

DC11

OFF-LINE DIAG TEST
MD-11-DZDCA-A

EP-DZDCA-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

The image displays a grid of 180 small, illegible data tables or charts, arranged in 12 columns and 15 rows. Each cell in the grid contains a small, dense table of text, which appears to be diagnostic test results or data points. The text is too small and faded to be read, but the layout suggests a structured format for recording test outcomes. The grid is located on the left side of the page, with the right side being a large, dark, mostly blank area.

B01

.REMARKSCODE

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDCA-A
PRODUCT NAME: DC11 OFF LINE DIAGNOSTIC TEST
DATE: APRIL 21, 1976
MAINTAINER: DIAGNOSTIC GROUP

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

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

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT 1973 ,1976 BY DIGITAL EQUIPMENT CORPORATION

DZDCA8.P11

0
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

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR THE DC-11 (ASYNCHRONOUS MODEM INTERFACE), MAINDEC-11-DZDCAA (DC-11 OFF LINE TESTS) AND MAINDEC-11-DZDCBA (DC-11 ON LINE TESTS). THE OFF LINE TESTS TEST ALL DC11 LOGIC AND MAY BE USED TO INDIVIDUALLY TEST UP TO 32 DC-11'S. THE OFF LINE TESTS DO NOT REQUIRE THE USE OF A MODEM, HOWEVER A SPECIAL JUMPER CONNECTOR IS REQUIRED. THE ON LINE TESTS ARE ESSENTIALLY DATA RELIABILITY TESTS REQUIRING THE USE OF MODEMS AND A SUITABLE TERMINAL DEVICE.

THIS DOCUMENT DESCRIBES THE OFF LINE TESTS.

THE AVAILABLE TESTS ARE:

- PRG0 INPUT/OUTPUT LOGIC TESTS
- PRG1 TRANSMITTER SCOPE LOOP
- PRG2 RECEIVER SCOPE LOOP
- PRG3 SINGLE CHARACTER MAINT. MODE DATA TEST
- PRG4 SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST

2. REQUIREMENTS

2.1 EQUIPMENT

- A. PDP 11/20 SYSTEM
- B. DC11
- C. SPECIAL JUMPER CONNECTOR (SEE DC11 MAINTENANCE MANUAL FOR DETAILED DESCRIPTION)

2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (4K) EXCEPT THAT AREA RESERVED FOR THE BOOTSTRAP AND ABSOLUTE LOADERS.

3. LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

4. USE PROCEDURE
BEFORE STARTING ANY OF THE SELECTABLE PROGRAMS MAKE SURE
THAT THE TTY IS IN REMOTE MODE; AND THAT THE PROGRAM SELECTED
IS A LEGAL PROGRAM, IE. SR 0-2=0-4, OTHERWISE PROGRAM OP-
ERATION IS UNSPECIFIED. RELOAD PROGRAM AND START OVER.
- 4.1 PRGD INPUT/OUTPUT LOGIC TESTS
- A. LOAD ADDRESS = 000200 (RESTART LOAD ADDR. = 000200)
LOAD SR 0-2 = 0, AND PRESS START SWITCH.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED.
'PRGD-INPUT-OUTPUT LOGIC TESTS. DISCONNECT DC11 FROM MODEM
AND CONNECT JUMPER TO CABLE'
DISCONNECT THE DC11 FROM THE MODEM AND INSERT THE JUMPER CON-
NECTOR IN THE MODEM END OF THE CABLE, AND PRESS CONTINUE.
NOTE, IF THE CABLE IS LEFT CONNECTED TO THE MODEM THE FOL-
LOWING TESTS WILL FAIL:
AT5, AT34-AT42, AT44, AND AT144
- B. THE PROGRAM WILL NOW REQUEST THE LINE # (8) YOU WISH TO
TEST. LOAD THE LINE # AS REQUESTED AND PRESS CONTINUE.
LINE NUMBER REFERS TO THE ADDRESSES TO WHICH THE DC11 RESPONDS.
- | | | | | | | | |
|--------|--------|---------|--------|---------|--------|---------|--------|
| LINE 0 | 77400X | LINE 10 | 77410X | LINE 20 | 77420X | LINE 30 | 77430X |
| LINE 1 | 77401X | LINE 11 | 77411X | LINE 21 | 77421X | LINE 31 | 77431X |
| LINE 2 | 77402X | LINE 12 | 77412X | LINE 22 | 77422X | LINE 32 | 77432X |
| LINE 3 | 77403X | LINE 13 | 77413X | LINE 23 | 77423X | LINE 33 | 77433X |
| LINE 4 | 77404X | LINE 14 | 77414X | LINE 24 | 77424X | LINE 34 | 77434X |
| LINE 5 | 77405X | LINE 15 | 77415X | LINE 25 | 77425X | LINE 35 | 77435X |
| LINE 6 | 77406X | LINE 16 | 77416X | LINE 26 | 77426X | LINE 26 | 77436X |
| LINE 7 | 77407X | LINE 17 | 77417X | LINE 27 | 77427X | LINE 27 | 77437X |
- C. THE PROGRAM WILL TYPE OUT INSTRUCTIONS TO SET IN THE DESIRED
SR OPTIONS. PRESS CONTINUE WHEN THE OPTIONS ARE IN THE SR.
THE AVAILABLE OPTIONS ARE:
SR 0-6 ROUTINE TO BE RUN (IF ENABLED BY SR9)
SR8 DISABLE STALL MODE
SR9 LOOP SELECTED ROUTINE
SR10 HALT AT END OF PROGRAM
SR11 INHIBIT ITERATION
SR13 INHIBIT PRINTOUT
SR14 SCOPE
SR15 HALT ON ERROR.
- D. THE PROGRAM WILL NOW BEGIN TESTING THE DC11 YOU SELECTED.
- E. REFER TO SECTION 5.1.2 FOR ERROR DESCRIPTION
- F. AFTER ONE COMPLETE PASS PRGEND WILL BE TYPED OUT

145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183

4.2 PRG1 - TRANSMITTER SCOPE LOOP

- A. LOAD ADDRESS = 000200
LOAD SR 0-2 = 1, AND PRESS START SWITCH.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS RE-
QUESTED AND PRESS CONTINUE.

