1
1754 033076 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1755 ;REPORT SIGQ NOT SET
1756 033102 ERRDF 63,EM63,ERR7
033102 104455 TRAP C$ERDF
033104 000077 .WORD 63
033106 014720 .WORD EM63
033110 020216 .WORD ERR7
1757 033112 6$:
1758 033112 ENDSUB
033112 L10066:
033112 104403 TRAP C$ESUB
1759
1760
1761
1762
1763 033114
033114 BGNSUB
033114 104402 T23.4:
1764 033116 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR TRAP C$BSUB
1765 033122 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
1766 033130 112737 000100 002352 MOVB #BPOLL,WRIBYT ;SET BPOLL IN LU REG 12
1767 033136 004737 003422 JSR PC,WRITLU
1768 033142 ENDSUB
033142 L10067:
033142 104403 TRAP C$ESUB
1769
1770
1771
1772
1773 033144
033144 BGNSUB
033144 104402 T23.5:
; SET DTR IN REG 13, CHECK FOR DTR AND MODR SET IN REG 13
TRAP C$BSUB

```

```

1774 033146 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1775 033152 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1776 033160 112737 000104 002352 MOVB #DTR!MAINT2,WRIBYT
1777 033166 004737 003422 JSR PC,WRITLU ;SET DTR IN REG 13
1778 ; (ALSO SET MAINT2 FOR MANUFACT. TEST CONN.)
1779 033172 142777 000010 147240 BICB #LULoop,@BSEL1 ;CLEAR LULoop
1780 033200 004737 003344 JSR PC,READLU ;READ REG 13
1781 033204 132737 000100 002350 BITB #DTR,REDBYT ;SEE IF DTR = 1
1782 033212 001010 BNE 6$ ;BR IF DTR = 1
1783 033214 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1784 ;REPORT DTR NOT SET
1785 033220 ERRDF 55,EM55,ERR7
033220 104455 TRAP C$ERDF
033222 000067 .WORD 55
033224 014555 .WORD EM55
033226 020216 .WORD ERR7
1786 033230 ESCAPE SUB
033230 104410 TRAP C$ESCAPE
033232 000026 .WORD L10070-.
1787 033234 132737 000010 002350 6$: BITB #MODR,REDBYT ;SEE IF MODR = 1
1788 033242 001006 BNE 12$ ;BR IF MODR = 1
1789 033244 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1790 ;REPORT MODR NOT SET
1791 033250 ERRDF 57,EM57,ERR7
033250 104455 TRAP C$ERDF
033252 000071 .WORD 57
033254 014606 .WORD EM57
033256 020216 .WORD ERR7
1792 033260 12$:
1793 033260 ENDSUB
033260 L10070: TRAP C$ESUB
033260 104403
1794
1795
1796
1797
1798 033262
033262 BGNSUB
033262 104402 T23.6: TRAP C$BSUB
1799 033264 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1800 033270 112737 000040 002352 MOVB #SELFR,WRIBYT
1801 033276 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1802 033304 004737 003422 JSR PC,WRITLU ;SET SELFR IN REG 13
1803 033310 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1804 033316 004737 003344 JSR PC,READLU ;READ REG 17
1805 033322 132737 000200 002350 BITB #SIGR,REDBYT ;SEE IF SIGR = 1
1806 033330 001006 BNE 6$ ;BR IF SIGR = 1
1807 033332 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1808 ;REPORT SIGR NOT SET
1809 033336 ERRDF 64,EM64,ERR7
033336 104455 TRAP C$ERDF
033340 000100 .WORD 64
033342 014735 .WORD EM64
033344 020216 .WORD ERR7
1810 033346 6$:
1811 033346 ENDSUB
033346 L10071:
    
```

```

1812 033346 104403 TRAP C$ESUB
1813
1814 -----
1815 : SET HDX IN REG 13, CHK FOR HDX SET IN REG 13
1816 -----
1816 033350 BGNSUB
1816 033350 T23.7: TRAP C$BSUB
1816 033350 104402
1817 033352 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1818 033356 112737 000020 002352 MOVB #HDX,WRIBYT
1819 033364 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1820 033372 004737 003422 JSR PC,WRITLU ;SET HDX IN REG 13
1821 033376 004737 003344 JSR PC,READLU ;READ REG 13
1822 033402 132737 000020 002350 BITB #HDX,REDBYT ;SEE IF HDX = 1
1823 033410 001006 BNE 6$ ;BR IF HDX = 1
1824 033412 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1825 :REPORT HDX NOT SET
1826 033416 ERRDF 58,EM58,ERR7
1826 033416 104455 TRAP C$ERDF
1826 033420 000072 .WORD 58
1826 033422 014623 .WORD EM58
1826 033424 020216 .WORD ERR7
1827 033426 6$:
1828 033426 ENDSUB
1828 033426 L10072: TRAP C$ESUB
1828 033426 104403
1829
1830 -----
1831 : SET MAINT1 IN REG 13, CHK FOR TEST MODE SET IN REG 17
1832 -----
1833 033430 BGNSUB
1833 033430 T23.8: TRAP C$BSUB
1833 033430 104402
1834 033432 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1835 033436 112737 000010 002352 MOVB #MAINT1,WRIBYT
1836 033444 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1837 033452 004737 003422 JSR PC,WRITLU ;SET MAINT1 IN REG 13
1838 033456 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1839 033464 142777 000010 146746 BICB #LULOOP,@BSEL1 ;CLEAR LULOOP
1840 033472 004737 003344 JSR PC,READLU ;READ REG 17
1841 033476 132737 000004 002350 BITB #TESTMD,REDBYT ;SEE IF TESTMD = 1
1842 033504 001006 BNE 6$ ;BR IF TESTMD = 1
1843 033506 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1844 :REPORT TEST MODE NOT SET BY MAINT1
1845 033512 ERRDF 52,EM52,ERR7
1845 033512 104455 TRAP C$ERDF
1845 033514 000064 .WORD 52
1845 033516 014443 .WORD EM52
1845 033520 020216 .WORD ERR7
1846 033522 6$:
1847 033522 ENDSUB
1847 033522 L10073: TRAP C$ESUB
1847 033522 104403
1848
1849 -----
1850 : SET SELSBY IN REG 13, CHK FOR STBY SET IN REG 13
1851 -----
    
```

```

1852 033524          BGNSUB
      033524          T23.9: TRAP C$BSUB
      033524 104402
1853 033526 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1854 033532 112737 000002 002352 MOVB #SELSBY,WRIBYT
1855 033540 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1856 033546 004737 003422 JSR PC,WRITLU ;SET SELSBY IN REG 13
1857 033552 004737 003344 JSR PC,READLU ;READ REG 13
1858 033556 132737 000002 002350 BITB #STBY,REDBYT ;SEE IF STBY = 1
1859 033564 001006 BNE 6$ ;BR IF STBY = 1
1860 033566 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1861 ;REPORT STBY NOT SET
1862 033572 ERRDF 59,EM59,ERR7
      033572 104455 TRAP C$ERDF
      033574 000073 .WORD 59
      033576 014637 .WORD EM59
      033600 020216 .WORD ERR7
1863 033602 6$:
1864 033602 ENDSUB
      033602 L10074: TRAP C$ESUB
      033602 104403
1865 ;-----
1866 ; DO MASTER CLEAR, LOAD 2 TSOM'S INTO TX SILO, CLOCK TRANSMITTER UNTIL ACTIVE,
1867 ; CHECK FOR RTS, CS, CARR SET IN REG 13
1868 ;-----
1869 033604          BGNSUB
      033604          T23.10: TRAP C$BSUB
      033604 104402
1870 033606 004737 005212 JSR PC,INITRN ;MST CLR, LOAD SOM'S, CLK TRANSMITTER
1871 033612 000000 000
1872 033614 000000 000
1873 033616 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1874 033624 004737 003344 JSR PC,READLU ;READ REG 13
1875 033630 132737 000040 002350 BITB #RTS,REDBYT ;SEE IF RTS = 1
1876 033636 001010 BNE 6$ ;BR IF RTS = 1
1877 033640 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1878 ;REPORT RTS NOT SET
1879 033644 ERRDF 60,EM60,ERR7
      033644 104455 TRAP C$ERDF
      033646 000074 .WORD 60
      033650 014654 .WORD EM60
      033652 020216 .WORD ERR7
1880 033654          ESCAPE SUB
      033654 104410 TRAP C$ESCAPE
      033656 000056 .WORD L10075-.
1881 033660 132737 000004 002350 6$: BITB #CS,REDBYT ;SEE IF CS = 1
1882 033666 001010 BNE 9$ ;BR IF CS = 1
1883 033670 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1884 ;REPORT CS NOT SET
1885 033674 ERRDF 61,EM61,ERR7
      033674 104455 TRAP C$ERDF
      033676 000075 .WORD 61
      033700 014670 .WORD EM61
      033702 020216 .WORD ERR7
1886 033704          ESCAPE SUB
      033704 104410 TRAP C$ESCAPE
      033706 000026 .WORD L10075-.
  
```



```

1887 033710 132737 000001 002350 9$:   BITB   #CARR,REDBYT   ;SEE IF CARR = 1
1888 033716 001006                BNE     12$           ;BR IF CARR = 1
1889 033720 004737 004200                JSR     PC,GETALL    ;GET REGS FOR PRINTOUT
1890                                ;REPORT CARR NOT SET
1891                                ERRDF   62,EM62,ERR7
                                TRAP   C$ERDF
                                .WORD  62
                                .WORD  EM62
                                .WORD  ERR7
1892 033734 104455                12$:
1893 033726 000076                ENDSUB
                                L10075:
                                TRAP   C$ESUB
1894 033730 014703
1895 033732 020216
1896 033734 104403
1897 033736 004737 003276                A12:
                                JSR     PC,MSTCLR   ;ISSUE MASTER CLEAR TO CLEAN UP
1898 033742 104401                ENDTST
                                L10063:
                                TRAP   C$ETST
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
    
```

```

:*****
:SBTTL      TEST 24 - DATA TEST - BIT MODE, NO ERR DET
:*
:* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
:* INHIBITED. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
:* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
:* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
:* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
:* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
:* TEST WILL NOT BE RUN.
:* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
:*             375,373,367,357,337,277,177
:* 8-BIT CHARACTERS ARE USED.
:*****
BGNTST
    
```

```

1918 033744 012737 000030 002434                MOV     #24,,TSTNUM   ;SET TEST NO.
1919 033752 012737 034132 002346                MOV     #24$,RETADR  ;SET TEST EXIT ADDRESS FOR ERRORS
1920 033760 004737 003276                JSR     PC,MSTCLR    ;ISSUE MASTER CLEAR
1921 033764 004737 010770                JSR     PC,CKLPBK    ;CHECK LOOPBACK, GET MODEM SELECTION
1922 033770 000000                0
1923 033772 013737 002422 034010                MOV     MODINT,6$    ;SET MODEM SELECTION
1924 034000 004737 010276                JSR     PC,SETUP     ;PROGRAM THE USYRT
1925 034004 000000                000
1926 034006 000300                CRC2!CRC1           ;BIT MODE, NO ERR DET
1927 034010 000000                6$:   .WORD  0        ;MODEM SELECTION GOES HERE
1928 034012 000000                000
1929 034014 004737 010156                JSR     PC,LODATA    ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1930 034020 012737 001177 003100                MOV     #RXEBL!177,RCVBUF+78. ; SET LAST DATA CHAR IN BUFFER
1931 034026 142777 000010 146404                BICB   #LULOOP,@BSEL1 ;CLEAR LULOOP, CLOCK MSG
1932 034034 012737 000012 002364                MOV     #12,REGNUM   ;SET LU REG NO. = 12
1933 034042 012703 002762                MOV     #RCVBUF,R3   ;GET POINTER TO RCV MSG BUF
1934 034046 013702 002264                9$:   MOV     TCOUNT,R2 ;INIT TIMER
    
```

```

1935 034052 004737 003344 10$: JSR PC,READLU ;READ REG 12
1936 034056 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1937 034064 001011 BNE 12$ ;BR IF YES
1938 034066 005202 INC R2 ;INCREMENT TIMER
1939 034070 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1940 034072 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1941 ;REPORT IRDY NOT SET
1942 034076 ERRDF 17,EM17,ERR7
                                TRAP C$ERDF
                                .WORD 17
                                .WORD EM17
                                .WORD ERR7
1943 034106 000411 BR 24$ ;ESCAPE TO END OF TEST
1944 034110 012337 034120 12$: MOV (R3)+,16$
1945 034114 004737 007266 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1946 034120 000000 16$: 0
1947 034122 000000 0
1948 034124 020327 003102 CMP R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
1949 034130 103746 BLO 9$ ;BR IF NOT YET
1950 034132 24$:
1951 034132 ENDTST
                                L10076:
                                TRAP C$ETST
1952 034132 104401
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971 034134
    
```

```

:*****
:SBTTL TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
:*
:* A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
:* INHIBITED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
:* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
:* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
:* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
:* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
:* TEST WILL NOT BE RUN.
:* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
:* 375,373,367,357,337,277,177
:* 8-BIT CHARACTERS ARE USED.
:*****
BGNTST
    
```

```

1972 034134 012737 000031 002434 MOV #25.,TSTNUM ;SET TEST NO.
1973 034142 012737 034322 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1974 034150 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1975 034154 004737 010770 JSR PC,CKLPBK ;CHECK LOOPBACK, GET MODEM SELECTION
1976 034160 000000 0
1977 034162 013737 002422 034200 MOV MODINT,6$ ;SET MODEM SELECTION
1978 034170 004737 010276 JSR PC,SETUP ;PROGRAM THE USYRT
1979 034174 000226 SYNCH
1980 034176 000311 CRC2!CRC1!STRIP!DDCMP ;CHAR MODE, NO ERR DET
1981 034200 000000 6$: .WORD 0 ;MODEM SELECTION GOES HERE
1982 034202 000000 000
1983 034204 004737 010156 JSR PC,LODATA ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1984 034210 012737 000177 003100 MOV #177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
    
```



```

2035 034370 000000          6$:      .WORD      0          ;MODEM SELECTION GOES HERE
2036 034372 000000          000
2037 034374 004737 010156   JSR      PC,LODATA      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2038 034400 012737 101177 003100   MOV      #CRCCHK!RXEBL!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
2039 034406 142777 000010 146024   BICB     #LULOOB,@BSEL1 ;CLEAR LULOOB, CLOCK MSG
2040 034414 012737 000012 002364   MOV      #12,REGNUM     ;SET LU REG NO. = 12
2041 034422 012703 002762          MOV      #RCVBUF,R3     ;GET POINTER TO RCV MSG BUF
2042 034426 013702 002264          MOV      TCOUNT,R2    ;INIT TIMER
2043 034432 004737 003344          9$:      JSR      PC,READLU ;READ REG 12
2044 034436 132737 000020 002350 10$:     BITB     #IRDY,REDBYT  ;SEE IF IRDY IS SET YET
2045 034444 001011          BNE     12$            ;BR IF YES
2046 034446 005202          INC     R2             ;INCREMENT TIMER
2047 034450 001370          BNE     10$           ;BR IF NO TIME-OUT YET
2048 034452 004737 004200          JSR     PC,GETALL     ;GET REGS FOR PRINTOUT
2049
2050 034456          ;REPORT IRDY NOT SET
      034456 104455          ERRDF  17,EM17,ERR7
      034460 000021          TRAP   C$ERDF
      034462 013430          .WORD  17
      034464 020216          .WORD  EM17
2051 034466 000411          .WORD  ERR7
2052 034470 012337 034500          BR      24$           ;ESCAPE TO END OF TEST
2053 034474 004737 007266          12$:    MOV      (R3)+,16$
2054 034500 000000          JSR     PC,CKDATA     ;COMPARE RCV'D DATA CHAR TO EXPECTED
2055 034502 000000          16$:    0
2056 034504 020327 003102          0
2057 034510 103746          CMP     R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
2058 034512          BLO     9$            ;BR IF NOT YET
2059 034512          24$:    ENDTST
      034512          L10100:
      034512 104401          TRAP   C$ETST

2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079 034514          ;*****
      034514          ;SBTTL      TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
2080 034514 012737 000033 002434          ;*
2081 034522 012737 034702 002346          ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
2082 034530 004737 003276          ;* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
2083 034534 004737 010770          ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
2084 034540 000000          ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
          ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
          ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
          ;* TEST WILL NOT BE RUN.
          ;* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
          ;* 375,373,367,357,337,277,177
          ;* 8-BIT CHARACTERS ARE USED.
          ;*****
          BGN:TST
          T27::
          MOV     #27, TSTNUM ;SET TEST NO.
          MOV     #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
          JSR     PC,MSTCLR  ;ISSUE MASTER CLEAR
          JSR     PC,CKLPBK  ;CHECK LOOPBACK, GET MODEM SELECTION
          0
    
```

```

2085 034542 013737 002422 034560      MOV      MODINT,6$      ;SET MODEM SELECTION
2086 034550 004737 010276      JSR      PC,SETUP      ;PROGRAM THE USYRT
2087 034554 000000      000
2088 034556 000100      CRC1          ;BIT MODE, CRC-CCITT-0
2089 034560 000000      5$: .WORD      0          ;MODEM SELECTION GOES HERE
2090 034562 000000      000
2091 034564 004737 010156      JSR      PC,LODATA     ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2092 034570 012737 101177 003100      MOV      #CRCCHK!RXEBL!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
2093 034576 142777 000010 145634      BICB     #LULOOK,@BSEL1 ;CLEAR LULOOK, CLOCK MSG
2094 034604 012737 000012 002364      MOV      #12,REGNUM    ;SET LU REG NO. = 12
2095 034612 012703 002762      MOV      #RCVBUF,R3    ;GET POINTER TO RCV MSG BUF
2096 034616 013702 002264      9$: MOV      TCOUNT,R2 ;INIT TIMER
2097 034622 004737 003344      10$: JSR      PC,READLU  ;READ REG 12
2098 034626 132737 000020 002350      BITB     #IRDY,REDBYT  ;SEE IF IRDY IS SET YET
2099 034634 001011      BNE     12$           ;BR IF YES
2100 034636 005202      INC     R2           ;INCREMENT TIMER
2101 034640 001370      BNE     10$          ;BR IF NO TIME-OUT YET
2102 034642 004737 004200      JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
2103      ;REPORT IRDY NOT SET
2104 034646      ERRDF  17,EM17,ERR7
                                TRAP  C$ERDF
                                .WORD 17
                                .WORD EM17
                                .WORD ERR7
2105 034656 000411      BR      24$          ;ESCAPE TO END OF TEST
2106 034660 012337 034670      12$: MOV      (R3)+,16$
2107 034664 004737 007266      JSR      PC,CKDATA    ;COMPARE RCV'D DATA CHAR TO EXPECTED
2108 034670 000000      16$: 0
2109 034672 000000      0
2110 034674 020327 003102      CMP     R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
2111 034700 103746      BLO     9$           ;BR IF NOT YET
2112 034702      24$:
2113 034702      ENDTST
                                L10101:
                                TRAP  C$ETST
2114 034702 104401
2115
2116
2117
2118
2119
2120      ;*****
2121      ;SBTTL      TEST 28 - DATA TEST - CHAR MODE, CRC-16
2122      ;*
2123      ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
2124      ;* DETECTION. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
2125      ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
2126      ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
2127      ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
2128      ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
2129      ;* TEST WILL NOT BE RUN.
2130      ;* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
2131      ;* 375,373,367,357,337,277,177
2132      ;* 8-BIT CHARACTERS ARE USED.
2133      ;*****
2134 034704 012737 000034 002434      MOV      #28.,TSTNUM  ;SET TEST NO.
                                T28::
    
```

```

2135 034712 012737 035072 002346      MOV    #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
2136 034720 004737 003276              JSR    PC,MSTCLR        ;ISSUE MASTER CLEAR
2137 034724 004737 010770              JSR    PC,CKLPBK        ;CHECK LOOPBACK, GET MODEM SELECTION
2138 034730 000000 000000 000000              0
2139 034732 013737 002422 034750      MOV    MODINT,6$        ;SET MODEM SELECTION
2140 034740 004737 010276              JSR    PC,SETUP         ;PROGRAM THE USYRT
2141 034744 000226 000000 000000      SYNCH
2142 034746 000011 000000 000000      STRIP!DDCMP
2143 034750 000000 000000 000000      6$: .WORD 0              ;MODEM SELECTION GOES HERE
2144 034752 000000 000000 000000      000
2145 034754 004737 010156 000000      JSR    PC,LODATA        ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2146 034760 012737 100577 003100      MOV    #CRCCHK!RXBCC!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
2147 034766 142777 000010 145444      BICB  #LULOOK,@BSEL1   ;CLEAR LULOOK, CLOCK MSG
2148 034774 012737 000012 002364      MOV    #12,REGNUM       ;SET LU REG NO. = 12
2149 035002 012703 002762 000000      MOV    #RCVBUF,R3       ;GET POINTER TO RCV MSG BUF
2150 035006 013702 002264 000000      9$: MOV    TCOUNT,R2   ;INIT TIMER
2151 035012 004737 003344 000000      10$: JSR    PC,READLU    ;READ REG 12
2152 035016 132737 000020 002350      BITB  #IRDY,REDBYT      ;SEE IF IRDY IS SET YET
2153 035024 001011 000000 000000      BNE   12$               ;BR IF YES
2154 035026 005202 000000 000000      INC   R2                ;INCREMENT TIMER
2155 035030 001370 000000 000000      BNE   10$               ;BR IF NO TIME-OUT YET
2156 035032 004737 004200 000000      JSR    PC,GETALL        ;GET REGS FOR PRINTOUT
2157                                     ;REPORT IRDY NOT SET
2158 035036 000000 000000 000000      ERRDF 17,EM17,ERR7
                                     TRAP  C$ERDF
035036 104455                                     .WORD 17
035040 000021                                     .WORD EM17
035042 013430                                     .WORD ERR7
035044 020216                                     .WORD
2159 035046 000411 000000 000000      BR    24$               ;ESCAPE TO END OF TEST
2160 035050 012337 035060 000000      12$: MOV    (R3)+,16$
2161 035054 004737 007266 000000      JSR    PC,CKDATA        ;COMPARE RCV'D DATA CHAR TO EXPECTED
2162 035060 000000 000000 000000      16$: 0
2163 035062 000000 000000 000000      0
2164 035064 020327 003102 000000      CMP    R3,#RCVBUF+80.   ;SEE IF ALL CHARS CHECKED YET
2165 035070 103746 000000 000000      BLO   9$                ;BR IF NOT YET
2166 035072 000000 000000 000000      24$:
2167 035072 000000 000000 000000      ENDTST
                                     L10102: TRAP  C$ETST
035072 104401

```

2168
 2169
 2170
 2171
 2172
 2173
 2174
 2175
 2176
 2177
 2178
 2179
 2180
 2181
 2182
 2183
 2184
 2185