B.
THE PROGRAM WILL REQUEST THE DC11 LINE PARAMETERS. LOAD THE
PARAMETERS AS REQUESTED AND PRESS CONTINUE.

- C. THE PROGRAM WILL REQUEST A CHARACTER CODE, AND A DELAY
TIME. THE CHARACTER CODE IS THE DATA THE DC11 WILL TRANSMIT
AND THE DELAY IS THE TIME ELAPSED BETWEEN SUCCESSIVE TRANS-
MISSIONS OF ONE CHARACTER. PRESS CONTINUE WHEN THIS IS DONE.

D. THE PROGRAM WILL RUN WITHOUT ERROR OR END TYPEOUTS.

4.3 PRG2 - RECEIVER SCOPE LOOP

- A. LOAD ADDRESS = 000200
LOAD SR 0-2 = 2, AND PRESS START.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS REQ-
UESTED AND PRESS CONTINUE.

B. THE PROGRAM WILL REQUEST THE DC11 LINE PARAMETERS. LOAD THE
PARAMETERS AS REQUESTED AND PRESS CONTINUE.

- C. THE PROGRAM WILL REQUEST A TEST CHARACTER CODE, AND A DELAY
TIME. THE CHARACTER CODE IS THE DATA THAT THE DC11 WILL BE
TRANSMITTING AND THE DELAY IS THE ELAPSED TIME BETWEEN SUCCE-
SSIVE CHARACTERS. PRESS CONTINUE WHEN THIS IS DONE.

D. THE PROGRAM WILL NOW RUN WITHOUT ERROR OR END TYPEOUTS.

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

4.4 PRG3 - SINGLE CHARACTER MAINT MODE DATA TEST

- A. LOAD ADDRESS = 000200
LOAD SR 0-2 = 3, AND PRESS START.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS REQ-
UESTED AND PRESS CONTINUE.
- B. THE PROGRAM WILL REQUEST THE DC11 LINE PARAMETERS. LOAD THE
PARAMETERS AS REQUESTED AND PRESS CONTINUE.
- C. THE PROGRAM WILL REQUEST A TEST CHARACTER. LOAD THE TEST CHAR-
ACTER AND PRESS CONTINUE.
- D. THE PROGRAM WILL NOW RUN CONTINUOUSLY REPORTING ANY DATA FAIL-
URES.

4.5 PRG4 - SPECIAL BINARY COUNT MAINT. MODE DATA TEST

- A. LOAD ADDRESS = 000200
LOAD SR 0-2 = 4, AND PRESS START.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS REQ-
UESTED AND PRESS CONTINUE.
- B. THE PROGRAM WILL NOW REQUEST THE DC11 LINE PARAMETERS. LOAD THE
PARAMETERS AS REQUESTED AND PRESS CONTINUE.
- C. THE PROGRAM WILL BEGIN TESTING THE LINE YOU SELECTED.
AND REPORT ANY DATA ERRORS.

5. PROGRAM DESCRIPTIONS

5.1 PRG0 - INPUT/OUTPUT LOGIC TESTS

THE INPUT/OUTPUT LOGIC TESTS CONSIST OF 103(8) ROUTINES WHICH
MAY BE RUN IN SEQUENTIAL ORDER OR INDIVIDUALLY LOOPED (SEE
SECT 4.1, C FOR SWITCH SETTINGS). THE JUMPER CONNECTOR MUST
BE INSERTED BEFORE STARTING.

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
AT0-AT3 AT4-AT42	ADDRESSABILITY OF CSRS & DBRS DIDDLES ALL BITS IN THE CSRS AND CHECKS THAT THEY CAN BE READ/WITTEN PROPERLY.
AT43-AT44 AT45-AT51 AT52-AT63	PROPER OPERATION OF RESET INSTRUCTION PROPER OPERATION OF READY BIT PROPER OPERATION OF CHAR LENGTH, SPEED CONTROL AND STOP CODE BITS.
AT64	PROPER OPERATION OF DATA OVERFLOW BIT
AT65-AT74	PROPER OPERATION OF INTERRUPTS
AT75	DATA OVERFLOW CLEARS DONE
AT76	ERROR CAUSES INTERRUPT
AT77	PROPER OPERATION OF PARITY BIT
AT100-AT137	DATA TESTS THESE TESTS TEST ALL POSSIBLE COMBINATIONS OF CHARACTER LENGTH SPEED AND STOP CODES USING MAINT. MODE.
AT140	DATA TEST HIGH SPEED (JUMPER)
AT141	PROPER OPERATION OF BREAK BIT

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
58
59
60

261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305

5.1.2 ERROR DESCRIPTION

IF A ROUTINE FAILS AND THE INHIBIT PRINTOUT SWITCH IS NOT ENABLED (SR13) A PRINTOUT RESULTS. THE PRINTOUT FORMAT IS:

P(PROG#) T(ROUTINE#) PC=(PC OF ERROR CALL) AND AN ADDITIONAL/MESSAGE (IF APPLICABLE)

POO T005 PC=XXXX INDICATING THAT TXCSR BIT 1 WAS SET (SHOULD'VE BEEN CLEAR)

POO T122 PC=XXXX DATA ERR S/B:---WAS--- INDICATING A DATA ERROR WHEN DC11 PARAMETERS WERE SET AT CHAR LENGTH=6 SPEED=00, AND STOP CODE=1

TO RESUME TESTING PRESS CONTINUE. IF ROUTINES 65 OR 71 FAIL DUE TO A BAD TRAP VECTOR, I.E. THE VECTOR PROVIDED BY THE INTERRUPTING DC11 IS INCORRECT THE PROGRAM WILL HALT AND DISPLAY THE VECTOR+2 THAT WAS PROVIDED BY THE INTERRUPTING DC11. TO RECOVER FROM THIS TYPE OF ERROR IT WILL BE NECESSARY TO PUT INTO THE INCORRECT VECTOR ADDRESS THE ADDRESS TO RUN THE ROUTINE. I.E. ADDRESS ATAA AND AXAA FOR ROUTINES 65 AND 71 RESPECTIVELY.

5.1.3 JUMPER CONNECTOR

THE JUMPER CONNECTOR TESTS THOSE F/F'S, GATES (RING INDICATOR, CARRIER TRANSITION, CLEAR TO SEND, AND SUPERVISORY RECEIVE DATA) WHICH CANNOT BE TESTED UNLESS A DATA SET IS ACTUALLY CONNECTED TO THE DC11. IN ADDITION TO TESTING DC11 LOGIC THE JUMPER ALSO TESTS CABLE WIRING TO/FROM THE DC11/DATA SET. THE FOLLOWING TESTS WILL FAIL IF THE CABLE IS NOT INSALLED IN THE DC11:

AT5, AT34-AT42, AT44
AT140 WILL LOOP CONTINUOUSLY

IF THE JUMPER IS REMOVED FROM THE END OF THE CABLE AND THE CABLE IS LEFT CONNECTED TO THE DC11 THE ABOVE TESTS WILL FAIL WITH THE PROBABLE EXCEPTIONS OF AT35 AND AT36.

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

5.2 PRG1-TRANSMITTER SCOPE LOOP

THE PURPOSE OF PRG1 IS TO ALLOW SCOPING OF TRANSMITTER FUNCTIONS IN A RUN CONDITION USING USER SPECIFIED DC11 PARMATERS AND DATA. NO ERROR PRINTOUTS ARE PROVIDED.

5.3 PKG2-RECEIVER SCOPE LOOP

THE PURPOSE OF PRG2 IS TO ALLOW SCOPING OF RECEIVER FUNCTIONS IN A RUN CONDITION USING USER SPECIFIED DC11 PARAMETERS AND DATA. NO ERROR PRINTOUTS ARE PROVIDED.

5.4 PRG3-SINGLE CHARACTER MAINT MODE DATA TEST

PRG3 TRANSMITS, RECEIVES AND CHECKS RECEIVED DATA USING USER SPECIFIED DC11, DC11 PARAMETERS, AND DATA.

5.4.1 ERROR PRINTOUTS

SELF EXPLANITORY ERROR PRINTOUTS ARE PROVIDED.

5.5 PRG4-SPECIAL BINARY COUNT MAINT MODE DATA TEST

PRG4 IS THE SAME AS PRG3 ROUTINE 77 AND 100 EXCEPT THAT THE USER SPECIFIES DC11 RUNNING PARAMETERS.

5.5.1 ERROR PRINTOUTS

SELF EXPLANATORY PRINTOUTS ARE PROVIDED.

6.0 POWER FAIL

A POWER FAIL ROUTINE IS INCLUDED IN THE PROGRAM. WHEN THE POWER FAILS THE PROGRAM WILL AUTOMATICALLY RESTART USING THE PRESENT SR OPTIONS AND THE LINE PREVIOUSLY SELECTED. NOTE: THE POWER MAY FAIL WHEN THE PROGRAM IS EXECUTING A 'RESET' INSTRUCTION. IN THIS CASE OPERATOR INTERVENTION IS NEEDED TO PRESS CONTINUE. AN ERROR TYPEOUT RESULTS AND WILL TYPE THE PROGRAM #, THE ROUTINE THAT WAS RUNNING AT THE TIME THE POWER FAILED (PROGRAM 0 ONLY), AND THE PC OF THE POWER FAIL ERROR CALL.

P(PRG#) T(ROUTINE #) PC = (ADDRESS OF ERROR CALL)

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
400
401
402
403
404
405
406

DC11 DATA

```

*****
LINE # * UNIT# * RXCSR ADRS. * FREQS. * TELEPHONE * REMARKS
(CUST ID) *(DEC ID)* RX VECTOR * AVAILABLE * NUMBER *
*****
* * * * *
* * * * *
* * * * *
*****
* * * * *
* * * * *
* * * * *
*****
* * * * *
* * * * *
* * * * *
*****
* * * * *
* * * * *
* * * * *
*****
* * * * *
* * * * *
* * * * *
*****
* * * * *
* * * * *
* * * * *
*****
*****
;DC11 DIAGNOSTIC PROGRAM (OFF LINE TESTS)
*****

```

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

:COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
:PRG0- INFJT-OUTPUT LOGIC TESTS
:PRG1- TRANSMITTER SCOPE LOOP
:PRG2- PECEIVER SCOPE LOOP
:PRG3- SINGLE CHARACTER MAINTENANCE MODE DATA TEST
:PRG4- SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST
:STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1)
:
:SR15- HALT ON ERROR
:SR14- SCOPE.
:SR13- INHIBIT PRINTOUT
:SR12- INHIBIT TRACE
:SR11- INHIBIT ITERATION.
:SR10- HALT AT END OF PROGRAM
:SR9- SELECT ROUTINE.
:SR8- DISABLE STALL MODE AND RUN FULL SPEED.
:SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE SELECTED.
:DATA TEST PARAMETERS