```

:*****
:SBTTL      TEST 29 - DATA TEST - CHAR MODE, ODD VRC
:*
:* A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
:* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
:* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
:* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
:* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
:* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
:* TEST WILL NOT BE RUN.
:* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
:*             375,373,367,357,337,277,177
:* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).

```

```

2186
2187 035074          ::*****
                BGNTST
                T29::
2188 035074 012737 000035 002434      MOV    #29.,TSTNUM      ;SET TEST NO.
2189 035102 012737 035256 002346      MOV    #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
2190 035110 004737 003276              JSR    PC,MSTCLR       ;ISSUE MASTER CLEAR
2191 035114 004737 010770              JSR    PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
2192 035120 000000                      0
2193 035122 013737 002422 035140      MOV    MODINT,6$       ;SET MODEM SELECTION
2194 035130 004737 010276              JSR    PC,SETUP        ;PROGRAM THE USYRT
2195 035134 000026                      026
2196 035136 000111                      CRC1!STRIP!DDCMP
2197 035140 000000                      6$: .WORD 0                ;MODEM SELECTION GOES HERE
2198 035142 000347                      TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO
2199 035144 004737 010156              JSR    PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2200 035150 142777 000010 145262      BICB   #LULoop,@BSEL1  ;CLEAR LULoop, CLOCK MSG
2201 035156 012737 000012 002364      MOV    #12,REGNUM      ;SET LU REG NO. = 12
2202 035164 012703 002762              MOV    #RCVBUF,R3     ;GET POINTER TO RCV MSG BUF
2203 035170 013702 002264              9$: MOV    TCOUNT,R2   ;INIT TIMER
2204 035174 004737 003344              10$: JSR    PC,READLU    ;READ REG 12
2205 035200 132737 000020 002350      BITB   #IRDY,REDBYT   ;SEE IF IRDY IS SET YET
2206 035206 001011                      BNE    12$             ;BR IF YES
2207 035210 005202                      INC    R2              ;INCREMENT TIMER
2208 035212 001370                      BNE    10$             ;BR IF NO TIME-OUT YET
2209 035214 004737 004200              JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
2210
2211 035220          ;REPORT IRDY NOT SET
                ERRDF 17,EM17,ERR7
                TRAP  C$ERDF
                .WORD 17
                .WORD EM17
                .WORD ERR7
                2212 035220 104455
                2213 035222 000021
                2214 035224 013430
                2215 035226 020216
                2216 035230 000412
                2217 035232 112337 035244      12$: BR    24$          ;ESCAPE TO END OF TEST
                2218 035236 005203              MOV    (R3)+,16$      ;GET AN EXPECTED DATA BYTE
                2219 035240 004737 007266      INC    R3              ;INCREMENT POINTER
                2220 035244 100000              JSR    PC,CKDATA      ;COMPARE RCV'D DATA CHAR TO EXPECTED
                2221 035246 000000              16$: BCCCHK
                2222 035250 020327 003102      0
                2223 035254 103745              CMP    R3,#RCVBUF+80. ;SFE IF ALL CHARS CHECKED YET
                2224 035256              BLO    9$             ;BR IF NOT YET
                2225
                2226
                2227
                2228          24$:
                2229          ENDTST
                L10103:
                2230          TRAP  C$ETST
                2231
                2232
                2233
                2234
                2235
                ::*****
                SBTTL TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
                *
                * A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
                * SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
                * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
                * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
                * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
                * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
    
```

```

2236      ;* TEST WILL NOT BE RUN.
2237      ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
2238      ;*                   375,373,367,357,337,277,177
2239      ;* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
2240      ;*****
2241 035260 BGNTST
          035260
2242 035260 012737 000036 002434      MOV      #30.,TSTNUM      ;SET TEST NO.
2243 035266 012737 035442 002346      MOV      #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
2244 035274 004737 003276              JSR      PC,MSTCLR       ;ISSUE MASTER CLEAR
2245 035300 004737 010770              JSR      PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
2246 035304 000000
2247 035306 013737 002422 035324      MOV      MODINT,6$      ;SET MODEM SELECTION
2248 035314 004737 010276              JSR      PC,SETUP        ;PROGRAM THE USYRT
2249 035320 000026
2250 035322 000211
2251 035324 000000
2252 035326 000347      6$:      CRC2!STRIP!DDCMP      ;MODEM SELECTION GOES HERE
          .WORD 0
          TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO
2253 035330 004737 010156              JSR      PC,LODATA      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2254 035334 142777 000010 145076      BICB     #LULoop,@BSEL1 ;CLEAR LULoop, CLOCK MSG
2255 035342 012737 000012 002364      MOV      #12,REGNUM     ;SET LU REG NO. = 12
2256 035350 012703 002762              MOV      #RCVBUF,R3     ;GET POINTER TO RCV MSG BUF
2257 035354 0i3702 002264      9$:      MOV      TCOUNT,R2 ;INIT TIMER
2258 035360 004737 003344      10$:     JSR      PC,READLU     ;READ REG 12
2259 035364 132737 000020 002350      BITB     #IRDY,REDBYT   ;SEE IF IRDY IS SET YET
2260 035372 001011              BNE      12$            ;BR IF YES
2261 035374 005202              INC      R2             ;INCREMENT TIMER
2262 035376 001370              BNE      10$           ;BR IF NO TIME-OUT YET
2263 035400 004737 004200              JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
2264      ;REPORT IRDY NOT SET
2265 035404      ERRDF 17,EM17,ERR7
          035404 104455
          035406 000021
          035410 013430
          035412 020216
          035414 000412
2266 035414 000412
2267 035416 112337 035430      12$:     BR      24$      ;ESCAPE TO END OF TEST
2268 035422 005203              MOVB     (R3)+,16$     ;GET AN EXPECTED DATA CHAR
2269 035424 004737 007266              INC      R3            ;INCREMENT POINTER
2270 035430 100000              JSR      PC,CKDATA     ;COMPARE RCV'D DATA CHAR TO EXPECTED
2271 035432 000000      16$:     BCCCHK
          0
2272 035434 020327 003102      CMP      R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
2273 035440 103745              BLO      9$            ;BR IF NOT YET
2274 035442
2275 035442      24$:     ENDTST
          035442
          035442 104401
          L10104:
          TRAP  C$ETST

2276
2277
2278
2279
2280
2281      ;*****
2282      ;SBITL TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
2283      ;*
2284      ;* IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
2285      ;* IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
  
```



```
2286          ;* FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
2287          ;* THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
2288          ;* ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
2289          ;* ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
2290          ;* AND TRANSMITTED AND RECEIVED CORRECTLY.
2291          ;*****
2292 035444      BGNTST
2293 035444      012737 035550 002346      MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
2294 035452      004737 005212      JSR      PC,INITRN      ;MST CLR, LOAD 2 SOM'S
2295 035456      000377
2296 035460      000320      CRC2!CRC1!SECA      ;BIT MODE, NO ERROR DET, SEC ADR MODE
2297 035462      004737 010720      JSR      PC,LODSIL      ;LOAD 5 377-CHARS INTO TX SILO
2298 035466      000377
2299 035470      000005
2300 035472      004737 010720      JSR      PC,LODSIL      ;LOAD 2 EOM'S INTO TX SILO
2301 035476      001000      TXEOM
2302 035500      000002
2303 035502      004737 004726      JSR      PC,STPLU      ;CLOCK MORE THAN ENTIRE MSG
2304 035506      000240
2305 035510      004737 007266      JSR      PC,CKDATA      ;RCV SEC ADRS = 377
2306 035514      000377
2307 035516      000000
2308 035520      012701 000003      MOV      #3,R1      ;RCV 3 MORE 377 CHARS
2309 035524      004737 007266      JSR      PC,CKDATA
2310 035530      000377
2311 035532      000000
2312 035534      005301      DEC      R1
2313 035536      001372      BNE      6$
2314 035540      004737 007266      JSR      PC,CKDATA      ;RCV LAST 377 CHAR, CHK EBLK = 1
2315 035544      001377
2316 035546      000000
2317 035550      004737 003276      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
2318 035554      ENDTST
2319 035554      104401
2320
2321
2322
2323
2324
2325          ;*****
2326          ;SBTTL      TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
2327          ;*
2328          ;* IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
2329          ;* DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
2330          ;* FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
2331          ;* AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
2332          ;* AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1' IS
2333          ;* SENT WITH ITS CORRESPONDING CRC.
2334          ;* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
2335          ;* 375,373,367,357,337,277,177
2336          ;* MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,000,SYNCH,SYNCH
2337          ;*****
2338 035556      012737 036334 002346      MOV      #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
```

T31::

L10105:

TRAP C\$ETST

T32::

```

2339
2340 -----
2341 : TRANSMIT AND RCV ENTIRE MSG
2342 035564 004737 005212 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
2343 035570 000226 SYNCH
2344 035572 000011 STRIP!DDCMP
2345 035574 004737 010546 JSR PC,LDBYTS ;LOAD 20 WORDS OF PAT A INTO TX SILO
2346 035600 002557 PATA
2347 035602 000024 20.
2348 035604 004737 010720 JSR PC,LODSIL ;LOAD AN EOM INTO TX SILO
2349 035610 001000 TXEOM
2350 035612 000001 1
2351 035614 004737 010546 JSR PC,LDBYTS ;LOAD 20 WORDS OF PAT A INTO TX SILO
2352 035620 002557 PATA
2353 035622 000024 20.
2354 035624 004737 010720 JSR PC,LODSIL ;LOAD 1 EOM INTO TX SILO
2355 035630 001000 TXEOM
2356 035632 000001 1
2357 035634 004737 010720 JSR PC,LODSIL ;LOAD 3 SOM'S INTO TX SILO
2358 035640 000400 TXSOM
2359 035642 000003 3
2360 035644 004737 010470 JSR PC,LODMSG ;LOAD MSG1 INTO TX SILO
2361 035650 002670 MSG1
2362 035652 000013 11.
2363 035654 004737 004726 JSR PC,STPLU ;CLOCK HDR MSG AND CRC CHARS
2364 035660 000300 192.
2365 035662 012737 000013 002364 MOV #13,REGNUM ;SET REG. NO. = 13
2366 035670 004737 003344 JSR PC,READLU ;READ REG 13
2367 035674 032737 000040 002350 BIT #RTS,REDBYT ;SEE IF RTS SET
2368 035702 001010 BNE 2$ ;BR IF RTS SET
2369 035704 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
2370 ;REPORT RTS NOT SET
2371 035710 ERRDF 60,EM60,ERR7
2372 035710 104455 TRAP C$ERDF
2373 035712 000074 .WORD 60
2374 035714 014654 .WORD EM60
2375 035716 020216 .WORD ERR7
2376 035720 000137 036334 JMP 24$ ;EXIT TEST
2377 035724 004737 004726 2$: JSR PC,STPLU ;CLK DATA MSG AND FIRST CRC CHAR
2378 035730 000250 168.
2379 035732 012703 000040 MOV #32,R3 ;SET COUNTER FOR CHECKING RTS
2380 035736 004737 004726 4$: JSR PC,STPLU ;CLOCK LINE UNIT FOR 1 CYCLE
2381 035742 000001 1
2382 035744 004737 003344 JSR PC,READLU ;READ REG 13
2383 035750 032737 000040 002350 BIT #RTS,REDBYT ;CHK FOR RTS SET
2384 035756 001007 BNE 5$ ;BR IF RTS SET
2385 035760 004737 004200 JSR PC,GETALL ;GET REGS FOR PRINTOUT
2386 035764 ERRDF 60,EM60,ERR7 ;REPORT RTS NOT SET
2387 035764 104455 TRAP C$ERDF
2388 035766 000074 .WORD 60
2389 035770 014654 .WORD EM60
2390 035772 020216 .WORD ERR7
2391 035774 000557 2384 BR 24$
2392 035776 005303 2385 DEC R3 ;DECR COUNTER
2393 036000 001356 2386 BNE 4$ ;BR IF NOT DONE YET

```

```

2388      : READ AND CHK HEADER AND CRC
2389      -----
2390 036002 012701 002557      MOV    #PATA,R1      ;INIT PATTERN A POINTER
2391 036006 112137 036016 7$:  MOVVB  (R1)+,8$      ;GET AN EXPECTED CHAR
2392 036012 004737 007266      JSR    PC,CKDATA    ;READ AND CHK A CHAR
2393 036016 000000 8$:    .WORD  0
2394 036020 000000      0
2395 036022 020127 002601      CMP    R1,#PATB-2   ;SEE IF CHKING NEXT-TO-LAST CHAR YET
2396 036026 103767      BLO   7$            ;BR IF NOT YET
2397 036030 004737 007266      JSR    PC,CKDATA    ;READ AND CHK CHAR, BCC=0
2398 036034 100277      CRCCHK!277
2399 036036 000000      0
2400 036040 004737 007266      JSR    PC,CKDATA    ;READ AND CHK LAST CHAR, BCC=1
2401 036044 100577      CRCCHK!RXBCC!177
2402 036046 000000      0
2403 036050 004737 007266      JSR    PC,CKDATA    ;READ AND CHK HI CRC BYTE
2404 036054 000156      156
2405 036056 000000      0
2406 036060 004737 007266      JSR    PC,CKDATA    ;READ AND CHK LO CRC BYTE
2407 036064 000236      236
2408 036066 000000      0
2409      -----
2410      : READ AND CHK DATA MSG AND CRC
2411      -----
2412 036070 012701 002557      MOV    #PATA,R1      ;INIT PATTERN A POINTER
2413 036074 112137 036104 9$:  MOVVB  (R1)+,12$     ;GET AN EXPECTED CHAR
2414 036100 004737 007266      JSR    PC,CKDATA    ;READ AND CHK A CHAR
2415 036104 000000 12$:  .WORD  0
2416 036106 000000      0
2417 036110 020127 002601      CMP    R1,#PATB-2   ;SEE IF CHKING NEXT-TO-LAST CHAR YET
2418 036114 103767      BLO   9$            ;BR IF NOT YET
2419 036116 004737 007266      JSR    PC,CKDATA    ;READ AND CHK CHAR, BCC=0
2420 036122 100277      CRCCHK!277
2421 036124 000000      0
2422 036126 004737 007266      JSR    PC,CKDATA    ;READ AND CHK LAST CHAR, BCC=1
2423 036132 100577      CRCCHK!RXBCC!177
2424 036134 000000      0
2425 036136 004737 007266      JSR    PC,CKDATA    ;READ AND CHK HI CRC BYTE
2426 036142 000156      156
2427 036144 000000      0
2428 036146 004737 007266      JSR    PC,CKDATA    ;READ AND CHK LO CRC BYTE
2429 036152 000236      236
2430 036154 000000      0
2431      -----
2432      : CLOCK 3RD MESSAGE ('MSG1' DATA)
2433      -----
2434 036156 012737 000012 002364  MOV    #12,REGNUM    ;SET REG NO. = 12
2435 036164 112737 000200 002352  MOVVB  #1C,WRIBYT    ;SET IC TO CLEAR RECEIVER FOR NEW MSG
2436 036172 004737 003422      JSR    PC,WRITLU
2437 036176 012737 000013 002364  MOV    #13,REGNUM    ;RESTORE REG NO. TO 13
2438 036204 004737 004726      JSR    PC,STPLU     ;CLOCK THE REST OF MSG
2439 036210 000150      104.
2440 036212 004737 003344      JSR    PC,READLU    ;READ REG 13
2441 036216 032737 000040 002350  BIT    #RTS,REDBYT   ;SEE IF RTS IS CLEARED
2442 036224 001407      BEQ   14$            ;BR IF RTS CLEARED
2443 036226 004737 004200      JSR    PC,GETALL    ;GET REGS FOR PRINTOUT
2444      :REPORT RTS NOT CLEARED
    
```

2445 036232
 036232 104455
 036234 000101
 036236 014752
 036240 020216
 2446 036242 000434

ERRDF 65,EM65,ERR7

TRAP C\$ERDF
 .WORD 65
 .WORD EM65
 .WORD ERR7

BR 24\$

 :READ AND CHECK 3RD MESSAGE AND CRC

2450	036244	004737	007266	14\$:	JSR	PC,CKDATA	;READ AND CHECK 000 DATA CHAR
2451	036250	000000				000	
2452	036252	000000				0	
2453	036254	004737	007266		JSR	PC,CKDATA	;READ AND CHECK 125 DATA CHAR
2454	036260	000125				125	
2455	036262	000000				0	
2456	036264	004737	007266		JSR	PC,CKDATA	;READ AND CHECK 252 DATA CHAR
2457	036270	000252				252	
2458	036272	000000				0	
2459	036274	004737	007266		JSR	PC,CKDATA	;READ AND CHECK 377 DATA CHAR, AND BCC=0
2460	036300	100377				CRCCHK!377	
2461	036302	000000				0	
2462	036304	004737	007266		JSR	PC,CKDATA	;READ AND CHECK 000 DATA CHAR, AND BCC=1
2463	036310	100400				CRCCHK!RXBCC!000	
2464	036312	000000				0	
2465	036314	004737	007266		JSR	PC,CKDATA	;READ AND CHK HI CRC BYTE
2466	036320	000160				160	
2467	036322	000000				0	
2468	036324	004737	007266		JSR	PC,CKDATA	;READ AND CHK LO CRC BYTE
2469	036330	000034				034	
2470	036332	000000				0	
2471	036334	004737	003276	24\$:	JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
2472	036340			ENDTST			
	036340						L10106:
	036340	104401					TRAP C\$ETST

2473
 2474
 2475
 2476
 2477

.SBTTL HARDWARE PARAMETER CODING SECTION

```

////////////////////////////////////
// THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
// THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
// MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
// INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
// MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
// WITH THE OPERATOR.
////////////////////////////////////
    
```

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14	036342		BGNHRD		
	036342	000046			.WORD L10107-L\$HARD/2
	036344				L\$HARD::
15					
16	036344		GPRMA	ADDRES,2,0,160000,177776,YES	
	036344	001031			.WORD T\$CODE
	036346	036460			.WORD ADDRES
	036350	160000			.WORD T\$LLOLIM
	036352	177776			.WORD T\$HILIM
17	036354		GPRMA	VECTOR,4,0,0,674,YES	
	036354	002031			.WORD T\$CODE
	036356	036506			.WORD VECTOR
	036360	000000			.WORD T\$LLOLIM
	036362	000674			.WORD T\$HILIM
18	036364		GPRMD	PRIRTY,6,0,7000,4,7,YES	
	036364	003032			.WORD T\$CODE
	036366	036537			.WORD PRIRTY
	036370	007000			.WORD 7000
	036372	000004			.WORD T\$LLOLIM
	036374	000007			.WORD T\$HILIM
19	036376		GPRMD	SWPAC1,12,0,377,0,056,YES	
	036376	005032			.WORD T\$CODE
	036400	036570			.WORD SWPAC1
	036402	000377			.WORD 377
	036404	000000			.WORD T\$LLOLIM
	036406	000056			.WORD T\$HILIM
20	036410		GPRMD	SWPAC2,14,0,377,0,377,YES	
	036410	006032			.WORD T\$CODE
	036412	036631			.WORD SWPAC2
	036414	000377			.WORD 377
	036416	000000			.WORD T\$LLOLIM
	036420	000377			.WORD T\$HILIM
21	036422		GPRMD	SWPAC3,16,0,377,0,377,YES	
	036422	007032			.WORD T\$CODE
	036424	036672			.WORD SWPAC3
	036426	000377			.WORD 377
	036430	000000			.WORD T\$LLOLIM
	036432	000377			.WORD T\$HILIM
22	036434		GPRMD	LOOPBK,20,0,7,0,4,YES	
	036434	010032			.WORD T\$CODE
	036436	036733			.WORD LOOPBK
	036440	000007			.WORD 7
	036442	000000			.WORD T\$LLOLIM
	036444	000004			.WORD T\$HILIM

23	036446			GPRMD	BAUDRT,22,0,7,0,7,YES		
	036446	011032				.WORD	T\$CODE
	036450	037050				.WORD	BAUDRT
	036452	000007				.WORD	7
	036454	000000				.WORD	T\$LOLIM
	036456	000007				.WORD	T\$HILIM

24							
25	036460			ENDHRD			
	036460					L10107:	.EVEN

26							
27	036460	104	105	126	ADDRES: .ASCIZ /DEVICE CSR ADDRESS : /		
	036463	111	103	105			
	036466	040	103	123			
	036471	122	040	101			
	036474	104	104	122			
	036477	105	123	123			
	036502	040	072	040			
	036505	000					

28	036506	104	105	126	VECTOR: .ASCIZ /DEVICE VECTOR ADDRESS : /		
	036511	111	103	105			
	036514	040	126	105			
	036517	103	124	117			
	036522	122	040	101			
	036525	104	104	122			
	036530	105	123	123			
	036533	040	072	040			
	036536	000					

29	036537	104	105	126	PRIPTY: .ASCIZ /DEVICE PRIORITY LEVEL : /		
	036542	111	103	105			
	036545	040	120	122			
	036550	111	117	122			
	036553	111	124	131			
	036556	040	114	105			
	036561	126	105	114			
	036564	040	072	040			
	036567	000					

30	036570	115	070	062	SWPAC1: .ASCIZ /M8203 SWITCH PACK #1 (REG 11) : /		
	036573	060	063	040			
	036576	123	127	111			
	036601	124	103	110			
	036604	040	120	101			
	036607	103	113	040			
	036612	043	061	040			
	036615	050	122	105			
	036620	107	040	061			
	036623	061	051	040			
	036626	072	040	000			

31	036631	115	070	062	SWPAC2: .ASCIZ /M8203 SWITCH PACK #2 (REG 15) : /		
	036634	060	063	040			
	036637	123	127	111			
	036642	124	103	110			
	036645	040	120	101			
	036650	103	113	040			
	036653	043	062	040			
	036656	050	122	105			
	036661	107	040	061			

	036664	065	051	040
	036667	072	040	000
32	036672	115	070	062
	036675	060	063	040
	036700	123	127	111
	036703	124	103	110
	036706	040	120	101
	036711	103	113	040
	036714	043	063	040
	036717	050	122	105
	036722	107	040	061
	036725	066	051	040
	036730	072	040	000
33	036733	124	125	122
	036736	116	101	122
	036741	117	125	116
	036744	104	040	124
	036747	131	120	105
	036752	040	055	015
	036755	012		
34	036756	050	060	075
	036761	110	063	062
	036764	065	064	046
	036767	110	063	062
	036772	065	065	054
	036775	040	061	075
	037000	103	101	102
	037003	114	105	054
	037006	040	062	075
	037011	115	117	104
	037014	040	114	117
	037017	103	054	040
	037022	063	075	115
	037025	117	104	040
	037030	122	105	115
	037033	054	040	064
	037036	075	116	117
	037041	116	105	051
	037044	040	072	040
	037047	000		
35	037050	120	114	105
	037053	101	123	105
	037056	040	123	105
	037061	114	105	103
	037064	124	040	102
	037067	101	125	104
	037072	040	122	101
	037075	124	105	073
	037100	040	124	131
	037103	120	105	040
	037106	047	060	047
	037111	040	106	117
	037114	122	040	062
	037117	056	064	113
	037122	073	040	047
	037125	061	047	040
	037130	106	117	122

SWPAC3: .ASCIZ /M8203 SWITCH PACK #3 (REG 16) : /

LOUPBK: .ASCII /TURNARGUND TYPE -/<15><12>

.ASCIZ /(0=H3254&H3255, 1=CABLE, 2=MOD LOC, 3=MOD REM, 4=NONE) : /

BAUDRT: .ASCII /PLEASE SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K; /<15><12>

	037133	040	064	056
	037136	070	113	073
	037141	015	012	
36	037143	047	062	047
	037146	040	106	117
	037151	122	040	071
	037154	056	066	113
	037157	073	040	047
	037162	063	047	040
	037165	106	117	122
	037170	040	061	071
	037173	056	062	113
	037176	073	040	047
	037201	064	047	040
	037204	106	117	122
	037207	040	065	066
	037212	113	073	040
	037215	047	065	047
	037220	040	106	117
	037223	122	040	062
	037226	065	060	113
	037231	073	015	012
37	037234	047	066	047
	037237	040	106	117
	037242	122	040	065
	037245	060	060	113
	037250	073	040	117
	037253	122	040	047
	037256	067	047	040
	037261	106	117	122
	037264	040	061	040
	037267	115	105	107
	037272	040	102	101
	037275	125	104	040
	037300	072	040	000

.ASCII /'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K; /<15><12>

.ASCIZ /'6' FOR 500K; OR '7' FOR 1 MEG BAUD : /

.EVEN

38
39
40
41
42
43
44

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

.SBTTL SOFTWARE PARAMETER CODING SECTION

:/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 :/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
 :/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 :/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
 :/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 :/ WITH THE OPERATOR.

```

037304          BGNSFT
037304 000016
037306          GPRML  ISMANI,0,1,YES
037310 000130          .WORD  T$CODE
037312 037342          .WORD  ISMANI
037314 000001          .WORD  1
037314 001130          GPRML  ISPRNT,2,1,YES
037316 037430          .WORD  T$CODE
037320 000001          .WORD  ISPRNT
037322          GPRML  ISWPAK,4,1,YES
037322 002130          .WORD  T$CODE
037324 037513          .WORD  ISWPAK
037326 000001          .WORD  1
037330          GPRMD  TIMCNT,6,0,177777,0,177777,YES
037330 003032          .WORD  T$CODE
037332 037560          .WORD  TIMCNT
037334 177777          .WORD  177777
037336 000000          .WORD  T$LOLIM
037340 177777          .WORD  T$HILIM

037342          ENDSFT

037342          L10110: .EVEN

037342          L10110:
037342 111 123 040 ISMANI: .ASCIZ /IS MAN. INTERVEN. DESIRED TO MOUNT TEST CONNECTOR(S) /
037345 115 101 116
037350 056 040 111
037353 116 124 105
037356 122 126 105
037361 116 056 040
037364 104 105 123
037367 111 122 105
037372 104 040 124
037375 117 040 115
037400 117 125 116
037403 124 040 124
037406 105 123 124
037411 040 103 117
037414 116 116 105
037417 103 124 117
037422 122 050 123
037425 051 040 000
    