	CHAR LENGTH	SPEED	STOP CODE
:NOTE0	8	00	UUUUUUUUUU
:NOTE1	7	00	UUUUUUUU
:NOTE2	6	00	UUUUUU
:NOTE3	5	00	UUUUU
:NOTE4	8	01	UUUUUUUUUU
:NOTE5	7	01	UUUUUUUU
:NOTE6	6	01	UUUUUU
:NOTE7	5	01	UUUUU
:NOTE10	8	10	UUUUUUUUUU
:NOTE11	7	10	UUUUUUUU
:NOTE12	6	10	UUUUUU
:NOTE13	5	10	UUUUU
:NOTE14	8	11	UUUUUUUUUU
:NOTE15	7	11	UUUUUUUU
:NOTE16	6	11	UUUUUU
:NOTE17	5	11	UUUUU
:NOTE20	8	00	UUUUUUUUUU
:NOTE21	7	00	UUUUUUUU
:NOTE22	6	00	UUUUUU
:NOTE23	5	00	UUUUU
:NOTE24	8	01	UUUUUUUUUU
:NOTE25	7	01	UUUUUUUU
:NOTE26	6	01	UUUUUU
:NOTE27	5	01	UUUUU
:NOTE30	8	10	UUUUUUUUUU
:NOTE31	7	10	UUUUUUUU
:NOTE32	6	10	UUUUUU
:NOTE33	5	10	UUUUU
:NOTE34	8	11	UUUUUUUUUU
:NOTE35	7	11	UUUUUUUU
:NOTE36	6	11	UUUUUU
:NOTE37	5	11	UUUUU

459		000000	. = 0	
460	000000	000002	. + 2	; UNASSIGNED TRAP
461	000002	000000	HALT	
462	000004	000006	. + 2	; SP OVERFLOW, BUS ERROR TRAP
463	000006	000000	HALT	
464	000010	000012	. + 2	; RESERVED INSTRUCTION TRAP
465	000012	000000	HALT	
466	000014	000016	. + 2	; TRACE TRAP
467	000016	000000	HALT	
468	000020	000022	. + 2	; TRAP TO CALL IOX
469	000022	000002	2	
470	000024	000026	. + 2	; POWER FAIL TRAP
471	000026	000000	HALT	
472	000030	002105	EMTINT	; EMT TRAP
473	000032	000340	PRTY?	
474	000034	000036	. + 2	
475	000036	000000	HALT	
476	000040	000042	. + 2	
477	000042	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
478	000044	000046	. + 2	
479	000046	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
480	000050	000052	. + 2	
481	000052	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
482	000054	000056	. + 2	
483	000056	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
484	000060	000062	. + 2	
485	000062	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
486	000064	000066	. + 2	
487	000066	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
488	000070	000072	. + 2	
489	000072	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
490	000074	000076	. + 2	
491	000076	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
492	000100	000102	. + 2	
493	000102	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
494	000104	000106	. + 2	
495	000106	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
496	000110	000112	. + 2	
497	000112	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
498	000114	000116	. + 2	
499	000116	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
500	000120	000122	. + 2	
501	000122	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
502	000124	000126	. + 2	
503	000126	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
504	000130	000132	. + 2	
505	000132	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
506	000134	000136	. + 2	
507	000136	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
508	000140	000142	. + 2	
509	000142	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
510	000144	000146	. + 2	
511	000146	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
512	000150	000152	. + 2	
513	000152	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
514	000154	000156	. + 2	

MACHER:

MO1

.MAIN. MACY11 27(732) 28-SEP-76 13:35 PAGE 13
 DZDCAB.P11

515	000156	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
516	000160	000162	.+2	
517	000162	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
518	000164	000166	.+2	
519	000166	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
520	000170	000172	.+2	
521	000172	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
522	000174	000176	.+2	
523	000176	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
524	000200	000202	.+2	
525	000202	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
526	000204	000206	.+2	
527	000206	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
528	000210	000212	.+2	
529	000212	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
530	000214	000216	.+2	
531	000216	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
532	000220	000222	.+2	
533	000222	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
534	000224	000226	.+2	
535	000226	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
536	000230	000232	.+2	
537	000232	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
538	000234	000236	.+2	
539	000236	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
540	000240	000242	.+2	
541	000242	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
542	000244	000246	.+2	
543	000246	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
544	000250	000252	.+2	
545	000252	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.

546				
547	000254	000256	.+2	
548	000256	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
549	000260	000262	.+2	
550	000262	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
551	000264	000266	.+2	
552	000266	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
553	000270	000272	.+2	
554	000272	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
555	000274	000276	.+2	
556	000276	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
557	000300	000302	.+2	
558	000302	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
559	000304	000306	.+2	
560	000306	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
561	000310	000312	.+2	
562	000312	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
563	000314	000316	.+2	
564	000316	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
565	000320	000322	.+2	
566	000322	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
567	000324	000326	.+2	
568	000326	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
569	000330	000332	.+2	
570	000332	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
571	000334	000336	.+2	
572	000336	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
573	000340	000342	.+2	
574	000342	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
575	000344	000346	.+2	
576	000346	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
577	000350	000352	.+2	
578	000352	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
579	000354	000356	.+2	
580	000356	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
581	000360	000362	.+2	
582	000362	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
583	000364	000366	.+2	
584	000366	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
585	000370	000372	.+2	
586	000372	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
587	000374	000376	.+2	
588	000376	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.

```

;EQUATE STATEMENTS
SR=177570
CC=177776
PSW=177776
SPBOT=1076
NOP=240
OPEN=0
MANUAL=BIT15
BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
    
```

589		
590		
591	177570	
592	177776	
593	177776	
594	001076	
595	000240	
596	000000	
597	100000	
598	100000	
599	040000	
600	020000	
601	010000	