```

23	037430	123	110	117	ISPRNT: .ASCIZ /SHOULD SWITCH PACK AND AX3-15 PRINTOUT BE ALLOWED /
	037433	125	114	104	
	037436	040	123	127	
	037441	111	124	103	
	037444	110	040	120	
	037447	101	103	113	
	037452	040	101	116	
	037455	104	040	101	
	037460	130	063	055	
	037463	061	065	040	
	037466	120	122	111	
	037471	116	124	117	
	037474	125	124	040	
	037477	102	105	040	
	037502	101	114	114	
	037505	117	127	105	
	037510	104	040	000	
24	037513	123	110	117	ISWPAK: .ASCIZ /SHOULD SWITCH PACK TESTS BE ALLOWED /
	037516	125	114	104	
	037521	040	123	127	
	037524	111	124	103	
	037527	110	040	120	
	037532	101	103	113	
	037535	040	124	105	
	037540	123	124	123	
	037543	040	102	105	
	037546	040	101	114	
	037551	114	117	127	
	037554	105	104	040	
	037557	000			
25	037560	115	123	107	TIMCNT: .ASCIZ /MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : /
	037563	040	124	111	
	037566	115	105	122	
	037571	040	126	101	
	037574	114	125	105	
	037577	040	050	060	
	037602	055	061	067	
	037605	067	067	067	
	037610	067	051	054	
	037613	040	060	040	
	037616	075	040	114	
	037621	117	116	107	
	037624	105	123	124	
	037627	040	124	111	
	037632	115	105	055	
	037635	117	125	124	
	037640	040	072	040	
	037643	000			

26
27
28
29
30
31
32
33
34

.EVEN

35
36 037644
37 037744
38 037744 000240
39 037746 000240
40 037750 000240
41
42
43
44
45 037752
46
47 037752

037752 000000
037754 000000
037756
48
49 000001

***** PATCH AREA FOR DEBUG *****
PATCH:
 .=.+100
 NOP
 NOP
 NOP

 ENDMOD
 LASTAD

L\$LAST::

 .END

.EVEN
.WORD 0
.WORD 0

ABORT = 000004	BSEL4 002442	C\$RDBU= 000007	EM40 014016	FMT18 013034
ADDRES 036460	CARR = 000001	C\$REFG= 000047	EM41 014032	FMT19 013076
ADR = 000020 G	CHKABT 012130	C\$RESE= 000033	EM42 014053	FMT2 012266
ANBITS 002547	CHPCHK= 100000	C\$REVI= 000003	EM43 014070	FMT25 013127
APA = 000200	CHPTYP 002420	C\$RFLA= 000021	EM44 014115	FMT26 013157
ASBC0 = 000020	CKDATA 007266	C\$RPRT = 000025	EM45 014142	FMT27 023104
ASBC1 = 000040	CKLPBK 010770	C\$SEFG= 000046	EM46 014167	FMT3 012310
ASBC2 = 000100	CRCCHK= 100000	C\$SPRI= 000041	EM47 014222	FMT4 012352
ASSEMB= 000010	CRCTY0= 000001	C\$SVEC= 000037	EM48 014255	FMT5 012365
AXNUM 002366	CRCTY1= 000002	C\$TPRI= 000013	EM49 014310	FMT6 012415
AX0.15= 002306	CRCTY2= 000004	C32BCC= 000040	EM50 014347	FMT7 012450
AX0.16= 002310	CRC1 = 000100	C32ENB= 000004	EM51 014403	FMT8 012460
AX1 = 000001	CRC2 = 000200	DDC = 000100	EM52 014443	FMT9 012514
AX1.15= 002312	CS = 000004	DDCMP = 000001	EM53 014477	FRSPAS 002400
AX1.16= 002314	C\$AU = 000052	DEVMAP 002426	EM54 014533	FRSTIM 002376
AX2 = 000002	C\$AUTO= 000061	DEVPTR 002430	EM55 014555	F\$AU = 000015
AX2.15= 002316	C\$BRK = 000022	DFPTBL 002226 G	EM56 014571	F\$AUTO= 000020
AX2.16= 002320	C\$BSEG= 000004	DH1 015013	EM57 014606	F\$BGN = 000040
AX3.15= 002322	C\$BSUB= 000002	DH2 015035	EM58 014623	F\$CLEA= 000007
AX3.16= 002324	C\$CEFG= 000045	DH3 015064	EM59 014637	F\$DU = 000016
AX315U= 000372	C\$CLCK= 000062	DH4 015122	EM60 014654	F\$END = 000041
A11 025426	C\$CLEA= 000012	DH5 015164	EM61 014670	F\$HARD= 000004
A12 033736	C\$CLOS= 000035	DH6 015167	EM62 014703	F\$HW = 000013
BADDAT 002372	C\$CLP1= 000006	DH7 015172	EM63 014720	F\$INIT= 000006
BAUDRT 037050	C\$CVEC= 000036	DH8 015224	EM64 014735	F\$JMP = 000050
BCC = 000001	C\$DCLN= 000044	DH9 015263	EM65 014752	F\$MOD = 000000
BCCCHK= 100000	C\$DODU= 000051	DIAGMC= 000000	EM66 014772	F\$MSG = 000011
BDRATE 002464	C\$DRPT= 000024	DISILO 002416	EM7 013276	F\$PROT= 000021
BIT0 = 000001 G	C\$DU = 000053	DISSI = 000040	EM8 013313	F\$PWR = 000017
BIT00 = 000001 G	C\$EDIT= 000003	DTR = 000100	EM9 013334	F\$RPT = 000012
BIT01 = 000002 G	C\$ERDF= 000055	EBLK = 000002	ENAX = 000004	F\$SEG = 000003
BIT02 = 000004 G	C\$ERHR= 000056	EF.CON= 000036 G	ENDIT 022512	F\$SOFT= 000005
BIT03 = 000010 G	C\$ERRO= 000060	EF.NEW= 000035 G	ENDPAT 002667	F\$SRV = 000010
BIT04 = 000020 G	C\$ERSF= 000054	EF.PWR= 000034 G	EOM = 000002	F\$SUB = 000002
BIT05 = 000040 G	C\$ERSO= 000057	EF.RES= 000037 G	ERRFLG 002342	F\$SW = 000014
BIT06 = 000100 G	C\$ESCA= 000010	EF.STA= 000040 G	ERROR1 002410	F\$TEST= 000001
BIT07 = 000200 G	C\$ESEG= 000005	EM10 013351	ERR1 015326 G	GETALL 004200
BIT08 = 000400 G	C\$ESUB= 000003	EM11 013372	ERR10 021356 G	GETPRM 022214
BIT09 = 001000 G	C\$ETST= 000001	EM12 013407	ERR2 015360 G	GETREG 003470
BIT1 = 000002 G	C\$EXIT= 000032	EM17 013430	ERR3 015666 G	GOAH = 000010
BIT10 = 002000 G	C\$GETB= 000026	EM18 013445	ERR4 016350 G	GOODAT 002370
BIT11 = 004000 G	C\$GETW= 000027	EM19 013466	ERR5 017026 G	G\$CNT0= 000200
BIT12 = 010000 G	C\$GMAN= 000043	EM2 013220	ERR6 017540 G	G\$DELM= 000372
BIT13 = 020000 G	C\$GPHR= 000042	EM20 013503	ERR7 020216 G	G\$DISP= 000003
BIT14 = 040000 G	C\$GPL0= 000030	EM21 013524	ERR8 020650 G	G\$EXCP= 000400
BIT15 = 100000 G	C\$GPRI= 000040	EM22 013541	EVL = 000004 G	G\$HILI= 000002
BIT2 = 000004 G	C\$INIT= 000011	EM28 013562	E\$END = 002100	G\$LOLI= 000001
BIT3 = 000010 G	C\$INLP= 000020	EM29 013603	E\$LOAD= 000035	G\$NO = 000000
BIT4 = 000020 G	C\$MANI= 000050	EM3 013257	FMT1 012256	G\$OFFS= 000400
BIT5 = 000040 G	C\$MEM = 000031	EM30 013620	FMT10 012521	G\$OF SI= 000376
BIT6 = 000100 G	C\$MSG = 000023	EM31 013641	FMT11 012552	G\$PRMA= 000001
BIT7 = 000200 G	C\$OPEN= 000034	EM34 013656	FMT12 012611	G\$PRMD= 000002
BIT8 = 000400 G	C\$PNTB= 000014	EM35 013704	FMT13 012655	G\$PRML= 000000
BIT9 = 001000 G	C\$PNTF= 000017	EM36 013724	FMT14 012721	G\$RADA= 000140
BOE = 000400 G	C\$PNTS= 000016	EM37 013740	FMT15 012765	G\$RADB= 000000
BPOLL = 000100	C\$PNTX= 000015	EM38 013761	FMT16 022514	G\$RADD= 000040
BSEL1 002440	C\$QIO = 000377	EM39 013776	FMT17 022636	G\$RADL= 000120

G\$RADO= 000020	LOT = 000010 G	L\$REPP 002062 G	L10056 031562	O\$DU = 000001
G\$XFER= 000004	LULOOP= 000010	L\$REV 002010 G	L10057 031744	O\$ERRT= 000000
G\$YES = 000010	LULP = 000040	L\$RPT 022010 G	L10060 032126	O\$GNSW= 000001
HDX = 000020	LUREG 002266	L\$SOFT 037306 G	L10061 032224	O\$POIN= 000001
HELP = 000001	LUR10 = 002266	L\$SPC 002056 G	L10062 032434	O\$SETU= 000000
HOE = 100000 G	LUR11 = 002270	L\$SPCP 002020 G	L10063 033742	PATA 002557
IACT = 000100	LUR12 = 002272	L\$SPTP 002024 G	L10064 032732	PATB 002603
IACTIV 006232	LUR13 = 002274	L\$STA 002030 G	L10065 033024	PATCH 037644
IBE = 010000 G	LUR14 = 002276	L\$SW 002256 G	L10066 033112	PATQ 002613
IC = 000200	LUR15 = 002300	L\$TEST 002114 G	L10067 033142	PATR 002623
ICIR = 000010	LUR16 = 002302	L\$TIML 002014 G	L10070 033260	PATS 002642
IDL = 000010	LUR17 = 002304	L\$UNIT 002012 G	L10071 033346	PATT 002662
IDLE = 000040	LUSW11 002454	L10000 002254	L10072 033426	PNT = 001000 G
IDU = 000040 G	LUSW12 002456	L10001 002266	L10073 033522	POLL = 000200
IER = 020000 G	LUSW13 002460	L10002 015356	L10074 033602	PRI = 002000 G
IERR = 000002	LU2MOD 002000 G	L10003 015664	L10075 033734	PRIOR 002334
INITRN 005212	L\$ACP 002110 G	L10004 016346	L10076 034132	PRIPTY 036537
INTFLG 002340	L\$APT 002036 G	L10005 017024	L10077 034322	PRI00 = 000000 G
INTGRL= 000010	L\$AU 023136 G	L10006 017536	L10100 034512	PRI01 = 000040 G
IRDY = 000020	L\$AUT 002070 G	L10007 020214	L10101 034702	PRI02 = 000100 G
ISIRDY 005746	L\$AUTO 022772 G	L10010 020646	L10102 035072	PRI03 = 000140 G
ISMANI 037342	L\$CCP 002106 G	L10011 021354	L10103 035256	PRI04 = 000200 G
ISPRNT 037430	L\$CLEA 023052 G	L10012 022006	L10104 035442	PRI05 = 000240 G
ISR = 000100 G	L\$CO 002032 G	L10013 022010	L10105 035554	PRI06 = 000300 G
ISWPAK 037513	L\$DEPO 002011 G	L10015 022512	L10106 036340	PRI07 = 000340 G
IXE = 004000 G	L\$DESC 003170 G	L10016 023050	L10107 036460	PRNFLG 002260
ISAU = 000041	L\$DESP 002076 G	L10017 023052	L10110 037342	PSTACK 002332
ISAUTO= 000041	L\$DEVP 002060 G	L10020 023102	MAINT1= 000010	RAB = 000004
ISCLN = 000041	L\$DISP 002124 G	L10021 023136	MAINT2= 000004	RABT = 000004
ISDU = 000041	L\$DLY 002116 G	L10022 023266	MCLK = 000002	RAX15 002354
ISHRD = 000041	L\$DTP 002040 G	L10023 023642	MCLR = 000100	RAX16 002356
ISINIT= 000041	L\$DTYP 002034 G	L10024 024200	MIFLAG 002256	RCVBUF 002762
ISMOD = 000041	L\$DU 023054 G	L10025 024262	MODINT 002422	RCV1ST 006752
ISMSG = 000041	L\$DUT 002072 G	L10026 024352	MODR = 000010	RDALL = 000004
ISPROT= 000040	L\$DVTY 003162 G	L10027 024546	MPCSR 002436	RDAX = 000020
ISPTAB= 000041	L\$EF 002052 G	L10030 025316	MPIVEC 002446	RDRXSI 006672
ISPWR = 000041	L\$ENVI 002044 G	L10031 025556	MPOVEC 002450	READAX 003576
ISRPT = 000041	L\$ETP 002102 G	L10032 025420	MPRIOR 002452	READLU 003344
ISSEG = 000041	L\$EXP1 002046 G	L10033 025540	MSG1 002670	READY = 000200
ISSETU= 000041	L\$EXP4 002064 G	L10034 025762	MSG2 002716	REDBYT 002350
ISSFT = 000041	L\$EXP5 002066 G	L10035 025660	MSG3 002732	REDDAT 002466
ISSRV = 000041	L\$HARD 036344 G	L10036 025760	MSTCLR 003276	REGNUM 002364
ISSUB = 000041	L\$HIME 002120 G	L10037 026116	MVIOX = 021000	REG0 002476
ISTST = 000041	L\$HPCP 002016 G	L10040 026570	MVIXO = 122000	REG1 002500
I422 = 000200	L\$HPTP 002022 G	L10041 026302	NEWST 022170	REG2 002502
JSJMP = 000167	L\$HW 002226 G	L10042 026434	OACT = 000100	REG3 002504
LDBYTS 010546	L\$ICP 002104 G	L10043 026566	OACTIV 005024	REG4 002506
LDMSG1 010630	L\$INIT 022020 G	L10044 026704	OC = 000200	REG5 002510
LDTXSI 004646	L\$LADP 002026 G	L10045 027242	OCOR = 000020	REG6 002512
LOADAT 002374	L\$LAST 037756 G	L10046 027652	OP = 000002	REG7 002514
LODATA 010156	L\$LOAD 002100 G	L10047 030324	ORDY = 000020	REOM = 000002
LODMSG 010470	L\$LUN 002074 G	L10050 030074	OSIRDY 004334	RERR = 000200
LODSIL 010720	L\$MREV 002050 G	L10051 030316	OVRR = 000010	RETADR 002346
LOE = 040000 G	L\$NAME 002000 G	L10052 031026	O\$APTS= 000000	RING = 000200
LOGDEV 002330	L\$PRIO 002042 G	L10053 030562	O\$AU = 000001	ROMI = 000002
LOOPBK 036733	L\$PROT 022012 G	L10054 031020	O\$BGNR= 000000	ROMO = 000004
LOOPIN 003546	L\$PRT 002112 G	L10055 031310	O\$BGNS= 000001	ROR = 000010

RRDYTO= 000001	STR = 000040	TXAB = 000004	T\$TSTM= 177777	T23.6 033262
RSEOM 006420	STRIP = 000010	TXABT = 002000	T\$TSTS= 000001	T23.7 033350
RSOM = 000001	SUBRPC 002336	TXCHAR 005574	T\$\$AU = 010021	T23.8 033430
RTS = 000040	SVCGBL= 000000	TXDATA= 000040	T\$\$AUT= 010016	T23.9 033524
RUN = 000200	SVCINS= 000001	TXEN = 000100	T\$\$CLE= 010017	T24 033744 G
RXABT = 002000	SVCSUB= 000001	TXEOM = 001000	T\$\$DU = 010020	T25 034134 G
RXBCC = 000400	SVCTAG= 000001	TXGA = 000010	T\$\$HAR= 010107	T26 034324 G
RXEBL = 001000	SVCTST= 000001	TXGOA = 004000	T\$\$HW = 010000	T27 034514 G
RXLENO= 000001	SWIFLG 002262	TXLENO= 000040	T\$\$INI= 010015	T28 034704 G
RXLEN1= 000002	SWPAC1 036570	TXLEN1= 000100	T\$\$MSG= 010012	T29 035074 G
RXLEN2= 000004	SWPAC2 036631	TXLEN2= 000200	T\$\$PRO= 010014	T3 023644 G
RXOVR = 004000	SWPAC3 036672	TXSOM = 000400	T\$\$RPT= 010013	T30 035260 G
RXWORD 002414	SW0 = 000002	TXWORD 002412	T\$\$SOF= 010110	T31 035444 G
RX0 = 000001	SW1 = 000004	TX0 = 000001	T\$\$SUB= 010075	T32 035556 G
RX1 = 000002	SW2 = 000010	TX1 = 000002	T\$\$SW = 010001	T4 024202 G
RX2 = 000004	SW3 = 000040	TX2 = 000004	T\$\$TES= 010106	T5 024264 G
RX3 = 000010	SYNCH = 000226	TX3 = 000010	T1 023140 G	T6 024354 G
RX4 = 000020	SYNO = 000001	TX4 = 000020	T10 025764 G	T7 024550 G
RX5 = 000040	SYN1 = 000002	TX5 = 000040	T11 026120 G	T8 025320 G
RX6 = 000100	SYN2 = 000004	TX6 = 000100	T11.1 026120	T8.1 025320
RX7 = 000200	SYN3 = 000010	TX7 = 000200	T11.2 026304	T8.2 025426
R14NRW 002546	SYN4 = 000020	TYPEY 022725	T11.3 026436	T9 025560 G
SAVE4 002404	SYN5 = 000040	T\$ARGC= 000002	T12 026572 G	T9.1 025560
SAVE6 002406	SYN6 = 000100	T\$CODE= 003032	T13 026706 G	T9.2 025662
SAVLEN 002424	SYN7 = 000200	T\$ERRN= 000101	T14 027244 G	UAM = 000200 G
SCRACH 002326	S\$LSYM= 010000	T\$EXCP= 000000	T15 027654 G	UNIT 002432
SEC = 000020	TCCHK= 100000	T\$FLAG= 000040	T15.1 027670	UNRR = 000001
SECA = 000020	TCOUNT 002264	T\$GMAN= 000000	T15.2 030112	UPBITS 002536
SELFR = 000040	TEOM = 000002	T\$HILI= 177777	T16 030326 G	VECTOR 036506
SELSBY= 000002	TERR = 000200	T\$LAST= 000001	T16.1 030342	V35 = 000020
SEL4 002442	TEST = 000001	T\$LOLI= 000000	T16.2 030600	WAIT50 004620
SEL6 002444	TESTMD= 000004	T\$LSYM= 010000	T17 031030 G	WAX = 000010
SETUP 010276	TIMCNT 037560	T\$LTNO= 000040	T18 031312 G	WAX15 002360
SFPTBL 002256 G	TIMFLG 002344	T\$NEST= 177777	T19 031564 G	WAX16 002362
SIGQ = 000100	TMP0 002516	T\$NS0 = 000000	T2 023270 G	WRDYTO= 000002
SIGR = 000200	TMP1 002520	T\$NS1 = 000005	T20 031746 G	WRIBYT 002352
SOM = 000001	TMP2 002522	T\$NS2 = 000002	T21 032130 G	WRITAX 003764
STALL 004636	TMP3 002524	T\$PTNU= 000000	T22 032226 G	WRITLU 003422
STARES 002402	TMP4 002526	T\$SAVL= 177777	T23 032436 G	XYZ = 000100
STARST 022160	TMP5 002530	T\$SEGL= 177777	T23.1 032460	X\$ALWA= 000000
STBY = 000002	TMP6 002532	T\$SUBN= 000000	T23.10 033604	X\$FALS= 000040
STEPLU= 000020	TMP7 002534	T\$TAGL= 177777	T23.2 032734	X\$OFFS= 000400
STEPMP= 000001	TSOM = 000001	T\$TAGN= 010111	T23.3 033026	X\$TRUE= 000020
STPCLK 003240	TSTCON 002462	T\$TEMP= 000000	T23.4 033114	\$LSTIN= 000001
STPERR 007164	TSTNUM 002434	T\$TEST= 000040	T23.5 033144	\$LSTTA= 000001
STPLU 004726				

. ABS. 037756 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 21865 WORDS (86 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 69 PAGES
CZDMSB.BIC,CZDMSB.SEQ/C/N:TOC=SVC34R.MLB,CZDMSB.P11

\$LSTIN	9-27#													
\$LSTTA	9-28#													
A11	26-538#	26-568												
A12	26-a75	26-B95#												
ABORT	14-61#													
ADDRES	18-102	18-108	18-123	18-143	18-161	18-182	18-200	18-219	18-238	27-16	27-27#			
ADR	14-20#													
ANBITS	15-113#													
APA	14-239#													
ASBC0	14-196#													
ASBC1	14-195#													
ASBC2	14-194#													
ASSEMB	9-18	9-18												
AX0.15	14-309#	17-230	18-132	18-151	18-171	18-190	18-208	18-228	18-246					
AX0.16	14-310#	18-132	18-151	18-171	18-190	18-208	18-228	18-246						
AX1	14-92#	14-164#												
AX1.15	14-311#	18-132	18-151	18-171	18-190	18-208	18-228	18-246						
AX1.16	14-312#	18-132	18-151	18-171	18-190	18-208	18-228	18-246						
AX2	14-91#	14-163#												
AX2.15	14-313#	18-134	18-153	18-173	18-192	18-210	18-230	18-248						
AX2.16	14-314#	18-134	18-153	18-173	18-192	18-210	18-230	18-248						
AX3.15	14-315#	18-134	18-153	18-173	18-192	18-210	18-230	18-248						
AX3.16	14-316#	18-134	18-153	18-173	18-192	18-210	18-230	18-248						
AX315U	14-259#													
AXNUM	15-33#	17-147	17-185	17-198	17-223	17-225	17-231*	17-237*	17-238	17-240*	17-242	17-433	17-441*	17-484*
	17-648	17-654*	17-683*	17-689*	17-966	17-968*	17-976*	17-987*	17-;14	17-;28*	17-;77*	17-;92	17-;95*	17-;08*
	17-;12*	26-403*	26-431*	26-434*	26-446*	26-815*	26-;06*	26-;52*	26-;28*	26-;81*	26-a24*			
BADDAT	15-;5#	17-820*	17-821*	17-842*	17-843*	18-111	18-126	18-165	18-222	26-352*	26-361*	26-468*	26-745*	26-769*
	26-;793*	26-;91*	26-;31*	26-a89*	26-a99*	26-A10*	26-A18*							
BAUDRT	27-23	27-35#												
BCC	1;-140#	17-846	17-848	17-854										
BCCCHK	14-324#	17-844	26-;64	26-;72	26-;17	26-;25	26-F16	26-F70						
BDRATE	15-67#	17-;48	17-;03	21-75*										
BIT0	14-20#	14-42	14-54	14-63	14-92	14-104	14-116	14-128	14-140	14-152	14-164	14-176	14-188	14-200
	14-212	14-221	14-233	14-246	14-258	14-269	14-342	17-225	17-266	17-395	17-575	17-616	17-656	21-45
	26-;70	26-;23												
BIT00	14-20	14-20#												
BIT01	14-20	14-20#												
BIT02	14-20	14-20#												
BIT03	14-20	14-20#												
BIT04	14-20	14-20#												
BIT05	14-20	14-20#												
BIT06	14-20	14-20#												
BIT07	14-20	14-20#												
BIT08	14-20	14-20#												
BIT09	14-20	14-20#												
BIT1	14-20#	14-41	14-53	14-62	14-81	14-91	14-103	14-115	14-127	14-139	14-151	14-163	14-175	14-187
	14-199	14-211	14-220	14-232	14-245	14-257	14-268	14-343	17-282	17-559	17-670			
BIT10	14-20#	14-279	14-291											
BIT11	14-20#	14-278	14-290											
BIT12	14-20#													
BIT13	14-20#													
BIT14	14-20#													
BIT15	14-20#	14-322	14-324	14-325	14-327	17-363	17-517							
BIT2	14-20#	14-40	14-52	14-61	14-80	14-90	14-102	14-114	14-126	14-138	14-150	14-162	14-174	14-186
	14-198	14-210	14-219	14-231	14-244	14-256	14-267							

EM52	17-<49	18-67#	26-B45															
EM53	17-<65	18-68#																
EM54	17-828	18-69#																
EM55	18-70#	26-A85																
EM56	18-71#	26-A37																
EM57	18-72#	26-A91																
EM58	18-73#	26-B26																
EM59	18-74#	26-B62																
EM60	18-75#	26-B79	26-G71	26-G83														
EM61	18-76#	26-B85																
EM62	18-77#	26-B91																
EM63	18-78#	26-A56																
EM64	18-79#	26-B09																
EM65	18-80#	26-H45																
EM66	18-81#	26->34																
EM7	17-272	18-33#																
EM8	17-278	18-34#																
EM9	17-288	18-35#																
ENAX	14-90#	14-162#	17-149	17-200														
ENDIT	21-35	21-98#																
ENDPAT	15-210#	26-567																
EOM	14-62#																	
ERR1	18-101#																	
ERR10	18-237#	26-458																
ERR2	18-107#	26-364	26-748	26-772	26-796	26-a92	26-A02											
ERR3	18-122#	26-471	26-94															
ERR4	17-272	17-278	17-288	17-294	17-401	17-407	17-565	17-571	17-581	17-587	17-622	17-628	17-828	17-<49				
	17-<65	18-141#																
ERR5	18-160#																	
ERR6	17-662	17-668	17-676	17-682	17-=01	17-=07	18-180#											
ERR7	18-199#	26-105	26-145	26-224	26-251	26-419	26-:37	26-:43	26-:83	26-:89	26-=72	26->34	26->49	26->99				
	26-?49	26-A37	26-A56	26-A85	26-A91	26-B09	26-B26	26-B45	26-B62	26-B79	26-B85	26-B91	26-C42	26-C96				