602	004000	BIT11=4000
603	002000	BIT10=2000
604	001000	BIT9=1000
605	000400	BIT8=400
606	000200	BIT7=200
607	000100	BIT6=100
608	000040	BIT5=40
609	000020	BIT4=20
610	000010	BIT3=10
611	000004	BIT2=4
612	000002	BIT1=2
613	000001	BIT0=1
614	005726	POPSP=5726
615	022626	POPSP2=022626
616	000340	PRTY7=340
617	000300	PRTY6=300
618	000240	PRTY5=240
619	000200	PRTY4=200
620	000140	PRTY3=140
621	000100	PRTY2=100
622	000040	PRTY1=40
623	000000	PRTY0=0
624	104000	TYPE=EMT+0
625	104001	TYPES=EMT+1
626	104002	STALL=EMT+2
627	104003	ERROR=EMT+3
628	104004	DATCHK=EMT+4
629	104005	CHALT=EMT+5
630	104006	STRXV=EMT+6
631	104007	STTXV=EMT+7
632	104010	EHALT=EMT+10
633	104011	SRESET=EMT+11
634	104012	SCOPE=EMT+12
635	104013	SAVREG=EMT+13
636	104014	RSTREG=EMT+14
637	104015	ERROR1=EMT+15
638	104016	DELAY=EMT+16
639	000007	BELL=007
640	177777	ATLAST=-1

```

:POP THE STACK. SAME AS TST (6)+
:POP STACK TWICE. SAME AS CMP (6)+,(6)+
:PRIORITY LEVEL DEFINITIONS
  
```

641						
642		000200		=200		
643	000200	000167	001356	JMP	START	;GO TO START OF PROGRAM.
644		001100		=1100		
645	001100	174000		RXCSR:		;RECEIVER CSR
646	001102	174002		RXBUF:		;RECEIVER BUFFER
647	001104	174004		TXCSR:		;TRANSMITTER CSR
648	001106	174006		TXBUF:		;TRANSMITTER BUFFER
649	001110	000000		RXVTR:	OPEN	;RECEIVER VECTOR
650	001112	000240		RXLVL:	PRTYS	;RECEIVER PRIORITY LEVEL
651	001114	000304		TXVTR:	304	;TRANSMITTER VECTOR
652	001116	000240		TXLVL:	PRTYS	;TRANSMITTER PRIORITY LEVEL
653	001120	177560		TKS:	177560	;LSR CSR
654	001122	177562		TKB:	177562	;LSR BUFFER
655	001124	177564		TPS:	177564	;LSP CSR
656	001126	177566		TPB:	177566	;LSP BUFFER
657	001130	000060		TKVTR:	60	;LSR INTERRUPT VECTOR
658	001132	000200		TKLVL:	PRTY4	;LSR PRIORITY LEVEL
659	001134	000064		TPVTR:	64	;LSP INTERRUPT VECTOR
660	001136	000200		TPLVL:	PRTY4	;LSP PRIORITY LEVEL
661	001140	000000		PRGNUM:	OPEN	;CONTAINS CURRENT PROGRAM#
662	001142	000000		KSTART:	OPEN	;CURRENT PROGRAM START ADDRESS.
663	001144	000000		CURTST:	OPEN	;CONTAINS ADDR OF CURRENT TEST.
664	001146	000000		RTNNO:	OPEN	;CONTAINS CURRENT TEST #.
665	001150	000000		NXTST:	OPEN	;CONTAINS ADDR OF NEXT TEST.
666	001152	000000		ICTR:	OPEN	;CONTAINS CURRENT ITERATION COUNT
667	001154	000000		SCOPTR:	OPEN	;CONTAINS CURRENT SCOPE POINTER.
668	001156	003654		PRGTAB:	PRG0	;PRG0 START ADDRESS
669	001160	014342			PRG1	;PRG1 START ADDRESS
670	001162	014406			PRG2	;PRG2 START ADDRESS
671	001164	014502			PRG3	;PRG3 START ADDRESS
672	001166	014540			PRG4	;PRG4 START ADDRESS
673	001170	002410		EMTTAB:	TYP	;POINTER TO TYPEOUT ROUTINE
674	001172	002532			TYPS	;POINTER TO CHAINED MESSAGES ROUTINE
675	001174	002664			STAL	;POINTER TO RANDOM STALL ROUTINE
676	001176	001406			ERR	;POINTER TO ERROR ROUTINE
677	001200	001344			DTCHK	
678	001202	000000			OPEN	
679	001204	002240			STLSRV	
680	001206	002270			STLSPV	
681	001210	001332			EHLT	
682	001212	002320			SRSETT	
683	001214	001752			CHAINN	
684	001216	002140			SAVRG	
685	001220	002200			RSTRG	
686	001222	001430			ERR1	
687	001224	002616			DLY	
688						
689						
690						

;POINTER TO ERROR HALT ROUTINE.