	26-D50	26-E04	26-E58	26-F11	26-F65	26-G71	26-G83	26-H45										
ERR8	17-833	17-852	17-858	17-867	17-873	17-882	17-888	17-897	17-903	18-217#								
ERRFLG	15-23#																	
ERROR1	15-42#	17-145*	17-158*	17-183*	17-209*	21-14*												
EVL	14-20#																	
F\$AU	9-18#	25-9	25-10															
F\$AUTO	9-18#	22-8	22-20															
F\$BGN	9-18#	9-24	18-101	18-107	18-122	18-141	18-160	18-180	18-199	18-217	18-237	19-9	20-8	21-8				
	22-8	23-8	24-8	25-9	26-20	26-46	26-65	26-147	26-173	26-253	26-267	26-284	26-299	26-317				
	26-332	26-366	26-384	26-480	26-506	26-510	26-510	26-533	26-539	26-539	26-565	26-570	26-589	26-593				
	26-593	26-616	26-620	26-620	26-643	26-644	26-663	26-695	26-722	26-726	26-726	26-750	26-754	26-754				
	26-774	26-778	26-778	26-798	26-799	26-812	26-827	26-850	26-944	26-969	26-:77	26-:02	26-:08	26-:08				
	26-:38	26-:44	26-:48	26-:54	26-:54	26-:84	26-:90	26-:94	26-:96	26-<24	26-<30	26-<30	26-<77	26-<83				
	26-<83	26-=30	26-=32	26-=48	26-=96	26->13	26->58	26->75	26-?08	26-?25	26-?58	26-?76	26-?94	26-?09				
	26-a42	26-a73	26-a82	26-a82	26-a93	26-A03	26-A21	26-A26	26-A26	26-A38	26-A40	26-A45	26-A45	26-A58				
	26-A63	26-A63	26-A68	26-A73	26-A73	26-A86	26-A93	26-A98	26-A98	26-B11	26-B16	26-B16	26-B28	26-B33				
	26-B33	26-B47	26-B52	26-B52	26-B64	26-B69	26-B69	26-B80	26-B86	26-B93	26-B97	26-C17	26-C51	26-C71				
	26-D05	26-D25	26-D59	26-D79	26-E13	26-E33	26-E67	26-E87	26-F21	26-F41	26-F75	26-F92	26-G18	26-G37				
	26-H72	27-14	28-13	28-45														
F\$CLEA	9-18#	23-8	23-11															
F\$DU	9-18#	24-8	24-13															
F\$END	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18	9-18				
	9-18	9-18	9-18#	9-24	18-103	18-116	18-135	18-154	18-174	18-193	18-211	18-231	18-249	19-11				
	21-99	22-20	23-11	24-13	25-10	26-20	26-20	26-20	26-46	26-46	26-65	26-65	26-65	26-147				
	26-147	26-173	26-173	26-173	26-253	26-253	26-267	26-267	26-267	26-284	26-284	26-299	26-299	26-299				

CROSS REFERENCE TABLE (CREF V01-05)

	26-317	26-317	26-332	26-332	26-332	26-366	26-366	26-384	26-384	26-384	26-480	26-480	26-506	26-506
	26-506	26-510	26-510	26-533	26-533	26-539	26-539	26-565	26-565	26-570	26-570	26-589	26-589	26-589
	26-593	26-593	26-616	26-616	26-620	26-620	26-643	26-643	26-644	26-644	26-663	26-663	26-663	26-695
	26-695	26-722	26-722	26-722	26-726	26-726	26-750	26-750	26-754	26-754	26-774	26-774	26-778	26-778
	26-798	26-798	26-799	26-799	26-812	26-812	26-812	26-827	26-827	26-850	26-850	26-850	26-944	26-944
	26-969	26-969	26-969	26-:77	26-:77	26-:02	26-:02	26-:02	26-:08	26-:08	26-:38	26-:44	26-:48	26-:48
	26-:54	26-:54	26-:84	26-:90	26-:94	26-:94	26-:96	26-:96	26-<24	26-<24	26-<24	26-<30	26-<30	26-<77
	26-<77	26-<83	26-<83	26-=30	26-=30	26-=32	26-=32	26-=48	26-=48	26-=48	26-=96	26-=96	26->13	26->13
	26->13	26->58	26->58	26->75	26->75	26->75	26-?08	26-?08	26-?25	26-?25	26-?25	26-?58	26-?58	26-?76
	26-?76	26-?76	26-?94	26-?94	26-a09	26-a09	26-a09	26-a42	26-a42	26-a73	26-a73	26-a73	26-a82	26-a82
	26-a93	26-A03	26-A21	26-A21	26-A26	26-A26	26-A38	26-A40	26-A40	26-A45	26-A45	26-A58	26-A58	26-A63
	26-A63	26-A68	26-A68	26-A73	26-A73	26-A86	26-A93	26-A93	26-A98	26-A98	26-B11	26-B11	26-B16	26-B16
	26-B28	26-B28	26-B33	26-B33	26-B47	26-B47	26-B52	26-B52	26-B64	26-B64	26-B69	26-B69	26-B80	26-B86
	26-B93	26-B93	26-B97	26-B97	26-C17	26-C17	26-C17	26-C51	26-C51	26-C71	26-C71	26-C71	26-D05	26-D05
	26-D25	26-D25	26-D25	26-D59	26-D59	26-D79	26-D79	26-D79	26-E13	26-E13	26-E33	26-E33	26-E33	26-E67
	26-E67	26-E87	26-E87	26-E87	26-F21	26-F21	26-F41	26-F41	26-F41	26-F75	26-F75	26-F92	26-F92	26-F92
	26-G18	26-G18	26-G37	26-G37	26-G37	26-H72	26-H72	27-25	28-20	28-45				
F\$HARD	9-18#	27-14	27-25											
F\$HW	9-18#	12-9	12-23											
F\$INIT	9-18#	21-8	21-99											
F\$JMP	9-18#													
F\$MOD	9-18#	9-24	28-45											
F\$MSG	9-18#	18-101	18-103	18-107	18-116	18-122	18-135	18-141	18-154	18-160	18-174	18-180	18-193	18-199
		18-211	18-217	18-231	18-237	18-249								
F\$PROT	9-18#	20-8	20-12											
F\$PWR	9-18#													
F\$RPT	9-18#	19-9	19-11											
F\$SEG	9-18#													
F\$SOFT	9-18#	28-13	28-20											
F\$SRV	9-18#													
F\$SUB	9-18#	26-510	26-533	26-539	26-565	26-593	26-616	26-620	26-643	26-726	26-750	26-754	26-774	26-778
		26-798	26-:08	26-:48	26-:54	26-:94	26-<30	26-<77	26-<83	26-=30	26-a82	26-A21	26-A26	26-A40
		26-A58	26-A63	26-A68	26-A73	26-A93	26-A98	26-B11	26-B16	26-B28	26-B33	26-B47	26-B52	26-B64
		26-B93												
F\$SW	9-18#	13-8	13-15											
F\$TEST	9-18#	26-20	26-46	26-65	26-147	26-173	26-253	26-267	26-284	26-299	26-317	26-332	26-366	26-384
		26-480	26-506	26-570	26-589	26-644	26-663	26-695	26-722	26-799	26-812	26-827	26-850	26-944
		26-:77	26-:02	26-:96	26-<24	26-=32	26-=48	26-=96	26->13	26->58	26->75	26-?08	26-?25	26-?58
		26-?94	26-a09	26-a42	26-a73	26-B97	26-C17	26-C51	26-C71	26-D05	26-D25	26-D59	26-D79	26-E13
		26-E67	26-E87	26-F21	26-F41	26-F75	26-F92	26-G18	26-G37	26-H72				
FMT1	18-9#	18-102	18-108	18-123	18-143	18-161	18-182	18-200	18-219	18-238				
FMT10	18-18#	18-142	18-181	18-218										
FMT11	18-19#	18-162												
FMT12	18-20#	26-739												
FMT13	18-21#	26-763												
FMT14	18-22#	26-787												
FMT15	18-23#	26-825												
FMT16	21-85	21-101#												
FMT17	21-90	21-103#												
FMT18	18-24#	26-737	26-823											
FMT19	17-:61	18-25#												
FMT2	18-10#	18-109	18-124	18-144	18-163	18-183	18-201	18-220	18-239					
FMT25	17-:56	18-26#												
FMT26	17-:70	18-27#												
FMT27	24-12	24-15#												
FMT3	18-11#	18-111	18-126	18-165	18-222									
FMT4	18-12#	18-112	18-127	18-131	18-146	18-150	18-166	18-170	18-185	18-189	18-203	18-207	18-223	18-227

L\$AU	10-17	25-9#	
L\$AUT	10-17#		
L\$AUTO	10-17	22-8#	
L\$CCP	10-17#		
L\$CLEA	10-17	23-8#	
L\$CO	10-17#		
L\$DEPO	10-17#		
L\$DESC	10-17	16-17#	
L\$DESP	10-17#		
L\$DEVP	10-17#		
L\$DISP	10-17	11-8#	
L\$DLY	10-17#		
L\$DTP	10-17#		
L\$DTYP	10-17#		
L\$DU	10-17	24-8#	
L\$DUT	10-17#		
L\$DVTY	10-17	16-12#	
L\$EF	10-17#		
L\$ENVI	10-17#		
L\$ETP	10-17#		
L\$EXP1	10-17#		
L\$EXP4	10-17#		
L\$EXP5	10-17#		
L\$HARD	10-17	27-14	27-14#
L\$HIME	10-17#		
L\$HPCP	10-17#		
L\$HPTP	10-17#		
L\$HW	10-17	12-9	12-9#
L\$IICP	10-17#		
L\$INIT	10-17	21-8#	
L\$LADP	10-17#		
L\$LAST	10-17	28-47#	
L\$LOAD	10-17#		
L\$LUN	10-17#		
L\$MREV	10-17#		
L\$NAME	10-17#		
L\$PRIO	10-17#		
L\$PROT	10-17	20-8#	
L\$PRT	10-17#		
L\$REPP	10-17#		
L\$REV	10-17#		
L\$RPT	19-9#		
L\$SOFT	10-17	28-13	28-13#
L\$SPC	10-17#		
L\$SPCP	10-17#		
L\$SPTP	10-17#		
L\$STA	10-17#		
L\$SW	10-17	13-8	13-8#
L\$TEST	10-17#		
L\$TIML	10-17#		
L\$UNIT	10-17#	21-50	
L10000	12-9	12-23#	
L10001	13-8	13-15#	
L10002	18-103#		
L10003	18-116#		
L10004	18-135#		
L10005	18-154#		

L10006	18-174#		
L10007	18-193#		
L10010	18-211#		
L10011	18-231#		
L10012	18-249#		
L10013	19-11#		
L10015	21-99#		
L10016	22-20#		
L10017	23-11#		
L10020	24-13#		
L10021	25-10#		
L10022	26-46#		
L10023	26-147#		
L10024	26-253#		
L10025	26-284#		
L10026	26-317#		
L10027	26-366#		
L10030	26-480#		
L10031	26-570#		
L10032	26-533#		
L10033	26-565#		
L10034	26-644#		
L10035	26-616#		
L10036	26-643#		
L10037	26-695#		
L10040	26-799#		
L10041	26-750#		
L10042	26-774#		
L10043	26-798#		
L10044	26-827#		
L10045	26-944#		
L10046	26-:77#		
L10047	26-:96#		
L10050	26-:38	26-:44	26-:48#
L10051	26-:84	26-:90	26-:94#
L10052	26-32#		
L10053	26-<77#		
L10054	26-30#		
L10055	26-96#		
L10056	26->58#		
L10057	26-?08#		
L10060	26-?58#		
L10061	26-?94#		
L10062	26-a42#		
L10063	26-B97#		
L10064	26-a93	26-A03	26-A21#
L10065	26-A38	26-A40#	
L10066	26-A58#		
L10067	26-A68#		
L10070	26-A86	26-A93#	
L10071	26-B11#		
L10072	26-B28#		
L10073	26-B47#		
L10074	26-B64#		
L10075	26-B80	26-B86	26-B93#
L10076	26-C51#		
L10077	26-D05#		

CROSS REFERENCE TABLE (CREF V01-05)

L10100	26-D59#													
L10101	26-E13#													
L10102	26-E67#													
L10103	26-F21#													
L10104	26-F75#													
L10105	26-G18#													
L10106	26-H72#													
L10107	27-14	27-25#												
L10110	28-13	28-20#												
LDBYTS	17-:25#	26-25	26-:13	26-:59	26-<38	26-<91	26-G45	26-G51						
LDMSG1	17-:50#	26-:60	26->37	26->87	26-?37									
LDTXSI	17-338#	17-451	17-452	17-510	17-943	17-950	17-951	17-:07	17-:33	17-:83	26-29	26-30		
LOADAT	15-36#	18-162												
LODATA	17-931#	26-C29	26-C83	26-D37	26-D91	26-E45	26-E99	26-F53						
LOMSGG	17-:00#	17-:55	26-181	26-205	26-233	26-272	26-668	26-861	26-891	26-922	26-980	26-:15	26-:50	26-?84
	26-G60													
LODSIL	17-934	17-:52	17-:79#	26-73	26-80	26-114	26-121	26-305	26-337	26-515	26-518	26-521	26-547	26-550
	26-553	26-598	26-601	26-604	26-625	26-628	26-631	26-:16	26-:62	26-<35	26-<41	26-<88	26-<94	26-F97
	26-G00	26-G48	26-G54	26-G57										
LOE	14-20#													
LOGDEV	15-17#	21-42*	21-49*	21-50	21-52	22-17	24-12							
LOOPBK	27-22	27-33#												
LOOPIN	17-125#													
LOT	14-20#													
LU2MOD	9-24#													
LULOOP	14-39#	17-38	17-229	17-367	17-460	17-982	17-984	17-:27	17-<79	26-814	26-:61	26->25	26->36	26->38
	26->88	26-?38	26-a94	26-A30	26-A79	26-B39	26-C31	26-C85	26-D39	26-D93	26-E47	26-F00	26-F54	
LULP	14-70#	14-135#	26-401											
LUR10	14-301#	17-101	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242				
LUR11	14-302#	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242					
LUR12	14-303#	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242					
LUR13	14-304#	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242					
LUR14	14-305#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244					
LUR15	14-306#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244					
LUR16	14-307#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244					
LUR17	14-308#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244					
LUREG	14-301	14-302	14-303	14-304	14-305	14-306	14-307	14-308	14-309	14-310	14-311	14-312	14-313	14-314
	14-315	14-316	15-11#											
LUSWI1	15-63#	21-71*	26-742	26-744										
LUSWI2	15-64#	21-72*	26-766	26-768										
LUSWI3	15-65#	21-73*	26-790	26-792										
MAINT1	14-79#	17-<38	26-B35											
MAINT2	14-80#	17-:78	17-<09	17-<54	26-A76									
MCLK	14-175#	26-A14												
MCLR	14-37#	17-32	17-33											
MIFLAG	13-10#	21-78												
MODINT	15-47#	17-:18*	17-:50*	17-:79*	17-:83*	17-:90*	17-<06*	26-C23	26-C77	26-D31	26-D85	26-E39	26-E93	26-F47
MODR	14-149#	26-:87	26-a86	26-a88	26-A87									
MPCSR	15-55#	18-102	18-108	18-123	18-143	18-161	18-182	18-200	18-219	18-238	21-59*	21-90	22-13	26-737
	26-823													
MPIVEC	15-60#	21-66*												
MPOVEC	15-61#	21-67*	21-68*											
MPRIOR	15-62#	21-69*												
MSG1	15-219#	17-:56	17-:58	17-:61	26-862	26-892	26-923	26-981	26-:16	26-:51	26-?85	26-G61		
MSG2	15-231#	26-669												
MSG3	15-238#	26-182	26-206	26-234	26-273									
MSTCLR	17-30#	17-436	26-45	26-139	26-146	26-245	26-252	26-283	26-316	26-365	26-390	26-479	26-569	26-694

CROSS REFERENCE TABLE (CREF V01-05)

RDRXSI	17-700#	17-811	26-100	26-140	26-219	26-246	26-353								
READAX	17-143#	17-232	17-655	17-:29	17-<96	26-448	26-816								
READLU	17-52#	17-103	17-153	17-161	17-165	17-204	17-265	17-281	17-394	17-463	17-475	17-558	17-574	17-615	
	17-702	17-706	17-745	17-823	17-<42	17-<58	26-411	26-729	26-757	26-781	26-:29	26-:75	26-=65	26-=86	
	26->27	26->42	26->92	26-?42	26-@85	26-@95	26-A06	26-A13	26-A32	26-A51	26-A80	26-B04	26-B21	26-B40	
	26-B57	26-B74	26-C35	26-C89	26-D43	26-D97	26-E51	26-F04	26-F58	26-G66	26-G78	26-H40			
READY	14-157#	17-154	17-205												
REDBYT	15-26#	17-64*	17-65*	17-104	17-154	17-162	17-166	17-205	17-268	17-274	17-284	17-290	17-397	17-403	
	17-464	17-476	17-561	17-567	17-577	17-583	17-618	17-624	17-703	17-707	17-746	17-824	17-<43	17-<59	
	26-412	26-730*	26-739	26-742	26-745	26-763	26-766	26-769	26-787	26-790	26-793	26-:33	26-:39	26-:79	
	26-:85	26-=66	26-=87*	26-=88	26-=91	26->28	26->31	26->43	26->93	26-?43	26-@86	26-@89	26-@96	26-@99	
	26-A07	26-A10	26-A14*	26-A15	26-A18	26-A33	26-A52	26-A81	26-A87	26-B05	26-B22	26-B41	26-B58	26-B75	
	26-B81	26-B87	26-C36	26-C90	26-D44	26-D98	26-E52	26-F05	26-F59	26-G67	26-G79	26-H41			
REDDAT	15-70#														
REG0	15-80#														
REG1	15-81#														
REG2	15-82#	21-91*	21-94	21-95											
REG3	15-83#														
REG4	15-84#														
REG5	15-85#														
REG6	15-86#														
REG7	15-87#														
REGNUM	15-32#	17-54	17-80	17-100	17-102*	17-106*	17-107	17-109*	17-144	17-146*	17-160*	17-164*	17-168*	17-182	
	17-184*	17-189*	17-193*	17-197*	17-210*	17-258	17-264*	17-280*	17-295*	17-301*	17-338	17-340*	17-343*	17-346*	
	17-387	17-393*	17-408*	17-414*	17-432	17-447*	17-459*	17-473*	17-485*	17-551	17-557*	17-573*	17-588*	17-594*	
	17-608	17-614*	17-629*	17-635*	17-700	17-701*	17-705*	17-708*	17-726	17-729*	17-761*	17-777	17-778*	17-789*	
	17-806	17-822*	17-830*	17-836*	17-911*	17-916*	17-967	17-972*	17-986*	17-:13	17-<35*	17-<40*	17-<56*	17-<68*	
	17-<78*	18-110	18-145	18-162	18-202	18-221	26-97*	26-138*	26-216*	26-244*	26-344*	26-356*	26-360*	26-396*	
	26-400*	26-408*	26-728*	26-756*	26-780*	26-:07*	26-:53*	26-<29*	26-<82*	26-=62*	26-=85*	26->26*	26->39*	26->89*	
	26-?39*	26-?78*	26-@17*	26-@20*	26-@84*	26-A12*	26-A31*	26-A48*	26-A50*	26-A65*	26-A75*	26-B01*	26-B03*	26-B19*	
	26-B36*	26-B38*	26-B55*	26-B73*	26-C32*	26-C86*	26-D40*	26-D94*	26-E48*	26-F01*	26-F55*	26-G65*	26-H34*	26-H37*	
REOM	14-199#	17-672	17-678	17-=03	26-449	26-467									
RERR	14-193#														
RE TADR	15-25#	17-297	17-410	17-590	17-631	17-685	17-913	17-:62	17-=10	26-21*	26-66*	26-174*	26-268*	26-300*	
	26-333*	26-385*	26-511*	26-540*	26-594*	26-621*	26-664*	26-854*	26-973*	26-:09*	26-:55*	26-<31*	26-<84*	26-=50*	
	26->15*	26->77*	26-?27*	26-?77*	26-@10*	26-@75*	26-C19*	26-C73*	26-D27*	26-D81*	26-E35*	26-E89*	26-F43*	26-F93*	
	26-G38*														
RING	14-145#	26-=87	26-@86	26-@88	26-A33										
ROMI	14-41#	17-16	17-18	17-19	17-126	17-128									
ROMO	14-40#	17-16	17-18	17-19	17-126	17-128									
ROR	14-197#														
RRDYTO	14-342#	17-145	17-158												
RSEOM	17-648#	17-739													
RSOM	14-200#	17-658	17-664												
RTS	14-147#	26-B75	26-G67	26-G79	26-H41										
RUN	14-36#	17-33	17-128	26-A29											
RX0	14-116#	14-188#													
RX1	14-115#	14-187#													
RX2	14-114#	14-186#													
RX3	14-113#	14-185#													
RX4	14-112#	14-184#													
RX5	14-111#	14-183#													
RX6	14-110#	14-182#													
RX7	14-109#	14-181#													
RXABT	14-291#	26-190	26-214	26-242	26-314										
RXBCC	14-293#	17-948	17-:63	26-<72	26-=25	26-E46	26-H01	26-H23	26-H63						
RXEBL	14-292#	26-41	26-190	26-214	26-242	26-281	26-314	26-562	26-640	26-C30	26-D38	26-D92			

26-:48	26-:48	26-:48	26-:54	26-:54	26-:54	26-:83	26-:83	26-:83	26-:83	26-:83	26-:83	26-:83	26-:83
26-:83	26-:83	26-:83	26-:83	26-:84	26-:84	26-:84	26-:84	26-:84	26-:84	26-:89	26-:89	26-:89	26-:89
26-:89	26-:89	26-:89	26-:89	26-:89	26-:89	26-:89	26-:89	26-:90	26-:90	26-:90	26-:90	26-:90	26-:90
26-:94	26-:94	26-:94	26-:96	26-:96	26-:96	26-<30	26-<30	26-<30	26-<77	26-<77	26-<77	26-<83	26-<83
26-<83	26-30	26-30	26-30	26-32	26-32	26-32	26-72	26-72	26-72	26-72	26-72	26-72	26-72
26-72	26-72	26-72	26-72	26-72	26-94	26-94	26-94	26-94	26-94	26-94	26-94	26-94	26-94
26-94	26-94	26-94	26-96	26-96	26-96	26->34	26->34	26->34	26->34	26->34	26->34	26->34	26->34
26->34	26->34	26->34	26->34	26->49	26->49	26->49	26->49	26->49	26->49	26->49	26->49	26->49	26->49
26->49	26->49	26->58	26->58	26->58	26->99	26->99	26->99	26->99	26->99	26->99	26->99	26->99	26->99
26->99	26->99	26->99	26-?08	26-?08	26-?08	26-?49	26-?49	26-?49	26-?49	26-?49	26-?49	26-?49	26-?49
26-?49	26-?49	26-?49	26-?49	26-?58	26-?58	26-?58	26-?94	26-?94	26-?94	26-a42	26-a42	26-a42	26-a82
26-a82	26-a82	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-a92
26-a92	26-a92	26-a92	26-a92	26-a92	26-a92	26-A02	26-A02	26-A02	26-A02	26-A02	26-A02	26-A02	26-A02
26-A02	26-A02	26-A02	26-A02	26-A03	26-A03	26-A03	26-A03	26-A03	26-A03	26-A21	26-A21	26-A21	26-A26
26-A26	26-A26	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37	26-A37
26-A37	26-A38	26-A38	26-A38	26-A38	26-A38	26-A40	26-A40	26-A40	26-A45	26-A45	26-A45	26-A56	26-A56
26-A56	26-A56	26-A56	26-A56	26-A56	26-A56	26-A56	26-A56	26-A56	26-A58	26-A58	26-A58	26-A58	26-A63
26-A63	26-A63	26-A68	26-A68	26-A68	26-A68	26-A73	26-A73	26-A73	26-A85	26-A85	26-A85	26-A85	26-A85
26-A85	26-A85	26-A85	26-A85	26-A85	26-A85	26-A86	26-A86	26-A86	26-A86	26-A86	26-A86	26-A91	26-A91
26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A91	26-A93	26-A98
26-A98	26-A98	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09	26-B09
26-B09	26-B11	26-B11	26-B11	26-B16	26-B16	26-B16	26-B26	26-B26	26-B26	26-B26	26-B26	26-B26	26-B26
26-B26	26-B26	26-B26	26-B26	26-B26	26-B28	26-B28	26-B28	26-B28	26-B33	26-B33	26-B33	26-B45	26-B45
26-B45	26-B45	26-B45	26-B45	26-B45	26-B45	26-B45	26-B45	26-B45	26-B47	26-B47	26-B47	26-B52	26-B52
26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B62	26-B64	26-B64
26-B64	26-B69	26-B69	26-B69	26-B79	26-B79	26-B79	26-B79	26-B79	26-B79	26-B79	26-B79	26-B79	26-B79
26-B79	26-B79	26-B80	26-B80	26-B80	26-B80	26-B80	26-B80	26-B80	26-B85	26-B85	26-B85	26-B85	26-B85
26-B85	26-B85	26-B85	26-B85	26-B85	26-B85	26-B85	26-B86	26-B86	26-B86	26-B86	26-B86	26-B86	26-B91
26-B91	26-B91	26-B91	26-B91	26-B91	26-B91	26-B91	26-B91	26-B91	26-B91	26-B91	26-B93	26-B93	26-B97
26-B97	26-B97	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42	26-C42
26-C42	26-C51	26-C51	26-C51	26-C96	26-C96	26-C96	26-C96	26-C96	26-C96	26-C96	26-C96	26-C96	26-C96
26-C51	26-C96	26-D05	26-D05	26-D05	26-D50	26-D50	26-D50	26-D50	26-D50	26-D50	26-D50	26-D50	26-D50
26-D05	26-D50	26-D50	26-D50	26-D50	26-D50	26-E04	26-E04	26-E04	26-E04	26-E04	26-E04	26-E04	26-E04
26-D50	26-E04	26-E04	26-E04	26-E13	26-E13	26-E13	26-E58	26-E58	26-E58	26-E58	26-E58	26-E58	26-E58
26-E04	26-E58	26-E58	26-E58	26-E58	26-E67	26-E67	26-E67	26-F11	26-F11	26-F11	26-F11	26-F11	26-F11
26-E58	26-F11	26-F11	26-F11	26-F11	26-F21	26-F21	26-F21	26-F21	26-F65	26-F65	26-F65	26-F65	26-F65
26-F11	26-F65	26-F65	26-F65	26-F65	26-F65	26-F65	26-F65	26-F65	26-G18	26-G18	26-G18	26-G71	26-G71
26-F65	26-G18	26-G18	26-G18	26-G18	26-G18	26-G18	26-G18	26-G18	26-G18	26-G18	26-G18	26-G83	26-G83
26-G18	26-G71	26-G71	26-G71	26-G71	26-G71	26-G71	26-G71	26-G71	26-G71	26-G83	26-G83	26-G83	26-G83
26-G71	26-G83	26-G83	26-G83	26-G83	26-G83	26-G83	26-G83	26-G83	26-H45	26-H45	26-H45	26-H45	26-H45
26-G83	26-H45	26-H45	26-H45	26-H45	26-H45	26-H45	26-H45	26-H45	26-H72	26-H72	26-H72	26-H72	26-H72
26-H45	26-H72	26-H72	26-H72	26-H72	26-H72	26-H72	26-H72	26-H72	27-14	27-14	27-14	27-16	27-16
26-H72	27-14	27-14	27-14	27-14	27-14	27-14	27-14	27-14	27-16	27-16	27-16	27-17	27-17
27-14	27-16	27-16	27-16	27-16	27-16	27-16	27-16	27-16	27-17	27-17	27-17	27-18	27-18
27-16	27-17	27-17	27-17	27-17	27-17	27-17	27-17	27-17	27-18	27-18	27-18	27-19	27-19
27-17	27-18	27-18	27-18	27-18	27-18	27-18	27-18	27-18	27-19	27-19	27-19	27-20	27-20
27-18	27-19	27-19	27-19	27-19	27-19	27-19	27-19	27-19	27-20	27-20	27-20	27-21	27-21
27-19	27-20	27-20	27-20	27-20	27-20	27-20	27-20	27-20	27-21	27-21	27-21	27-22	27-22
27-20	27-21	27-21	27-21	27-21	27-21	27-21	27-21	27-21	27-22	27-22	27-22	27-23	27-23
27-21	27-22	27-22	27-22	27-22	27-22	27-22	27-22	27-22	27-23	27-23	27-23	27-23	27-23
27-22	27-23	27-23	27-23	27-23	27-23	27-23	27-23	27-23	27-25	27-25	27-25	28-13	28-13
27-23	27-25	27-25	27-25	28-13	28-13	28-13	28-15	28-15	28-15	28-15	28-15	28-15	28-15
27-25	28-13	28-13	28-13	28-13	28-13	28-13	28-15	28-15	28-15	28-15	28-15	28-17	28-17
28-13	28-15	28-15	28-15	28-15	28-15	28-15	28-16	28-16	28-16	28-16	28-16	28-17	28-17
28-15	28-16	28-16	28-16	28-16	28-16	28-16	28-16	28-16	28-16	28-16	28-16	28-17	28-17
28-16	28-17	28-17	28-17	28-17	28-17	28-17	28-18	28-18	28-18	28-18	28-18	28-18	28-18
28-17	28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-18
28-18	28-18	28-18	28-18	28-18	28-18	28-18	28-20	28-20	28-20	28-20	28-20	28-20	28-20
28-18	28-20	28-20	28-20	28-20	28-20	28-20	28-20	28-20	28-47	28-47	28-47	28-47	28-47
28-20	28-47	28-47	28-47	28-47	28-47	28-47	28-47	28-47	28-47	28-47	28-47	28-47	28-47
SVC SUB 9-18#	9-31#	26-510	26-539	26-593	26-620	26-726	26-754	26-778	26-:08	26-:54	26-<30	26-<83	26-a82
26-A26	26-A45	26-A63	26-A73	26-A98	26-B16	26-B33	26-B52	26-B69					
SVC TAG 9-18#	9-33#	12-23	13-15	18-103	18-116	18-135	18-154	18-174	18-193	18-211	18-231	18-249	19-11
21-94	21-99	22-20	23-11	24-13	25-10	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-533

	26-565	26-570	26-616	26-643	26-644	26-695	26-750	26-774	26-798	26-799	26-827	26-944	26-:77	26-:48
	26-:94	26-:96	26-<77	26-=30	26-=32	26-=96	26->58	26-?08	26-?58	26-?94	26-a42	26-A21	26-A40	26-A58
	26-A68	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93	26-B97	26-C51	26-D05	26-D59	26-E13	26-E67	26-F21
	26-F75	26-G18	26-H72	27-25	28-20									
SVCTST	9-18#	9-30#	26-20	26-65	26-173	26-267	26-299	26-332	26-384	26-506	26-589	26-663	26-722	26-812
	26-850	26-969	26-:02	26-<24	26-=48	26->13	26->75	26-?25	26-?76	26-a09	26-a73	26-C17	26-C71	26-D25
	26-D79	26-E33	26-E87	26-F41	26-F92	26-G37								
SW0	14-127#													
SW1	14-126#													
SW2	14-125#													
SW3	14-123#													
SWIFLG	13-12#	26-740	26-764	26-788										
SWPAC1	27-19	27-30#												
SWPAC2	27-20	27-31#												
SWPAC3	27-21	27-32#												
SYNO	14-233#													
SYN1	14-232#													
SYN2	14-231#													
SYN3	14-230#													
SYN4	14-229#													
SYN5	14-228#													
SYN6	14-227#													
SYN7	14-226#													
SYNCH	14-234#	26-71	26-112	26-666	26-857	26-975	26-=56	26->21	26->83	26-?33	26-?80	26-a13	26-C79	26-E41
	26-G43													
TSSAU	25-9#	25-10												
TSSAUT	22-8#	22-20												
TSSCLE	23-8#	23-11												
TSSDU	24-8#	24-13												
TSSHAR	27-14	27-14#	27-25											
TSSHW	12-9	12-9#	12-23											
TSSINI	21-8#	21-99												
TSSMSG	18-101#	18-103	18-107#	18-116	18-122#	18-135	18-141#	18-154	18-160#	18-174	18-180#	18-193	18-199#	18-211
	18-217#	18-231	18-237#	18-249										
TSSPRO	20-8#													
TSSRPT	19-9#	19-11												
TSSSOF	28-13	28-13#	28-20											
TSSSUB	26-510#	26-533	26-539#	26-565	26-593#	26-616	26-620#	26-643	26-726#	26-750	26-754#	26-774	26-778#	26-798
	26-:08#	26-:38	26-:44	26-:48	26-:54#	26-:84	26-:90	26-:94	26-<30#	26-<77	26-<83#	26-=30	26-a82#	26-a93
	26-A03	26-A21	26-A26#	26-A38	26-A40	26-A45#	26-A58	26-A63#	26-A68	26-A73#	26-A86	26-A93	26-A98#	26-B11
	26-B16#	26-B28	26-B33#	26-B47	26-B52#	26-B64	26-B69#	26-B80	26-B86	26-B93				
TSSSW	13-8	13-8#	13-15											
TSSTES	26-20#	26-46	26-65#	26-147	26-173#	26-253	26-267#	26-284	26-299#	26-317	26-332#	26-366	26-384#	26-480
	26-506#	26-570	26-589#	26-644	26-663#	26-695	26-722#	26-799	26-812#	26-827	26-850#	26-944	26-969#	26-:77
	26-:02#	26-:96	26-<24#	26-=32	26-=48#	26-=96	26->13#	26->58	26->75#	26-?08	26-?25#	26-?58	26-?76#	26-?94
	26-a09#	26-a42	26-a73#	26-B97	26-C17#	26-C51	26-C71#	26-D05	26-D25#	26-D59	26-D79#	26-E13	26-E33#	26-E67
	26-E87#	26-F21	26-F41#	26-F75	26-F92#	26-G18	26-G37#	26-H72						
T\$ARGC	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	17-:56	17-:56	17-:56#	17-:61	17-:61	17-:61	17-:61#	17-:61#	17-:70	17-:70	17-:70#
	18-102	18-102	18-102	18-102	18-102#	18-102#	18-102#	18-108	18-108	18-108	18-108	18-108#	18-108#	18-108#
	18-109	18-109	18-109#	18-110	18-110	18-110	18-110	18-110#	18-110#	18-110#	18-110#	18-111	18-111	18-111
	18-111#	18-111#	18-111#	18-112	18-112	18-112	18-112	18-112#	18-112#	18-112#	18-112#	18-113	18-113	18-113
	18-113	18-113	18-113#	18-113#	18-113#	18-113#	18-113#	18-114	18-114	18-114	18-114#	18-114#	18-115	18-115
	18-115	18-115	18-115	18-115	18-115#	18-115#	18-115#	18-115#	18-115#	18-123	18-123	18-123	18-123	18-123#
	18-123#	18-123#	18-124	18-124	18-124#	18-125	18-125	18-125	18-125	18-125#	18-125#	18-125#	18-126	18-126
	18-126	18-126	18-126#	18-126#	18-126#	18-127	18-127	18-127	18-127	18-127#	18-127#	18-127#	18-128	18-128
	18-128	18-128	18-128	18-128	18-128#	18-128#	18-128#	18-128#	18-128#	18-129	18-129	18-129	18-129#	18-129#

18-130	18-130	18-130	18-130	18-130	18-130	18-130#	18-130#	18-130#	18-130#	18-130#	18-131	18-131	18-131
18-131	18-131#	18-131#	18-131#	18-132	18-132	18-132	18-132	18-132	18-132	18-132#	18-132#	18-132#	18-132#
18-132#	18-133	18-133	18-133	18-133#	18-133#	18-134	18-134	18-134	18-134	18-134	18-134	18-134#	18-134#
18-134#	18-134#	18-134#	18-142	18-142	18-142	18-142#	18-142#	18-143	18-143	18-143	18-143	18-143#	18-143#
18-143#	18-144	18-144	18-144#	18-145	18-145	18-145	18-145	18-145#	18-145#	18-145#	18-146	18-146	18-146
18-146	18-146#	18-146#	18-146#	18-147	18-147	18-147	18-147	18-147	18-147	18-147#	18-147#	18-147#	18-147#
18-147#	18-148	18-148	18-148	18-148#	18-148#	18-149	18-149	18-149	18-149	18-149	18-149	18-149#	18-149#
18-149#	18-149#	18-149#	18-150	18-150	18-150	18-150	18-150#	18-150#	18-150#	18-150#	18-151	18-151	18-151
18-151	18-151	18-151#	18-151#	18-151#	18-151#	18-151#	18-152	18-152	18-152	18-152#	18-152#	18-153	18-153
18-153	18-153	18-153	18-153	18-153#	18-153#	18-153#	18-153#	18-153#	18-153#	18-161	18-161	18-161	18-161#
18-161#	18-161#	18-162	18-162	18-162	18-162	18-162#	18-162#	18-162#	18-162#	18-163	18-163	18-163#	18-164
18-164	18-164	18-164#	18-164#	18-164#	18-165	18-165	18-165	18-165	18-165	18-165#	18-165#	18-166	18-166
18-166	18-166	18-166#	18-166#	18-166#	18-167	18-167	18-167	18-167	18-167	18-167	18-167#	18-167#	18-167#
18-167#	18-167#	18-168	18-168	18-168	18-168#	18-168#	18-169	18-169	18-169	18-169	18-169	18-169	18-169#
18-169#	18-169#	18-169#	18-169#	18-170	18-170	18-170	18-170#	18-170#	18-170#	18-170#	18-171	18-171	18-171
18-171	18-171	18-171#	18-171#	18-171#	18-171#	18-171#	18-172	18-172	18-172	18-172#	18-172#	18-173	18-173
18-173	18-173	18-173	18-173	18-173#	18-173#	18-173#	18-173#	18-173#	18-173#	18-181	18-181	18-181	18-181#
18-181#	18-182	18-182	18-182	18-182	18-182#	18-182#	18-183	18-183	18-183#	18-183#	18-184	18-184	18-184
18-184	18-184#	18-184#	18-184#	18-185	18-185	18-185	18-185	18-185#	18-185#	18-185#	18-186	18-186	18-186
18-186	18-186	18-186	18-186#	18-186#	18-186#	18-186#	18-187	18-187	18-187	18-187	18-187#	18-187#	18-188
18-188	18-188	18-188	18-188	18-188	18-188#	18-188#	18-188#	18-188#	18-188#	18-189	18-189	18-189	18-189
18-189#	18-189#	18-189#	18-190	18-190	18-190	18-190	18-190	18-190#	18-190#	18-190#	18-190#	18-190#	18-190#
18-191	18-191	18-191	18-191#	18-191#	18-192	18-192	18-192	18-192	18-192	18-192	18-192#	18-192#	18-192#
18-192#	18-192#	18-200	18-200	18-200	18-200	18-200#	18-200#	18-200#	18-200#	18-201	18-201	18-201#	18-202
18-202	18-202	18-202#	18-202#	18-202#	18-203	18-203	18-203	18-203	18-203#	18-203#	18-203#	18-204	18-204
18-204	18-204	18-204	18-204	18-204#	18-204#	18-204#	18-204#	18-204#	18-204#	18-205	18-205	18-205#	18-205#
18-206	18-206	18-206	18-206	18-206	18-206	18-206#	18-206#	18-206#	18-206#	18-206#	18-207	18-207	18-207
18-207	18-207#	18-207#	18-207#	18-208	18-208	18-208	18-208	18-208	18-208	18-208#	18-208#	18-208#	18-208#
18-208#	18-209	18-209	18-209	18-209#	18-209#	18-210	18-210	18-210	18-210	18-210	18-210#	18-210#	18-210#
18-210#	18-210#	18-210#	18-218	18-218	18-218	18-218#	18-218#	18-219	18-219	18-219	18-219	18-219#	18-219#
18-219#	18-220	18-220	18-220#	18-221	18-221	18-221	18-221	18-221#	18-221#	18-221#	18-222	18-222	18-222
18-222	18-222#	18-222#	18-222#	18-223	18-223	18-223	18-223	18-223#	18-223#	18-223#	18-224	18-224	18-224
18-224	18-224	18-224	18-224#	18-224#	18-224#	18-224#	18-224#	18-225	18-225	18-225	18-225#	18-225#	18-226
18-226	18-226	18-226	18-226	18-226	18-226#	18-226#	18-226#	18-226#	18-226#	18-227	18-227	18-227	18-227
18-227#	18-227#	18-227#	18-228	18-228	18-228	18-228	18-228	18-228	18-228#	18-228#	18-228#	18-228#	18-228#
18-229	18-229	18-229	18-229#	18-229#	18-230	18-230	18-230	18-230	18-230	18-230	18-230#	18-230#	18-230#
18-230#	18-230#	18-238	18-238	18-238	18-238	18-238#	18-238#	18-238#	18-239	18-239	18-239#	18-240	18-240
18-240	18-240	18-240#	18-240#	18-240#	18-241	18-241	18-241	18-241	18-241#	18-241#	18-241#	18-242	18-242
18-242	18-242	18-242	18-242	18-242#	18-242#	18-242#	18-242#	18-242#	18-243	18-243	18-243	18-243#	18-243#
18-244	18-244	18-244	18-244	18-244	18-244	18-244#	18-244#	18-244#	18-244#	18-244#	18-245	18-245	18-245
18-245	18-245#	18-245#	18-245#	18-246	18-246	18-246	18-246	18-246	18-246	18-246#	18-246#	18-246#	18-246#
18-246#	18-247	18-247	18-247	18-247#	18-247#	18-248	18-248	18-248	18-248	18-248	18-248	18-248#	18-248#
18-248#	18-248#	18-248#	21-85	21-85	21-85#	21-90	21-90	21-90	21-90#	21-90#	24-12	24-12	24-12
24-12#	24-12#	26-737	26-737	26-737	26-737#	26-737#	26-739	26-739	26-739	26-739#	26-739#	26-763	26-763
26-763	26-763#	26-763#	26-787	26-787	26-787	26-787#	26-787#	26-823	26-823	26-823	26-823#	26-823#	26-825
26-825	26-825	26-825#	26-825#										
TSCODE	21-94	21-94	21-94	21-94#	21-94#	27-16	27-16	27-16	27-16#	27-16#	27-16#	27-17	27-17
	27-17	27-17#	27-17#	27-17#	27-18	27-18	27-18#	27-18#	27-18#	27-19	27-19	27-19	27-19#
	27-19#	27-19#	27-20	27-20	27-20	27-20#	27-20#	27-21	27-21	27-21	27-21#	27-21#	27-21#
	27-22	27-22	27-22	27-22#	27-22#	27-23	27-23	27-23	27-23#	27-23#	27-23#	28-15	28-15
	28-15	28-15#	28-15#	28-15#	28-16	28-16	28-16#	28-16#	28-16#	28-17	28-17	28-17	28-17#
TERRN	28-17#	28-17#	28-18	28-18	28-18	28-18#	28-18#						
	9-18#	17-272	17-272#	17-278	17-278#	17-288	17-294	17-294#	17-401	17-401#	17-407	17-407#	17-565
	17-565#	17-571	17-571#	17-581	17-581#	17-587	17-622	17-622#	17-628	17-628#	17-662	17-662#	17-668
	17-668#	17-676	17-676#	17-682	17-682#	17-828	17-833	17-833#	17-852	17-852#	17-858	17-858#	17-867
	17-867#	17-873	17-873#	17-882	17-882#	17-888	17-888#	17-897	17-897#	17-903	17-903#	17-903#	17-903#
	17-<65#	17-01	17-01#	17-07	17-07#	26-105	26-105#	26-145	26-145#	26-224	26-224#	26-251	26-364

TSEXCP	26-364#	26-419	26-419#	26-458	26-458#	26-471	26-471#	26-748	26-748#	26-772	26-772#	26-796	26-796#	26-:37
T\$FLAG	26-:37#	26-:43	26-:43#	26-:83	26-:83#	26-:89	26-:89#	26-72	26-72#	26-94	26-94#	26->34	26->34#	26->49
	26->49#	26->99	26->99#	26-?49	26-?49#	26-a92	26-a92#	26-A02	26-A02#	26-A37	26-A37#	26-A56	26-A56#	26-A85
	26-A85#	26-A91	26-A91#	26-B09	26-B09#	26-B26	26-B26#	26-B45	26-B45#	26-B62	26-B62#	26-B79	26-B79#	26-B85
	26-B85#	26-B91	26-B91#	26-C42	26-C42#	26-C96	26-C96#	26-D50	26-D50#	26-E04	26-E04#	26-E58	26-E58#	26-F11
	26-F11#	26-F65	26-F65#	26-G71	26-G71#	26-G83	26-G83#	26-H45	26-H45#					
	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#	27-21	27-21#	27-22	27-22#
	27-23	27-23#	28-18	28-18#										
	26-:38	26-:38#	26-:38#	26-:44	26-:44#	26-:44#	26-:84	26-:84#	26-:84#	26-:90	26-:90#	26-:90#	26-a93	26-a93#
	26-a93#	26-A03	26-A03#	26-A03#	26-A38	26-A38#	26-A38#	26-A86	26-A86#	26-A86#	26-B80	26-B80#	26-B80#	26-B86
	26-B86#	26-B86#												
T\$GMAN	9-18#													
T\$HILI	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#	27-21	27-21#	27-22	27-22#
	27-23	27-23#	28-18	28-18#										
T\$LAST	9-18#	28-47#												
T\$LOLI	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#	27-21	27-21#	27-22	27-22#
	27-23	27-23#	28-18	28-18#										
T\$LSYM	9-18	9-18#	12-23	13-15	18-103	18-116	18-135	18-154	18-174	18-193	18-211	18-231	18-249	19-11
	21-99	22-20	23-11	24-13	25-10	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-533	26-565
	26-570	26-616	26-643	26-644	26-695	26-750	26-774	26-798	26-799	26-827	26-944	26-:77	26-:48	26-:94
	26-:96	26-<77	26-:30	26-:32	26-:96	26->58	26-?08	26-?58	26-?94	26-a42	26-A21	26-A40	26-A58	26-A68
	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93	26-B97	26-C51	26-D05	26-D59	26-E13	26-E67	26-F21	26-F75
	26-G18	26-H72	27-25	28-20										
T\$LTNO	28-47#													
T\$NEST	9-18#	9-24	9-24	9-24#	12-9	12-9	12-9#	12-23	12-23	12-23	12-23#	13-8	13-8	13-8#
	13-15	13-15	13-15	13-15#	18-101	18-101	18-101#	18-103	18-103	18-103	18-103#	18-107	18-107	18-107#
	18-116	18-116	18-116	18-116#	18-122	18-122	18-122#	18-135	18-135	18-135	18-135#	18-141	18-141	18-141#
	18-154	18-154	18-154	18-154#	18-160	18-160	18-160#	18-174	18-174	18-174	18-174#	18-180	18-180	18-180#
	18-193	18-193	18-193	18-193#	18-199	18-199	18-199#	18-211	18-211	18-211	18-211#	18-217	18-217	18-217#
	18-231	18-231	18-231	18-231#	18-237	18-237	18-237#	18-249	18-249	18-249	18-249#	19-9	19-9	19-9#
	19-11	19-11	19-11	19-11#	20-8	20-8	20-8#	20-12	20-12	20-12	20-12#	21-8	21-8	21-8#
	21-99	21-99	21-99	21-99#	22-8	22-8	22-8#	22-20	22-20	22-20	22-20#	23-8	23-8	23-8#
	23-11	23-11	23-11	23-11#	24-8	24-8	24-8#	24-13	24-13	24-13	24-13#	25-9	25-9	25-9#
	25-10	25-10	25-10	25-10#	26-20	26-20	26-20#	26-46	26-46	26-46	26-46#	26-65	26-65	26-65#
	26-147	26-147	26-147	26-147#	26-173	26-173	26-173#	26-253	26-253	26-253	26-253#	26-267	26-267	26-267#
	26-284	26-284	26-284	26-284#	26-299	26-299	26-299#	26-317	26-317	26-317	26-317#	26-332	26-332	26-332#
	26-366	26-366	26-366	26-366#	26-384	26-384	26-384#	26-480	26-480	26-480	26-480#	26-506	26-506	26-506#
	26-510	26-510	26-510#	26-533	26-533	26-533	26-533#	26-539	26-539	26-539	26-539#	26-565	26-565	26-565#
	26-570	26-570	26-570	26-570#	26-589	26-589	26-589#	26-593	26-593	26-593	26-593#	26-616	26-616	26-616#
	26-620	26-620	26-620#	26-643	26-643	26-643	26-643#	26-644	26-644	26-644	26-644#	26-663	26-663	26-663#
	26-695	26-695	26-695	26-695#	26-722	26-722	26-722#	26-726	26-726	26-726	26-726#	26-750	26-750	26-750#
	26-754	26-754	26-754#	26-774	26-774	26-774	26-774#	26-778	26-778	26-778	26-778#	26-798	26-798	26-798#
	26-799	26-799	26-799	26-799#	26-812	26-812	26-812#	26-827	26-827	26-827	26-827#	26-850	26-850	26-850#
	26-944	26-944	26-944	26-944#	26-969	26-969	26-969#	26-:77	26-:77	26-:77	26-:77#	26-:02	26-:02	26-:02#
	26-:08	26-:08	26-:08#	26-:48	26-:48	26-:48	26-:48#	26-:54	26-:54	26-:54	26-:54#	26-:94	26-:94	26-:94#
	26-:96	26-:96	26-:96	26-:96#	26-<24	26-<24	26-<24#	26-<30	26-<30	26-<30	26-<30#	26-<77	26-<77	26-<77#
	26-<83	26-<83	26-<83#	26-:30	26-:30	26-:30	26-:30#	26-:32	26-:32	26-:32	26-:32#	26-:48	26-:48	26-:48#
	26-:96	26-:96	26-:96	26-:96#	26->13	26->13	26->13#	26->58	26->58	26->58	26->58#	26->75	26->75	26->75#
	26-?08	26-?08	26-?08	26-?08#	26-?25	26-?25	26-?25#	26-?58	26-?58	26-?58	26-?58#	26-?76	26-?76	26-?76#
	26-?94	26-?94	26-?94	26-?94#	26-a09	26-a09	26-a09#	26-a42	26-a42	26-a42	26-a42#	26-a73	26-a73	26-a73#
	26-a82	26-a82	26-a82#	26-A21	26-A21	26-A21	26-A21#	26-A26	26-A26	26-A26	26-A26#	26-A40	26-A40	26-A40#
	26-A45	26-A45	26-A45#	26-A58	26-A58	26-A58	26-A58#	26-A63	26-A63	26-A63	26-A63#	26-A68	26-A68	26-A68#
	26-A73	26-A73	26-A73#	26-A93	26-A93	26-A93	26-A93#	26-A98	26-A98	26-A98	26-A98#	26-B11	26-B11	26-B11#
	26-B16	26-B16	26-B16#	26-B28	26-B28	26-B28	26-B28#	26-B33	26-B33	26-B33	26-B33#	26-B47	26-B47	26-B47#
	26-B52	26-B52	26-B52#	26-B64	26-B64	26-B64	26-B64#	26-B69	26-B69	26-B69	26-B69#	26-B93	26-B93	26-B93#