691										
692	001226	000000								
693	001230	000000								
694	001232	000000								
695	001234	000000								
696	001236	000000								
697	001240	000000								
698	001242	000000								
699	001244	000000								
700	001246	000000								
701	001250	000000								
702	001252	000000								
703	001254	000000								
704	001256	000000								
705	001260	000000								
706	001262	000000								
707	001264	000000								
708	001266	000000								
709	001270	177740								
710	001272	104000								
711	001274	016353								
712	001276	000000								
713	001300	000207								
714	001302	104000								
715	001304	016451								
716	001306	000000								
717	001310	000207								
718	001312	104000								
719	001314	016506								
720	001316	032767	002000	176244						
721	001324	001401								
722	001326	000000								
723	001330	000207								
724										
725										
726	001332	005767	176232							
727	001336	100001								
728	001340	000000								
729	001342	000002								
730										
731										
732	001344	026767	177660	177660						
733	001352	001414								
734	001354	004567	001652							
735	001360	001230								
736	001362	016344								
737	001364	000003								
738	001366	004567	001640							
739	001372	001232								
740	001374	016332								
741	001376	000003								
742	001400	104015								
743	001402	016311								
744	001404	000002								
745										
746										

```

RCNT: OPEN
CRBUF: OPEN
CRBUFA: OPEN
CARMSK: OPEN
CHR1: OPEN
CHR2: OPEN
CHR3: OPEN
ERCTR: OPEN
CTRA: OPEN
CTRB: OPEN
CTRC: OPEN
CTRD: OPEN
TXCSRT: OPEN
RXCSRT: OPEN
RXBUFT: OPEN
TEMP: OPEN
SRT: OPEN
STLMSK: 177740
SETSR: TYPE ;TYPE SELECT OPTION MESSAGE.
      ASETSR ;COMMON HALT.
      HALT ;EXIT.
      RTS %7 ;TYPE INCORRECT ROUTINE SELECTED.
INCRTN: TYPE
      AINCRT ;COMMON HALT.
      HALT ;EXIT.
      RTS %7 ;TYPE PROGRAM END.
PRGEND: TYPE
      APGEND ;TEST END OF PROGRAM HALT OPTION
      BIT #BIT10,SR ;BRANCH IF NOT SELECTED
      BEQ .+4
      HALT
      RTS %7 ;EXIT.
;CONDITIONAL ERROR HALT ROUTINE.
EHLT: TST SR ;CHECK FOR HALT ON ERROR.
      BPL EHLTA ;BRANCH IF NO HALT DESIRED.
      HALT
EHLTA: RTI ;IN DATA LIGHTS.
;DATA CHECK ROUTINE.
DTCHK: CMP CRBUF,CRBUFA ;COMPARE EXPECTED AND RECEIVED
      BEQ DTCHKA ;CHARS. BRANCH IF SAME.
      JSR %5,0ACNV ;GO TO OCTAL TO ASCII CONVERT.
      CRBUF ;SOURCE ADDR.
      AWAS ;DESTINATION ADDR.
      3 ;#OF DIGITS TO CONVERT.
      JSR %5,0ACNV ;GO TO OCTAL TO ASCII CONVERT.
      CRBUFA ;SOURCE ADDR.
      AASB ;DESTINATION ADDR.
      3 ;#OF DIGITS TO CONVERT.
      ERROR1
      ERDAT
DTCHKA: RTI ;EXIT.

```

747	001406	012767	177777	000126	ERR:	MOV	#-1,ERRB	;SET UP ONE MESSAGE CALL.
748	001414	012767	000240	000122		MOV	#240,ERRB+2	
749	001422	005067	000132			CLR	ERRE	
750	001426	000413				BR	ERRA	
751	001430	011667	000106		ERR1:	MOV	%6,ERRB	;DEVELOP ADDT'L MESSAGE ADDR.
752	001434	017767	000102	000100		MOV	%ERRB,ERRB	;STORE AT ERRB.
753	001442	012767	177777	000074		MOV	#-1,ERRB+2	
754	001450	012767	000002	000102		MOV	#2,ERRE	
755	001456	032767	020000	176104	ERRA:	BIT	#BIT13,SR	;INHIBIT ERROR PRINT?
756	001464	001030				BNE	ERRC	;BRANCH TO INHIBIT PRINT.
757	001466	011667	000064			MOV	%6,ERRD	;DEVELOP CALLING ADDR.
758	001472	162767	000002	000056		SUB	#2,ERRD	
759	001500	004567	001526			JSR	%5,OACNV	;GO TO OCTAL TO ASCII CONVERT.
760	001504	001556				ERRD		;SOURCE ADDR.
761	001506	015220				APC		;DESTINATION ADDR.
762	001510	000006				6		;#OF DIGITS TO CONVERT.
763	001512	004567	001514			JSR	%5,OACNV	;GO TO OCTAL TO ASCII CONVERT.
764	001516	001140				PRGNUM		;SOURCE ADDR.
765	001520	015203				APNUMB		;DESTINATION ADDR.
766	001522	000002				2		;#OF DIGITS TO CONVERT.
767	001524	004567	001502			JSR	%5,OACNV	;GO TO OCTAL TO ASCII CONVERT.
768	001530	001146				RTNNO		;SOURCE ADDR.
769	001532	015211				ATNUMB		;DESTINATION ADDR.
770	001534	000003				3		;#OF DIGITS TO CONVERT.
771	001536	104001				TYPES		;TYPE:
772	001540	015200				EMO		;ERROR HEADER,
773	001542	000000			ERRB:	OPEN		;ADDT'L ERROR MESSAGE IF ANY.
774	001544	177777				-1		
775	001546	104010			ERRC:	EHALT		;GO ERR HALT IF DESIRED.
776	001550	066716	000004			ADD	ERRE,%6	
777	001554	000002				RTI		;EXIT.
778	001556	000000			ERRD:	OPEN		
779	001560	000000			ERRE:	OPEN		