	26-B97	26-B97	26-B97	26-B97#	26-C17	26-C17	26-C17#	26-C51	26-C51	26-C51	26-C51#	26-C71	26-C71	26-C71#
	26-D05	26-D05	26-D05	26-D05#	26-D25	26-D25	26-D25#	26-D59	26-D59	26-D59	26-D59#	26-D79	26-D79	26-D79#

	26-E13	26-E13	26-E13	26-E13#	26-E33	26-E33	26-E33#	26-E67	26-E67	26-E67	26-E67#	26-E87	26-E87	26-E87#
	26-F21	26-F21	26-F21	26-F21#	26-F41	26-F41	26-F41#	26-F75	26-F75	26-F75	26-F75#	26-F92	26-F92	26-F92#
	26-G18	26-G18	26-G18	26-G18#	26-G37	26-G37	26-G37#	26-H72	26-H72	26-H72	26-H72#	27-14	27-14	27-14#
	27-25	27-25	27-25	27-25#	28-13	28-13	28-13#	28-20	28-20	28-20	28-20#	28-45	28-45	28-45#
	28-45#													
T\$NSO	9-24#	28-45												
T\$NS1	12-9#	12-23	13-8#	13-15	18-101#	18-103	18-107#	18-116	18-122#	18-135	18-141#	18-154	18-160#	18-174
	18-180#	18-193	18-199#	18-211	18-217#	18-231	18-237#	18-249	19-9#	19-11	20-8#	20-12	21-8#	21-99
	22-8#	22-20	23-8#	23-11	24-8#	24-13	25-9#	25-10	26-20#	26-46	26-65#	26-147	26-173#	26-253
	26-267#	26-284	26-299#	26-317	26-332#	26-366	26-384#	26-480	26-506#	26-570	26-589#	26-644	26-663#	26-695
	26-722#	26-799	26-812#	26-827	26-850#	26-944	26-969#	26-:77	26-:02#	26-:96	26-<24#	26-=32	26-=48#	26-=96
	26->13#	26->58	26->75#	26-?08	26-?25#	26-?58	26-?76#	26-?94	26-@09#	26-@42	26-@73#	26-B97	26-C17#	26-C51
	26-C71#	26-D05	26-D25#	26-D59	26-D79#	26-E13	26-E33#	26-E67	26-E87#	26-F21	26-F41#	26-F75	26-F92#	26-G18
	26-G37#	26-H72	27-14#	27-25	28-13#	28-20								
T\$NS2	26-510#	26-533	26-539#	26-565	26-593#	26-616	26-620#	26-643	26-726#	26-750	26-754#	26-774	26-778#	26-798
	26-:08#	26-:48	26-:54#	26-:94	26-<30#	26-<77	26-<83#	26-=30	26-@82#	26-A21	26-A26#	26-A40	26-A45#	26-A58
	26-A63#	26-A68	26-A73#	26-A93	26-A98#	26-B11	26-B16#	26-B28	26-B33#	26-B47	26-B52#	26-B64	26-B69#	26-B93
T\$PTNU	9-18#													
T\$SAVL	9-18#													
T\$SEGL	9-18#													
T\$SUBN	9-18#	26-20#	26-65#	26-173#	26-267#	26-299#	26-332#	26-384#	26-506#	26-510	26-510	26-510#	26-539	26-539
	26-539#	26-589#	26-593	26-593	26-593#	26-620	26-620	26-620#	26-663#	26-722#	26-726	26-726	26-726#	26-754
	26-754	26-754#	26-778	26-778	26-778#	26-812#	26-850#	26-969#	26-:02#	26-:08	26-:08	26-:08#	26-:54	26-:54
	26-:54#	26-<24#	26-<30	26-<30	26-<30#	26-<83	26-<83	26-<83#	26-<48#	26->13#	26->75#	26-?25#	26-?76#	26-@09#
	26-@73#	26-@82	26-@82	26-@82#	26-A26	26-A26	26-A26#	26-A45	26-A45	26-A45#	26-A63	26-A63	26-A63#	26-A73
	26-A73	26-A73#	26-A98	26-A98	26-A98#	26-B16	26-B16	26-B16#	26-B33	26-B33	26-B33#	26-B52	26-B52	26-B52#
	26-B69	26-B69	26-B69#	26-C17#	26-C71#	26-D25#	26-D79#	26-E33#	26-E87#	26-F41#	26-F92#	26-G37#		
T\$TAGL	9-18#													
T\$TAGN	9-18#	12-9	12-9	12-9#	13-8	13-8	13-8#	18-101	18-101	18-101#	18-107	18-107	18-107#	18-122
	18-122	18-122#	18-141	18-141	18-141#	18-160	18-160	18-160#	18-180	18-180	18-180#	18-199	18-199	18-199#
	18-217	18-217	18-217#	18-237	18-237	18-237#	19-9	19-9	19-9#	20-8	20-8	20-8#	21-8	21-8
	21-8#	22-8	22-8	22-8#	23-8	23-8	23-8#	24-8	24-8	24-8#	25-9	25-9	25-9#	26-20
	26-20	26-20#	26-65	26-65	26-65#	26-173	26-173	26-173#	26-267	26-267	26-267#	26-299	26-299	26-299#
	26-332	26-332	26-332#	26-384	26-384	26-384#	26-506	26-506	26-506#	26-510	26-510	26-510#	26-539	26-539
	26-539#	26-589	26-589	26-589#	26-593	26-593	26-593#	26-620	26-620	26-620#	26-663	26-663	26-663#	26-722
	26-722	26-722#	26-726	26-726	26-726#	26-754	26-754	26-754#	26-778	26-778	26-778#	26-812	26-812	26-812#
	26-850	26-850	26-850#	26-969	26-969	26-969#	26-:02	26-:02	26-:02#	26-:08	26-:08	26-:08#	26-:54	26-:54
	26-:54#	26-<24	26-<24	26-<24#	26-<30	26-<30	26-<30#	26-<83	26-<83	26-<83#	26=<48	26=<48	26=<48#	26->13
	26->13	26->13#	26->75	26->75	26->75#	26-?25	26-?25	26-?25#	26-?76	26-?76	26-?76#	26-@07	26-@09	26-@09#
	26-@73	26-@73	26-@73#	26-@82	26-@82	26-@82#	26-A26	26-A26	26-A26#	26-A45	26-A45	26-A45#	26-A63	26-A63
	26-A63#	26-A73	26-A73	26-A73#	26-A98	26-A98	26-A98#	26-B16	26-B16	26-B16#	26-B33	26-B33	26-B33#	26-B52
	26-B52	26-B52#	26-B69	26-B69	26-B69#	26-C17	26-C17	26-C17#	26-C71	26-C71	26-C71#	26-D25	26-D25	26-D25#
	26-D79	26-D79	26-D79#	26-E33	26-E33	26-F33#	26-E87	26-E87	26-E87#	26-F41	26-F41	26-F41#	26-F92	26-F92
	26-F92#	26-G37	26-G37	26-G37#	27-14	27-14	27-14#	28-13	28-13	28-13#				
T\$TEMP	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154	18-154#	18-174	18-174#	18-193
	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	20-12	20-12#	21-94	21-94	21-94
	21-94#	21-94#	21-94#	21-99	21-99#	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46
	26-46#	26-147	26-147#	26-253	26-253#	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533
	26-533#	26-565	26-565#	26-570	26-570#	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-750
	26-750#	26-774	26-774#	26-798	26-798#	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:38

TSOM	14-221#	26-405	26-a26												
TSTCON	15-66#	17-:21	17-:25	17-:46	17-:76	17-:88	17-<07	17-<23	17-<36	17-<52	21-74*	26-E34*	26-E88*	26-F42*	
TSTNUM	15-52#	17-:61	26-=49*	26->14*	26->76*	26-?26*	26-a74*	26-c18*	26-c72*	26-D26*	26-D80*	26-E34*	26-E88*	26-F42*	
TX0	14-54#	14-212#													
TX1	14-53#	14-211#													
TX2	14-52#	14-210#													
TX3	14-51#	14-209#													
TX4	14-50#	14-208#													
TX5	14-49#	14-207#													
TX6	14-48#	14-206#													
TX7	14-47#	14-205#													
TXAB	14-219#														
TXABT	14-279#	15-242													
TXCHAR	17-505#	26-977	26-:12	26-:47											
TXDATA	14-171#	26-:33	26-:39	26-:79	26-:85	26-A15	26-A17								
TXEN	14-86#	14-158#	26-304	26-397	26-399	26-a21	26-a23								
TXEOM	14-280#	15-226	15-227	15-228	15-229	15-235	15-236	17-949	26-28	26-522	26-554	26-605	26-632	26-:17	
	26-:63	26-<42	26-<95	26-G01	26-G49	26-G55									
TXGA	14-218#														
TXGOA	14-278#														
TXLENO	14-266#	17-812	26-475	26-:20	26-:66	26-<45	26-<98	26-E98	26-F52						
TXLEN1	14-265#	17-812	26-389	26-:20	26-:66	26-<45	26-<98	26-E98	26-F52						
TXLEN2	14-264#	17-812	26-:20	26-:66	26-<45	26-<98	26-E98	26-F52							
TXSOM	14-281#	15-219	15-220	15-231	15-232	15-243	15-244	17-443	17-935	17-:53	26-<36	26-<89	26-G58		
TXWORD	15-43#	17-339*	17-341	17-344	17-443*	17-444*	17-509*	17-940*	17-941*	17-942	17-949*	17-:06*	17-:31*	17-:32*	
	17-:80*	26-28*													
TYPEY	21-94	21-104#													
UAM	14-20#														
UNIT	15-51#														
UNRR	14-128#	17-824													
UPBITS	15-100#														
V35	14-254#	14-259	17-:18	17-:35	17-:79	17-:81	17-:83	17-<17	26-=54	26->18	26->81	26-?31			
VECTOR	27-17	27-28#													
WAIT50	17-311#	17-453	17-511	17-743	17-:10	17-:36	17-:86	26-=80							
WAX	14-89#	14-161#	17-200												
WAX15	15-30#	17-190*	17-191	17-442*	17-444	17-969*	17-978*	26-404*	26-:19*	26-:65*	26-<44*	26-<97*	26-a25*	26-a30*	
WAX16	15-31#	17-194*	17-195	17-445*	17-970*	17-980*	17-981	26-405*	26-424*	26-432*	26-435*	26-438*	26-444*	26-:20*	
	26-:66*	26-<45*	26-<46	26-<98*	26-<99	26-a26*	26-a31*								
WRDYTO	14-343#	17-183	17-209												
WRIBYT	15-27#	17-84*	17-85	17-147*	17-148*	17-149*	17-150*	17-185*	17-186*	17-187*	17-191*	17-195*	17-198*	17-199*	
	17-200*	17-201*	17-341*	17-344*	17-449*	17-779*	17-780*	17-786*	17-974*	17-:78*	17-<09*	17-<34*	17-<38*	17-<54*	
	17-<69*	26-98*	26-217*	26-345*	26-397*	26-401*	26-?82*	26-a18*	26-a21*	26-A04*	26-A47*	26-A66*	26-A76*	26-B00*	
	26-B18*	26-B35*	26-B54*	26-H35*											
WRITAX	17-181#	17-446	17-971	17-983	26-406	26-421	26-425	26-428	26-433	26-439	26-445	26-:21	26-:67	26-<47	
	26-=00	26-a27	26-a32	26-a35											
WRITLU	17-78#	17-151	17-188	17-192	17-196	17-202	17-342	17-345	17-450	17-781	17-787	17-975	17-<39	17-<55	
	17-<70	26-99	26-218	26-346	26-398	26-402	26-?83	26-a19	26-a22	26-A05	26-A49	26-A67	26-A77	26-B02	
	26-B20	26-B37	26-B56	26-H36											
X\$ALWA	9-18#														
X\$FALS	9-18#														
X\$OFFS	9-18#														
X\$TRUE	9-18#														
XYZ	14-252#	14-259	17-:18	17-:40	17-<00	17-<06	17-<26	26-=54	26->19	26->80	26-?31				

ENDSFT	1-568#	9-18#	28-20											
ENDSRV	1-580#	9-18#												
ENDSUB	1-596#	9-18#	26-533	26-565	26-616	26-643	26-750	26-774	26-798	26-:48	26-:94	26-<77	26-=30	26-A21
	26-A40	26-A58	26-A68	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93					
ENDSW	1-614#	9-18#	13-15											
ENDTST	1-624#	9-18#	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-570	26-644	26-695	26-799	26-827
	26-944	26-:77	26-:96	26-=32	26-=96	26->58	26-?08	26-?58	26-?94	26-a42	26-B97	26-C51	26-D05	26-D59
	26-E13	26-E67	26-F21	26-F75	26-G18	26-H72								
EQUALS	1-642#	9-18#	14-20											
ERRDF	1-714#	9-18#	17-272	17-278	17-288	17-294	17-401	17-407	17-565	17-571	17-581	17-587	17-622	17-628
	17-662	17-668	17-676	17-682	17-828	17-833	17-852	17-858	17-867	17-873	17-882	17-888	17-897	17-903
	17-<49	17-<65	17-=01	17-=07	26-105	26-145	26-224	26-251	26-364	26-419	26-458	26-471	26-748	26-772
	26-796	26-:37	26-:43	26-:83	26-:89	26-=72	26-=94	26->34	26->49	26->99	26-?49	26-a92	26-A02	26-A37
	26-A56	26-A85	26-A91	26-B09	26-B26	26-B45	26-B62	26-B79	26-B85	26-B91	26-C42	26-C96	26-D50	26-E04
	26-E58	26-F11	26-F65	26-G71	26-G83	26-H45								
ERRHRD	1-718#	9-18#												
ERROR	1-722#	9-18#												
ERRSF	1-726#	9-18#												
ERRSOF	1-730#	9-18#												
ERRTBL	1-734#	9-18#												
ESCAPE	1-744#	9-18#	26-:38	26-:44	26-:84	26-:90	26-a93	26-A03	26-A38	26-A86	26-B80	26-B86		
EXIT	1-771#	9-18#												
FEQUAL	1-810#	9-18#												
GETBYT	1-824#	9-18#												
GETPRI	1-834#	9-18#												
GETWOR	1-829#	9-18#												
GMANIA	1-839#	9-18#												
GMANID	1-848#	9-18#												
GMANIL	1-859#	9-18#	21-94											
GPHARD	1-868#	9-18#	21-52											
GPRMA	1-874#	9-18#	27-16	27-17										
GPRMD	1-903#	9-18#	27-18	27-19	27-20	27-21	27-22	27-23	28-18					
GPRML	1-934#	9-18#	21-94	21-94#	28-15	28-16	28-17							
HEADER	1-954#	9-18#	10-17											
INLOOP	1-962#	9-18#												
IOSETU	1-966#	9-18#												
IOSTAR	1-974#	9-18#												
KT11	1-982#	9-18#												
LASTAD	1-:47#	9-18#	28-47											
MSBYTE	1-D00#	9-18#	10-17	10-17	10-17	10-17#								
MSCHEC	1-E18#	9-18#												
MSCNTO	1-E82#	9-18#	21-94	21-94#	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#
	27-21	27-21#	27-22	27-22#	27-23	27-23#	28-15	28-15#	28-16	28-16#	28-17	28-17#	28-18	28-18#
MSCOUN	1-D66#	9-18#	17-:56	17-:56#	17-:61	17-:61#	17-:70	17-:70#	18-102	18-102#	18-102#	18-108	18-108	18-108#
	18-109	18-109#	18-110	18-110	18-110#	18-111	18-111	18-111#	18-112	18-112#	18-112#	18-113	18-113	18-113#
	18-113	18-113#	18-114	18-114#	18-115	18-115	18-115	18-115#	18-115	18-115#	18-123	18-123#	18-124	18-124#
	18-125	18-125	18-125#	18-126	18-126	18-126#	18-127	18-127#	18-127	18-127#	18-128	18-128	18-128	18-128#
	18-129	18-129#	18-130	18-130	18-130	18-130#	18-131	18-131#	18-131	18-131#	18-132	18-132	18-132	18-132#
	18-132#	18-133	18-133#	18-134	18-134	18-134#	18-134	18-134#	18-142	18-142#	18-143	18-143	18-143#	18-144
	18-144#	18-145	18-145	18-145#	18-146	18-146#	18-146#	18-147	18-147	18-147#	18-147	18-147#	18-148	18-148#
	18-149	18-149	18-149	18-149	18-149#	18-150	18-150	18-150#	18-151	18-151#	18-151	18-151#	18-151#	18-152
	18-152#	18-153	18-153	18-153	18-153	18-153#	18-161	18-161#	18-161#	18-162	18-162	18-162#	18-163	18-163#
	18-164	18-164	18-164#	18-165	18-165	18-165#	18-166	18-166	18-166#	18-167	18-167	18-167	18-167	18-167#
	18-168	18-168#	18-169	18-169	18-169	18-169	18-169#	18-170	18-170	18-170#	18-171	18-171	18-171	18-171
	18-171#	18-172	18-172#	18-173	18-173	18-173	18-173	18-173#	18-181	18-181#	18-182	18-182	18-182#	18-183
	18-183#	18-184	18-184	18-184#	18-185	18-185	18-185#	18-186	18-186	18-186	18-186	18-186#	18-187	18-187#
	18-188	18-188	18-188	18-188	18-188#	18-189	18-189	18-189#	18-190	18-190	18-190	18-190	18-190#	18-191

	26-:08#	26-:08#	26-:08#	26-:37#	26-:38#	26-:43#	26-:44#	26-:48#	26-:54	26-:54	26-:54	26-:54#	26-:54#	26-:54#
	26-:83#	26-:84#	26-:89#	26-:90#	26-:94#	26-:96#	26-<24	26-<24	26-<24	26-<24#	26-<24#	26-<24#	26-<30	26-<30
	26-<30	26-<30#	26-<30#	26-<30#	26-<77#	26-<83	26-<83	26-<83	26-<83#	26-<83#	26-<83#	26-<83#	26-<30#	26-<32#
	26-=48	26-=48	26-=48#	26-=48#	26-=48#	26-=72#	26-=94#	26-=96#	26->13	26->13	26->13	26->13#	26->13#	26->13#
	26->34#	26->49#	26->58#	26->75	26->75	26->75	26->75#	26->75#	26->75#	26->99#	26-?08#	26-?25	26-?25	26-?25
	26-?25#	26-?25#	26-?25#	26-?49#	26-?58#	26-?76	26-?76	26-?76	26-?76#	26-?76#	26-?76#	26-?94#	26-a09	26-a09
	26-a09	26-a09#	26-a09#	26-a09#	26-a42#	26-a73	26-a73	26-a73	26-a73#	26-a73#	26-a73#	26-a82	26-a82	26-a82
	26-a82#	26-a82#	26-a82#	26-a92#	26-a93#	26-A02#	26-A03#	26-A21#	26-A26	26-A26	26-A26	26-A26#	26-A26#	26-A26#
	26-A37#	26-A38#	26-A40#	26-A45	26-A45	26-A45	26-A45#	26-A45#	26-A45#	26-A56#	26-A58#	26-A63	26-A63	26-A63
	26-A63#	26-A63#	26-A63#	26-A68#	26-A73	26-A73	26-A73	26-A73#	26-A73#	26-A73#	26-A85#	26-A86#	26-A91#	26-A93#
	26-A98	26-A98	26-A98	26-A98#	26-A98#	26-A98#	26-B09#	26-B11#	26-B16	26-B16	26-B16	26-B16#	26-B16#	26-B16#
	26-B26#	26-B28#	26-B33	26-B33	26-B33	26-B33#	26-B33#	26-B33#	26-B45#	26-B47#	26-B52	26-B52	26-B52	26-B52#
	26-B52#	26-B52#	26-B62#	26-B64#	26-B69	26-B69	26-B69	26-B69#	26-B69#	26-B69#	26-B79#	26-B80#	26-B85#	26-B86#
	26-B91#	26-B93#	26-B97#	26-C17	26-C17	26-C17#	26-C17#	26-C17#	26-C17#	26-C42#	26-C51#	26-C71	26-C71	26-C71
	26-C71#	26-C71#	26-C71#	26-C96#	26-D05#	26-D25	26-D25	26-D25	26-D25#	26-D25#	26-D25#	26-D50#	26-D59#	26-D79
	26-D79	26-D79	26-D79#	26-D79#	26-E04#	26-E13#	26-E33	26-E33	26-E33	26-E33	26-E33#	26-E33#	26-E33#	26-E58#
	26-E67#	26-E87	26-E87	26-E87	26-E87#	26-E87#	26-F11#	26-F21#	26-F41	26-F41	26-F41	26-F41#	26-F41#	26-F41#
	26-F41#	26-F65#	26-F75#	26-F92	26-F92	26-F92	26-F92#	26-F92#	26-F92#	26-G18#	26-G37	26-G37	26-G37	26-G37#
	26-G37#	26-G37#	26-G71#	26-G83#	26-H45#	26-H72#	27-14	27-14	27-14#	27-14#	27-14#	28-13	28-13	28-13#
MSIOSE	1-A00#	9-18#												
MSLDRO	1-C42#	9-18#	21-25	21-25#	21-28	21-28#	21-31	21-31#	21-34	21-34#	21-52	21-52#	22-10	22-10#
	22-17	22-17#												
MSMASK	1-a71#	9-18#												
MSMCHI	1-4#	9-18	9-18#	9-18#										
MSMCLO	1-a24#	9-18	9-18#	9-18#										
MSMSK1	1-a77#	9-18#												
MSPOP	1-B81#	9-18#	12-23	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154	18-154#
	18-174	18-174#	18-193	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	20-12	20-12#
	21-99	21-99#	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-147	26-147#
	26-253	26-253#	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533	26-533#	26-565	26-565#
	26-570	26-570#	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-750	26-750#	26-774	26-774#
	26-798	26-798#	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:48	26-:48#	26-:94	26-:94#
	26-:96	26-:96#	26-<77	26-<77#	26-=30	26-=30#	26-=32	26-=32#	26-=96	26-=96#	26->58	26->58#	26-?08	26-?08#
	26-?58	26-?58#	26-?94	26-?94#	26-a42	26-a42#	26-A21	26-A21#	26-A40	26-A40#	26-A58	26-A58#	26-A68	26-A68#
	26-A93	26-A93#	26-B11	26-B11#	26-B28	26-B28#	26-B47	26-B47#	26-B64	26-B64#	26-B93	26-B93#	26-B97	26-B97#
	26-C51	26-C51#	26-D05	26-D05#	26-D59	26-D59#	26-E13	26-E13#	26-E67	26-E67#	26-F21	26-F21#	26-F75	26-F75#
	26-G18	26-G18#	26-H72	26-H72#	27-25	27-25#	28-20	28-20#	28-45	28-45#				
MSPRIN	1-a36#	9-18#	17-:56	17-:56#	17-:61	17-:61#	17-:70	17-:70#	18-102	18-102#	18-108	18-108#	18-109	18-109#
	18-110	18-110#	18-111	18-111#	18-112	18-112#	18-113	18-113#	18-114	18-114#	18-115	18-115#	18-123	18-123#
	18-124	18-124#	18-125	18-125#	18-126	18-126#	18-127	18-127#	18-128	18-128#	18-129	18-129#	18-130	18-130#
	18-131	18-131#	18-132	18-132#	18-133	18-133#	18-134	18-134#	18-142	18-142#	18-143	18-143#	18-144	18-144#
	18-145	18-145#	18-146	18-146#	18-147	18-147#	18-148	18-148#	18-149	18-149#	18-150	18-150#	18-151	18-151#
	18-152	18-152#	18-153	18-153#	18-161	18-161#	18-162	18-162#	18-163	18-163#	18-164	18-164#	18-165	18-165#
	18-166	18-166#	18-167	18-167#	18-168	18-168#	18-169	18-169#	18-170	18-170#	18-171	18-171#	18-172	18-172#
	18-173	18-173#	18-181	18-181#	18-182	18-182#	18-183	18-183#	18-184	18-184#	18-185	18-185#	18-186	18-186#
	18-187	18-187#	18-188	18-188#	18-189	18-189#	18-190	18-190#	18-191	18-191#	18-192	18-192#	18-200	18-200#
	18-201	18-201#	18-202	18-202#	18-203	18-203#	18-204	18-204#	18-205	18-205#	18-206	18-206#	18-207	18-207#
	18-208	18-208#	18-209	18-209#	18-210	18-210#	18-218	18-218#	18-219	18-219#	18-220	18-220#	18-221	18-221#
	18-222	18-222#	18-223	18-223#	18-224	18-224#	18-225	18-225#	18-226	18-226#	18-227	18-227#	18-228	18-228#
	18-229	18-229#	18-230	18-230#	18-238	18-238#	18-239	18-239#	18-240	18-240#	18-241	18-241#	18-242	18-242#
	18-243	18-243#	18-244	18-244#	18-245	18-245#	18-246	18-246#	18-247	18-247#	18-248	18-248#	21-85	21-85#
	21-90	21-90#	24-12	24-12#	26-737	26-737#	26-739	26-739#	26-763	26-763#	26-787	26-787#	26-823	26-823#
	26-825	26-825#												
MSPUSH	1-a31#	9-18#	9-24	9-24#	12-9	12-9#	13-8	13-8#	18-101	18-101#	18-107	18-107#	18-122	18-122#
	18-141	18-141#	18-160	18-160#	18-180	18-180#	18-199	18-199#	18-217	18-217#	18-237	18-237#	19-9	19-9#
	20-8	20-8#	21-8	21-8#	22-8	22-8#	23-8	23-8#	24-8	24-8#	25-9	25-9#	26-20	26-20#
	26-65	26-65#	26-173	26-173#	26-267	26-267#	26-299	26-299#	26-332	26-332#	26-384	26-384#	26-506	26-506#

J 1

CROSS REFERENCE TABLE (CREF V01-05)

	26-510	26-510#	26-539	26-539#	26-589	26-589#	26-593	26-593#	26-620	26-620#	26-663	26-663#	26-722	26-722#
	26-726	26-726#	26-754	26-754#	26-778	26-778#	26-812	26-812#	26-850	26-850#	26-969	26-969#	26-:02	26-:02#
	26-:08	26-:08#	26-:54	26-:54#	26-<24	26-<24#	26-<30	26-<30#	26-<83	26-<83#	26-48	26-48#	26->13	26->13#