780										
781	001562	012706	001076		START:	MOV	#SPBOT,%6		;SET BOTTOM OF SP STACK.	
782	001566	005067	176204			CLR	PSW			
783	001572	012767	000006	176204		MOV	#6,MACHER			
784	001600	005067	177342			CLR	RTNNO			
785	001604	016700	175760			MOV	SR,%0		;(SR) TO RD	
786	001610	042700	177770			BIC	#177770,%0		;LIMIT (SR) TO BITS 3-0	
787	001614	010067	177320			MOV	%0,PRGNUM		;SAVE PROGRAM #	
788	001620	006300				ASL	%0			
789	001622	012767	003576	176174		MOV	#PFAIL,24			
790	001630	012767	000340	176170		MOV	#PRTY7,26			
791	001636	000170	001156			JMP	JPBGTAB(0)		;GO TO SELECTED PROGRAM.	
792	001642	016767	177274	177300	GETRDY:	MOV	KSTART,NXTST		;ADDR OF 1ST ROUTINE TO NXTST	
793	001650	012767	000006	176126	GTRDYX:	MOV	#6,MACHER		;RESET MACHER TRAP.	
794	001656	005067	176114			CLR	PSW			
795	001662	012706	001076			MOV	#SPBOT,%6		;SET BOTTOM OF STACK.	
796	001666	104011				SRESET			;ISSUE RESET	
797	001670	004767	000142		GTRDYA:	JSR	%7,FORWD		;ROLL FORWARD TO "NEXT" ROUTINE.	
798	001674	032767	001000	175666	GTRDYB:	BIT	#BIT9,SR		;CHECK SELECT ROUTINE SWITCH	
799	001702	001003				BNE	GTRDYC		;BRANCH IF SELECT ROUTINE SWITCH IS SET.	
800	001704	000177	177234			JMP	JCURTST		;GO RUN CURRENT ROUTINE.	
801	001710	000437				BR	CHNB		;NO GO. MANUAL RTN BYPASSED.	
802	001712	016700	175652		GTRDYC:	MOV	SR,%0		;(SR) TO RD	
803	001716	042700	177600			BIC	#177600,%0		;MASK UNDESIRED BITS	
804	001722	126700	177220			CMPB	RTNNO,%0		;COMPARE RTNNO TO (RD)	
805	001726	001002				BNE	GTRDYD		;BRANCH IF ROUTINE NOT FOUND YET.	
806	001730	000177	177210			JMP	JCURTST		;GO RUN ROUTINE.	
807	001734	022767	177777	177206	GTRDYD:	CMP	#-1,NXTST		;NO. CHECK FOR LAST ROUTINE.	
808	001742	001352				BNE	GTRDYA		;BRANCH IF NOT LAST ROUTINE.	
809	001744	004767	177332			JSR	%7,INCRN		;YES. INCORRECT ROUTINE SELECTED.	
810	001750	000734				BR	GETRDY		;START OVER.	
811	001752	032767	040000	175610	CHAINN:	BIT	#BIT14,SR		;CHECK FOR SCOPE OPTION.	
812	001760	001403				BEG	CHNA		;BRANCH IF SCOPE SW NOT SET.	
813	001762	016716	177166		CHNAB:	MOV	SCOPTR,%6		;SET UP TO RETURN TO ROUTINE.	
814	001766	000032				RTI			;RETURN TO ROUTINE.	
815	001770	032767	004000	175572	CHNA:	BIT	#BIT11,SR		;TEST INHIBIT ITERATION SWITCH	
816	001776	001003				BNE	CHNAA		;BRANCH IF INHIBIT ITERATION SW SET.	
817	002000	005367	177146			DEC	ICTR		;DECREMENT ITERATION COUNT.	
818	002004	001366				BNE	CHNAB		;BRANCH IF COUNT NOT 0.	
819	002006	022626			CHNAA:	POPSP2			;POP STACK TWICE	
820										
821	002010	032767	001000	175552	CHNB:	BIT	#BIT9,SR		;CHECK SELECT ROUTINE SWITCH	
822	002016	001311				BNE	GETRDY		;BRANCH IF SELECT RTN SW SET	
823	002020	022767	177777	177122		CMP	#-1,NXTST		;LAST TEST?	
824	002026	001310				BNE	GTRDYX		;BRANCH IF NOT LAST TEST.	
825	002030	004767	177256			JSR	%7,PRGEND		;PROGRAM END.	
826	002034	000702				BR	GETRDY			
827	002036	016705	177106		FORWD:	MOV	NXTST,%5		;ADDR OF NEXT ROUTINE TO R5.	
828	002042	012567	177100			MOV	(5)+,RTNNO		;GET NEXT ROUTINE NUMBER.	
829	002046	012567	177076			MOV	(5)+,NXTST		;GET ADDR OF NEXT "NEXT" ROUTINE.	
830	002052	012567	177074			MOV	(5)+,ICTR		;GET ITERATION COUNT.	
831	002056	012567	177072			MOV	(5)+,SCOPTR		;GET SCOPE LOOP ENTRY POINTER.	
832	002062	010567	177056		FORWDA:	MOV	%5,CURTST		;ADDR OF NOW CURRENT TEST TO CURTST.	
833	002066	000207				RTS	%7		;EXIT FORWD SUBROUTINE.	
834	002070	012767	177777	177056	FORWDB:	MOV	#-1,SCOPTR		;FORCE "NO SCOPE"	
835	002076	012767	000001	177046		MOV	#1,ICTR		;FORCE I COUNT OF 1	

```

836 002104 000766          BR      FORWDA
837 002106 011646          EMTINT: MOV    @%6,-(6)      ;GET SAVED PC.
838 002110 162716 000002          SUB    #2,@%6      ;DECREMENT PC BY 2.
839 002114 017616 000000          MOV    @%6,@%6
840 002120 006316          EMTA:  ASL    @%6      ;EMT ARG X 2.
841 002122 042716 177001          BIC    #177001,@%6  ;REMOVE 7 MSB.
842 002126 062716 001170          ADD    #EMTTAB,@%6  ;FORM EMT RTN ADDR.
843 002132 017616 000000          MOV    @%6,@%6
844 002136 000136          JMP    @%6+         ;GO TO EMT ROUTINE.
845
846          ;SAVE REGS 0 TO 4 SUBROUTINE.
847 002140 012667 000030          SAVRG: MOV    (6)+,SVRPC      ;SAVE PC AND PSW.
848 002144 012667 000026          MOV    (6)+,SVRPSW
849 002150 010446          MOV    %4,-(6)      ;SAVE REGS 0 - 4
850 002152 010346          MOV    %3,-(6)      ;IN STACK.
851 002154 010246          MOV    %2,-(6)
852 002156 010146          MOV    %1,-(6)
853 002160 010046          MOV    %0,-(6)
854 002162 016746 000010          MOV    SVRPSW,-(6)  ;RESTORE PC AND PSW.
855 002166 016746 000002          MOV    SVRPC,-(6)
856 002172 000002          RTI
857 002174 000000          SVRPC: OPEN
858 002176 000000          SVRPSW: OPFN
859          ;RESTORE REGS 0 TO 4 SUBROUTINE.
860 002200 012667 000030          RSTRG: MOV    (6)+,RSTPC      ;SAVE PC AND PSW.
861 002204 012667 000026          MOV    (6)+,RSTPSW
862 002210 012600          MOV    (6)+,%0      ;RESTORE REGS 0 - 4
863 002212 012601          MOV    (6)+,%1      ;FROM STACK.
864 002214 012602          MOV    (6)+,%2
865 002216 012603          MOV    (6)+,%3
866 002220 012604          MOV    (6)+,%4
867
868 002222 016746 000010          MOV    RSTPSW,-(6)  ;RESTORE PC AND PSW.
869 002226 016746 000002          MOV    RSTPC,-(6)
870 002232 000002          RTI
871 002234 000000          RSTPC: OPEN
872 002236 000000          RSTPSW: OPEN
873          ;ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
874 002240 017667 000000 000012          STLSRV: MOV    @%6,STPRA+2  ;MOVE VECTOR ADDR TO STPRA+2
875 002246 062716 000002          ADD    #2,@%6      ;SET UP EXIT
876 002252 016701 176632          MOV    RXVTR,%1
877 002256 012721 000000          STPRA: MOV    #OPEN,(1)+  ;SET VECTOR ADDRESS
878 002262 016721 176624          MOV    RXLVL,(1)+  ;SET PRIORITY
879 002266 000002          RTI
880          ;ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
881 002270 017667 000000 000012          STLSPV: MOV    @%6,STPPA+2  ;MOVE VECTOR ADDR TO STPPA+2
882 002276 062716 000002          ADD    #2,@%6      ;SET UP EXIT
883 002302 016701 176606          MOV    TXVTR,%1
884 002306 012721 000000          STPPA: MOV    #OPEN,(1)+  ;SET VECTOR ADDRESS.
885 002312 016721 176600          MOV    TXLVL,(1)+  ;SET PRIORITY
886 002316 000002          RTI
887          ;ROUTINE TO ISSUE RESET.
888 002320 012700 052525          SRSETT: MOV    #52525,%0  ;DATA TO R0.
889 002324 005100          COM    %0          ;COMPLEMENT (R0).
890 002326 010067 177770          MOV    %0,SRSETT+2  ;(R0) TO SRSETT+2.
891 002332 000005          RESET          ;ISSUE RESET. (R0) IS