	26->75	26->75#	26-?25	26-?25#	26-?76	26-?76#	26-a09	26-a09#	26-a73	26-a73#	26-a82	26-a82#	26-A26	26-A26#
	26-A45	26-A45#	26-A63	26-A63#	26-A73	26-A73#	26-A98	26-A98#	26-B16	26-B16#	26-B33	26-B33#	26-B52	26-B52#
	26-B69	26-B69#	26-C17	26-C17#	26-C71	26-C71#	26-D25	26-D25#	26-D79	26-D79#	26-E33	26-E33#	26-E87	26-E87#
	26-F41	26-F41#	26-F92	26-F92#	26-G37	26-G37#	27-14	27-14#	28-13	28-13#				
MSPUT	1-C72#	9-18#	17-:56	17-:56#	17-:56#	17-:61	17-:61	17-:61	17-:61#	17-:70	17-:70	17-:70#	18-102	18-102
	18-102	18-102	18-102#	18-108	18-108	18-108	18-108	18-108#	18-109	18-109	18-109#	18-110	18-110	18-110
	18-110	18-110#	18-111	18-111	18-111	18-111	18-111#	18-112	18-112	18-112	18-112#	18-113	18-113	18-113
	18-113	18-113	18-113	18-113	18-113#	18-114	18-114	18-114#	18-115	18-115	18-115#	18-115	18-115	18-115
	18-115	18-115#	18-123	18-123	18-123	18-123	18-123#	18-124	18-124	18-124#	18-125	18-125	18-125	18-125
	18-125#	18-126	18-126	18-126	18-126	18-126#	18-127	18-127	18-127	18-127#	18-128	18-128	18-128	18-128
	18-128	18-128	18-128	18-128#	18-129	18-129	18-129#	18-130	18-130	18-130	18-130	18-130	18-130	18-130
	18-130#	18-131	18-131	18-131	18-131	18-131#	18-132	18-132	18-132	18-132	18-132	18-132#	18-133	18-133
	18-133	18-133	18-133#	18-134	18-134	18-134	18-134	18-134#	18-142	18-142	18-142	18-142#	18-142#	18-142#
	18-143	18-143	18-143	18-143	18-143#	18-144	18-144	18-144#	18-145	18-145	18-145	18-145#	18-146	18-146
	18-146	18-146	18-146	18-146#	18-147	18-147	18-147	18-147#	18-148	18-148	18-148	18-148#	18-148	18-148
	18-148#	18-149	18-149	18-149	18-149	18-149	18-149#	18-150	18-150	18-150	18-150	18-150#	18-151	18-151
	18-151	18-151	18-151	18-151	18-151	18-151#	18-152	18-152	18-152	18-152#	18-153	18-153	18-153	18-153
	18-153	18-153	18-153#	18-161	18-161	18-161	18-161	18-161#	18-162	18-162	18-162	18-162#	18-163	18-163
	18-163	18-163#	18-164	18-164	18-164	18-164	18-164#	18-165	18-165	18-165	18-165#	18-166	18-166	18-166
	18-166	18-166#	18-166#	18-167	18-167	18-167	18-167	18-167#	18-168	18-168	18-168	18-168#	18-168#	18-168#
	18-169	18-169	18-169	18-169	18-169	18-169	18-169#	18-170	18-170	18-170	18-170	18-170#	18-171	18-171
	18-171	18-171	18-171	18-171	18-171#	18-172	18-172	18-172	18-172#	18-173	18-173	18-173	18-173	18-173
	18-173	18-173#	18-181	18-181	18-181	18-181#	18-182	18-182	18-182	18-182#	18-183	18-183	18-183	18-183#
	18-184	18-184	18-184	18-184	18-184#	18-185	18-185	18-185	18-185#	18-186	18-186	18-186	18-186	18-186
	18-186	18-186	18-186#	18-187	18-187	18-187	18-187#	18-188	18-188	18-188	18-188	18-188	18-188	18-188#
	18-189	18-189	18-189	18-189	18-189#	18-190	18-190	18-190	18-190	18-190	18-190	18-190#	18-191	18-191
	18-191	18-191#	18-192	18-192	18-192	18-192	18-192	18-192#	18-200	18-200	18-200	18-200#	18-200	18-200#
	18-201	18-201	18-201#	18-202	18-202	18-202	18-202#	18-203	18-203	18-203	18-203#	18-204	18-204	18-204
	18-204	18-204	18-204	18-204	18-204	18-204#	18-205	18-205	18-205	18-205#	18-206	18-206	18-206	18-206
	18-206	18-206	18-206#	18-207	18-207	18-207	18-207#	18-208	18-208	18-208	18-208	18-208	18-208	18-208
	18-208#	18-209	18-209	18-209	18-209#	18-210	18-210	18-210	18-210	18-210	18-210	18-210#	18-218	18-218
	18-218	18-218#	18-219	18-219	18-219	18-219	18-219#	18-220	18-220	18-220#	18-221	18-221	18-221	18-221
	18-221#	18-222	18-222	18-222	18-222	18-222#	18-223	18-223	18-223	18-223#	18-224	18-224	18-224	18-224
	18-224	18-224	18-224	18-224#	18-225	18-225	18-225	18-225#	18-226	18-226	18-226	18-226	18-226	18-226
	18-226#	18-227	18-227	18-227	18-227	18-227#	18-228	18-228	18-228	18-228	18-228	18-228	18-228#	18-229
	18-229	18-229	18-229#	18-230	18-230	18-230	18-230	18-230	18-230	18-230#	18-238	18-238	18-238	18-238
	18-238#	18-239	18-239	18-239#	18-240	18-240	18-240	18-240#	18-241	18-241	18-241	18-241	18-241	18-241#
	18-242	18-242	18-242	18-242	18-242	18-242#	18-243	18-243	18-243	18-243#	18-244	18-244	18-244	18-244
	18-244	18-244	18-244	18-244#	18-245	18-245	18-245	18-245#	18-246	18-246	18-246	18-246	18-246	18-246
	18-246	18-246#	18-247	18-247	18-247	18-247#	18-248	18-248	18-248	18-248	18-248	18-248	18-248#	21-85
	21-85	21-85#	21-90	21-90	21-90	21-90#	24-12	24-12	24-12	24-12#	26-737	26-737	26-737	26-737#
	26-739	26-739	26-739	26-739#	26-763	26-763	26-763	26-763#	26-787	26-787	26-787	26-787#	26-823	26-823
	26-823	26-823#	26-825	26-825	26-825	26-825#								
MSPUT1	1-C81#	9-18#	17-:56	17-:56#	17-:56#	17-:61	17-:61	17-:61	17-:61#	17-:61#	17-:61#	17-:61#	17-:70	17-:70
	17-:70#	17-:70#	18-102	18-102	18-102	18-102#	18-102#	18-102#	18-102#	18-108	18-108	18-108	18-108	18-108
	18-108#	18-108#	18-108#	18-108#	18-109	18-109	18-109#	18-109#	18-110	18-110	18-110	18-110	18-110#	18-110#
	18-110#	18-110#	18-111	18-111	18-111	18-111	18-111#	18-111#	18-111#	18-111#	18-112	18-112	18-112	18-112
	18-112#	18-112#	18-112#	18-112#	18-113	18-113	18-113	18-113	18-113	18-113#	18-113#	18-113#	18-113#	18-113#
	18-113#	18-113#	18-114	18-114	18-114	18-114#	18-114#	18-114#	18-115	18-115	18-115	18-115	18-115	18-115
	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#	18-123	18-123	18-123	18-123#	18-123#	18-123#	18-123#	18-123#
	18-124	18-124	18-124#	18-124#	18-125	18-125	18-125	18-125#	18-125#	18-125#	18-125#	18-125#	18-126	18-126
	18-126	18-126	18-126#	18-126#	18-126#	18-126#	18-127	18-127	18-127	18-127#	18-127#	18-127#	18-127#	18-127#
	18-128	18-128	18-128	18-128	18-128	18-128	18-128#	18-128#	18-128#	18-128#	18-128#	18-128#	18-129	18-129
	18-129	18-129#	18-129#	18-129#	18-130	18-130	18-130	18-130	18-130	18-130#	18-130#	18-130#	18-130#	18-130#

	20-8	20-8#	21-8	21-8#	22-8	22-8#	23-8	23-8#	24-8	24-8#	25-9	25-9#	26-20	26-20#
	26-65	26-65#	26-173	26-173#	26-267	26-267#	26-299	26-299#	26-332	26-332#	26-384	26-384#	26-506	26-506#
	26-510	26-510#	26-539	26-539#	26-589	26-589#	26-593	26-593#	26-620	26-620#	26-663	26-663#	26-722	26-722#
	26-726	26-726#	26-754	26-754#	26-778	26-778#	26-812	26-812#	26-850	26-850#	26-969	26-969#	26-;02	26-;02#
	26-;08	26-;08#	26-;54	26-;54#	26-<24	26-<24#	26-<30	26-<30#	26-<83	26-<83#	26-=48	26-=48#	26->13	26->13#
	26->75	26->75#	26-?25	26-?25#	26-?76	26-?76#	26-a09	26-a09#	26-a73	26-a73#	26-a82	26-a82#	26-A26	26-A26#
	26-A45	26-A45#	26-A63	26-A63#	26-A73	26-A73#	26-A98	26-A98#	26-B16	26-B16#	26-B33	26-B33#	26-B52	26-B52#
	26-B69	26-B69#	26-C17	26-C17#	26-C71	26-C71#	26-D25	26-D25#	26-D79	26-D79#	26-E33	26-E33#	26-E87	26-E87#
	26-F41	26-F41#	26-F92	26-F92#	26-G37	26-G37#	27-14	27-14#	28-13	28-13#				
MSSTAR	1-A33#	9-18#												
MS SVC	1-C33#	9-18#	17-272	17-278	17-288	17-294	17-401	17-407	17-565	17-571	17-581	17-587	17-622	17-628
	17-662	17-668	17-676	17-682	17-828	17-833	17-852	17-858	17-867	17-873	17-882	17-888	17-897	17-903
	17-;56	17-;56#	17-;61	17-;61#	17-;70	17-;70#	17-<49	17-<65	17-=01	17-=07	18-102	18-102#	18-103	18-103#
	18-108	18-108#	18-109	18-109#	18-110	18-110#	18-111	18-111#	18-112	18-112#	18-113	18-113#	18-114	18-114#
	18-115	18-115#	18-116	18-116#	18-123	18-123#	18-124	18-124#	18-125	18-125#	18-126	18-126#	18-127	18-127#
	18-128	18-128#	18-129	18-129#	18-130	18-130#	18-131	18-131#	18-132	18-132#	18-133	18-133#	18-134	18-134#
	18-135	18-135#	18-142	18-142#	18-143	18-143#	18-144	18-144#	18-145	18-145#	18-146	18-146#	18-147	18-147#
	18-148	18-148#	18-149	18-149#	18-150	18-150#	18-151	18-151#	18-152	18-152#	18-153	18-153#	18-154	18-154#
	18-161	18-161#	18-162	18-162#	18-163	18-163#	18-164	18-164#	18-165	18-165#	18-166	18-166#	18-167	18-167#
	18-168	18-168#	18-169	18-169#	18-170	18-170#	18-171	18-171#	18-172	18-172#	18-173	18-173#	18-174	18-174#
	18-181	18-181#	18-182	18-182#	18-183	18-183#	18-184	18-184#	18-185	18-185#	18-186	18-186#	18-187	18-187#
	18-188	18-188#	18-189	18-189#	18-190	18-190#	18-191	18-191#	18-192	18-192#	18-193	18-193#	18-200	18-200#
	18-201	18-201#	18-202	18-202#	18-203	18-203#	18-204	18-204#	18-205	18-205#	18-206	18-206#	18-207	18-207#
	18-208	18-208#	18-209	18-209#	18-210	18-210#	18-211	18-211#	18-218	18-218#	18-219	18-219#	18-220	18-220#
	18-221	18-221#	18-222	18-222#	18-223	18-223#	18-224	18-224#	18-225	18-225#	18-226	18-226#	18-227	18-227#
	18-228	18-228#	18-229	18-229#	18-230	18-230#	18-231	18-231#	18-238	18-238#	18-239	18-239#	18-240	18-240#
	18-241	18-241#	18-242	18-242#	18-243	18-243#	18-244	18-244#	18-245	18-245#	18-246	18-246#	18-247	18-247#
	18-248	18-248#	18-249	18-249#	19-11	19-11#	21-25	21-25#	21-28	21-28#	21-31	21-31#	21-34	21-34#
	21-52	21-52#	21-81	21-81#	21-85	21-85#	21-86	21-86#	21-90	21-90#	21-94	21-94#	21-99	21-99#
	22-10	22-10#	22-17	22-17#	22-20	22-20#	23-11	23-11#	24-10	24-10#	24-12	24-12#	24-13	24-13#
	25-10	25-10#	26-46	26-46#	26-105	26-145	26-147	26-147#	26-224	26-251	26-253	26-253#	26-284	26-284#
	26-317	26-317#	26-364	26-366	26-366#	26-419	26-458	26-471	26-480	26-480#	26-510	26-510#	26-533	26-533#
	26-539	26-539#	26-565	26-565#	26-570	26-570#	26-593	26-593#	26-616	26-616#	26-620	26-620#	26-643	26-643#
	26-644	26-644#	26-695	26-695#	26-726	26-726#	26-737	26-737#	26-739	26-739#	26-748	26-750	26-750#	26-754
	26-754#	26-763	26-763#	26-772	26-774	26-774#	26-778	26-778#	26-787	26-787#	26-796	26-798	26-798#	26-799
	26-799#	26-823	26-823#	26-825	26-825#	26-827	26-827#	26-944	26-944#	26-;77	26-;77#	26-;08	26-;08#	26-;37
	26-;38	26-;38#	26-;43	26-;44	26-;44#	26-;48	26-;48#	26-;54	26-;54#	26-;83	26-;84	26-;84#	26-;89	26-;90
	26-;90#	26-;94	26-;94#	26-;96	26-;96#	26-<30	26-<30#	26-<77	26-<77#	26-<83	26-<83#	26-=30	26-=30#	26-=32
	26-=32#	26-=72	26-=94	26-=96	26-=96#	26->34	26->49	26->58	26->58#	26->99	26-?08	26-?08#	26-?49	26-?58
	26-?58#	26-?94	26-?94#	26-a42	26-a42#	26-a82	26-a82#	26-a92	26-a93	26-a93#	26-A02	26-A03	26-A03#	26-A21
	26-A21#	26-A26	26-A26#	26-A37	26-A38	26-A38#	26-A40	26-A40#	26-A45	26-A45#	26-A56	26-A58	26-A58#	26-A63
	26-A63#	26-A68	26-A68#	26-A73	26-A73#	26-A85	26-A86	26-A86#	26-A91	26-A93	26-A93#	26-A98	26-A98#	26-B09
	26-B11	26-B11#	26-B16	26-B16#	26-B26	26-B28	26-B28#	26-B33	26-B33#	26-B45	26-B47	26-B47#	26-B52	26-B52#
	26-B62	26-B64	26-B64#	26-B69	26-B69#	26-B79	26-B80	26-B80#	26-B85	26-B86	26-B86#	26-B91	26-B93	26-B93#
	26-B97	26-B97#	26-C42	26-C51	26-C51#	26-C96	26-D05	26-D05#	26-D50	26-D59	26-D59#	26-E04	26-E13	26-E13#
	26-E58	26-E67	26-E67#	26-F11	26-F21	26-F21#	26-F65	26-F75	26-F75#	26-G18	26-G18#	26-G71	26-G83	26-H45
	26-H72	26-H72#												
MSLAB	1-C29#	9-18#	17-272#	17-278#	17-288#	17-294#	17-401#	17-407#	17-565#	17-571#	17-581#	17-587#	17-622#	17-628#
	17-662#	17-668#	17-676#	17-682#	17-828#	17-833#	17-852#	17-858#	17-867#	17-873#	17-882#	17-888#	17-897#	17-903#
	17-;56#	17-;61#	17-;70#	17-<49#	17-<65#	17-=01#	17-=07#	18-102#	18-103#	18-108#	18-109#	18-110#	18-111#	18-112#
	18-113#	18-114#	18-115#	18-116#	18-123#	18-124#	18-125#	18-126#	18-127#	18-128#	18-129#	18-130#	18-131#	18-132#
	18-133#	18-134#	18-135#	18-142#	18-143#	18-144#	18-145#	18-146#	18-147#	18-148#	18-149#	18-150#	18-151#	18-152#
	18-153#	18-154#	18-161#	18-162#	18-163#	18-164#	18-165#	18-166#	18-167#	18-168#	18-169#	18-170#	18-171#	18-172#
	18-173#	18-174#	18-181#	18-182#	18-183#	18-184#	18-185#	18-186#	18-187#	18-188#	18-189#	18-190#	18-191#	18-192#
	18-193#	18-200#	18-201#	18-202#	18-203#	18-204#	18-205#	18-206#	18-207#	18-208#	18-209#	18-210#	18-211#	18-218#
	18-219#	18-220#	18-221#	18-222#	18-223#	18-224#	18-225#	18-226#	18-227#	18-228#	18-229#	18-230#	18-231#	18-238#
	18-239#	18-240#	18-241#	18-242#	18-243#	18-244#	18-245#	18-246#	18-247#	18-248#	18-249#	19-11#	21-25#	21-28#

MSTSTL

21-31#	21-34#	21-52#	21-81#	21-85#	21-86#	21-90#	21-94#	21-99#	22-10#	22-17#	22-20#	23-11#	24-10#
24-12#	24-13#	25-10#	26-46#	26-105#	26-145#	26-147#	26-224#	26-251#	26-253#	26-284#	26-317#	26-364#	26-366#
26-419#	26-458#	26-471#	26-480#	26-510#	26-533#	26-539#	26-565#	26-570#	26-593#	26-616#	26-620#	26-643#	26-644#
26-695#	26-726#	26-737#	26-739#	26-748#	26-750#	26-754#	26-763#	26-772#	26-774#	26-778#	26-787#	26-796#	26-798#
26-799#	26-823#	26-825#	26-827#	26-944#	26-:77#	26-:08#	26-:37#	26-:38#	26-:43#	26-:44#	26-:48#	26-:54#	26-:83#
26-:84#	26-:89#	26-:90#	26-:94#	26-:96#	26-<30#	26-<77#	26-<83#	26-=30#	26-=32#	26-=72#	26-=94#	26-=96#	26->34#
26->49#	26->58#	26->99#	26-?08#	26-?49#	26-?58#	26-?94#	26-@42#	26-@82#	26-@92#	26-@93#	26-A02#	26-A03#	26-A21#
26-A26#	26-A37#	26-A38#	26-A40#	26-A45#	26-A56#	26-A58#	26-A63#	26-A68#	26-A73#	26-A85#	26-A86#	26-A91#	26-A93#
26-A98#	26-B09#	26-B11#	26-B16#	26-B26#	26-B28#	26-B33#	26-B45#	26-B47#	26-B52#	26-B62#	26-B64#	26-B69#	26-B79#
26-B80#	26-B85#	26-B86#	26-B91#	26-B93#	26-B97#	26-C42#	26-C51#	26-C96#	26-D05#	26-D50#	26-D59#	26-E04#	26-E13#
26-E58#	26-E67#	26-F11#	26-F21#	26-F65#	26-F75#	26-G18#	26-G71#	26-G83#	26-H45#	26-H72#			
1-C21#	9-18#	17-272	17-272#	17-272#	17-278	17-278#	17-278#	17-288	17-288#	17-288#	17-294	17-294#	17-294#
17-401	17-401#	17-401#	17-407	17-407#	17-407#	17-565	17-565#	17-565#	17-571	17-571#	17-571#	17-581	17-581#
17-581#	17-587	17-587#	17-587#	17-622	17-622#	17-622#	17-628	17-628#	17-628#	17-662	17-662#	17-662#	17-668
17-668#	17-668#	17-676	17-676#	17-676#	17-682	17-682#	17-682#	17-828	17-828#	17-828#	17-833	17-833#	17-833#
17-852	17-852#	17-852#	17-858	17-858#	17-858#	17-867	17-867#	17-867#	17-873	17-873#	17-873#	17-882	17-882#
17-882#	17-888	17-888#	17-888#	17-897	17-897#	17-897#	17-903	17-903#	17-903#	17-:56	17-:56#	17-:61	17-:61#
17-:70	17-:70#	17-<49	17-<49#	17-<49#	17-<65	17-<65#	17-<65#	17-=01	17-=01#	17-=01#	17-=07	17-=07#	17-=07#
18-102	18-102#	18-103	18-103#	18-108	18-108#	18-109	18-109#	18-110	18-110#	18-111	18-111#	18-112	18-112#
18-113	18-113#	18-114	18-114#	18-115	18-115#	18-116	18-116#	18-123	18-123#	18-124	18-124#	18-125	18-125#
18-126	18-126#	18-127	18-127#	18-128	18-128#	18-129	18-129#	18-130	18-130#	18-131	18-131#	18-132	18-132#
18-133	18-133#	18-134	18-134#	18-135	18-135#	18-142	18-142#	18-143	18-143#	18-144	18-144#	18-145	18-145#
18-146	18-146#	18-147	18-147#	18-148	18-148#	18-149	18-149#	18-150	18-150#	18-151	18-151#	18-152	18-152#
18-153	18-153#	18-154	18-154#	18-161	18-161#	18-162	18-162#	18-163	18-163#	18-164	18-164#	18-165	18-165#
18-166	18-166#	18-167	18-167#	18-168	18-168#	18-169	18-169#	18-170	18-170#	18-171	18-171#	18-172	18-172#
18-173	18-173#	18-174	18-174#	18-181	18-181#	18-182	18-182#	18-183	18-183#	18-184	18-184#	18-185	18-185#
18-186	18-186#	18-187	18-187#	18-188	18-188#	18-189	18-189#	18-190	18-190#	18-191	18-191#	18-192	18-192#
18-193	18-193#	18-200	18-200#	18-201	18-201#	18-202	18-202#	18-203	18-203#	18-204	18-204#	18-205	18-205#
18-206	18-206#	18-207	18-207#	18-208	18-208#	18-209	18-209#	18-210	18-210#	18-211	18-211#	18-218	18-218#
18-219	18-219#	18-220	18-220#	18-221	18-221#	18-222	18-222#	18-223	18-223#	18-224	18-224#	18-225	18-225#
18-226	18-226#	18-227	18-227#	18-228	18-228#	18-229	18-229#	18-230	18-230#	18-231	18-231#	18-238	18-238#
18-239	18-239#	18-240	18-240#	18-241	18-241#	18-242	18-242#	18-243	18-243#	18-244	18-244#	18-245	18-245#
18-246	18-246#	18-247	18-247#	18-248	18-248#	18-249	18-249#	19-11	19-11#	21-25	21-25#	21-28	21-28#
21-31	21-31#	21-34	21-34#	21-52	21-52#	21-81	21-81#	21-85	21-85#	21-86	21-86#	21-90	21-90#
21-94	21-94#	21-99	21-99#	22-10	22-10#	22-17	22-17#	22-20	22-20#	23-11	23-11#	24-10	24-10#
24-12	24-12#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-105	26-105#	26-105#	26-145	26-145#	26-145#
26-147	26-147#	26-224	26-224#	26-224#	26-251	26-251#	26-251#	26-253	26-253#	26-284	26-284#	26-317	26-317#
26-364	26-364#	26-364#	26-366	26-366#	26-419	26-419#	26-419#	26-458	26-458#	26-458#	26-471	26-471#	26-471#
26-480	26-480#	26-510	26-510#	26-533	26-533#	26-539	26-539#	26-565	26-565#	26-570	26-570#	26-593	26-593#
26-616	26-616#	26-620	26-620#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-726	26-726#	26-737	26-737#
26-739	26-739#	26-748	26-748#	26-748#	26-750	26-750#	26-754	26-754#	26-763	26-763#	26-772	26-772#	26-772#
26-774	26-774#	26-778	26-778#	26-787	26-787#	26-796	26-796#	26-796#	26-798	26-798#	26-799	26-799#	26-823
26-823#	26-825	26-825#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:08	26-:08#	26-:37	26-:37#	26-:37#
26-:38	26-:38#	26-:43	26-:43#	26-:43#	26-:44	26-:44#	26-:48	26-:48#	26-:54	26-:54#	26-:83	26-:83#	26-:83#
26-:84	26-:84#	26-:89	26-:89#	26-:89#	26-:90	26-:90#	26-:94	26-:94#	26-:96	26-:96#	26-<30	26-<30#	26-<77
26-<77#	26-<83	26-<83#	26-=30	26-=30#	26-=32	26-=32#	26-=72	26-=72#	26-=72#	26-=94	26-=94#	26-=94#	26-=96
26-=96#	26->34	26->34#	26->34#	26->49	26->49#	26->49#	26->58	26->58#	26->58#	26->99	26->99#	26->99#	26-?08
26-?49	26-?49#	26-?49#	26-?58	26-?58#	26-?94	26-?94#	26-@42	26-@42#	26-@82	26-@82#	26-@92	26-@92#	26-@92#
26-@93	26-@93#	26-A02	26-A02#	26-A02#	26-A03	26-A03#	26-A21	26-A21#	26-A26	26-A26#	26-A37	26-A37#	26-A37#
26-A38	26-A38#	26-A40	26-A40#	26-A45	26-A45#	26-A56	26-A56#	26-A56#	26-A58	26-A58#	26-A63	26-A63#	26-A68
26-A68#	26-A73	26-A73#	26-A85	26-A85#	26-A85#	26-A86	26-A86#	26-A91	26-A91#	26-A91#	26-A93	26-A93#	26-A98
26-A98#	26-B09	26-B09#	26-B09#	26-B11	26-B11#	26-B16	26-B16#	26-B26	26-B26#	26-B26#	26-B28	26-B28#	26-B33
26-B33#	26-B45	26-B45#	26-B45#	26-B47	26-B47#	26-B52	26-B52#	26-B62	26-B62#	26-B62#	26-B64	26-B64#	26-B69
26-B69#	26-B79	26-B79#	26-B79#	26-B80	26-B80#	26-B85	26-B85#	26-B85#	26-B86	26-B86#	26-B91	26-B91#	26-B91#
26-B93	26-B93#	26-B97	26-B97#	26-C42	26-C42#	26-C42#	26-C51	26-C51#	26-C96	26-C96#	26-C96#	26-D05	26-D05#
26-D50	26-D50#	26-D50#	26-D59	26-D59#	26-E04	26-E04#	26-E04#	26-E13	26-E13#	26-E13#	26-E58	26-E58#	26-E67
26-E67#	26-F11	26-F11#	26-F11#	26-F21	26-F21#	26-F65	26-F65#	26-F65#	26-F75	26-F75#	26-G18	26-G18#	26-G71