```

```

892 002334 000002 RTI ;DISPLAYED. EXIT.
893
894 ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
895 002336 016700 000042 RNGEN: MOV RP1,%0
896 002342 006100 ROL %0
897 002344 006100 ROL %0
898 002346 066700 000034 ADD RP2,%0
899 002352 010067 000026 MOV %0,RP1
900 002356 006100 ROL %0
901 002360 006100 ROL %0
902 002362 066700 000020 ADD RP2,%0
903 002366 006100 ROL %0
904 002370 006100 ROL %0
905 002372 010067 000010 MOV %0,RP2
906 002376 016700 000002 MOV RP1,%0
907 002402 000207 RTS ;EXIT. NUMBER IN R0
908 002404 001233 RP1: 1233
909 002406 007622 RP2: 7622
910 ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
911 002410 011600 TYP: MOV @%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
912 002412 062716 000002 ADD #2,@%6 ;SET UP EXIT.
913 002416 011000 MOV @%0,%0 ;ADDRESS OF MESSAGE TO R0.
914 002420 112067 000104 TYPA: MOVB (0)+,TYPDAT ;GET CHARACTER
915 002424 122767 000100 000076 CMPB #100,TYPDAT ;CHECK FOR"@"CHARACTER
916 002432 001001 BNE TYPC ;BRANCH IF NOT"@".
917 002434 000002 RTI ;TERMINATOR CHAR. DONE. EXIT.
918 002436 122767 000045 000064 TYPC: CMPB #45,TYPDAT ;CHECK FOR"%".
919 002444 001416 BEQ TYPF ;BRANCH IF"%".
920 002446 122767 000043 000054 CMPB #43,TYPDAT ;NOT"%".CHECK FOR"#".
921 002454 001417 BEQ TYPG ;BRANCH IF"#".
922 002456 004767 000002 JSR %7,TYPD ;TYPE CHAR IN TYPDAT
923 002462 000756 BR TYPA
924 002464 116777 000040 176434 TYPD: MOVB TYPDAT,@TPB ;OUTPUT CHARACTER TO PRINTER
925 002472 105777 176426 TSTB @TPS ;WAIT FOR DONE FLAG.
926 002476 100375 BPL .-4
927 002500 000207 RTS ;EXIT
928 002502 112767 000015 000020 TYPF: MOVB #15,TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
929 002510 004767 177750 JSR %7,TYPD ;GO TYPE CHAR.
930 002514 112767 000012 000006 TYPG: MOVB #12,TYPDAT ;MOVE LF CODE TO TYPDAT.
931 002522 004767 177736 JSR %7,TYPD ;GO TYPE CHAR.
932 002526 000734 BR TYPA
933 002530 000000 TYPDAT: OPEN
934 ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
935 002532 011600 TYP5: MOV @%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
936 002534 062716 000002 ADD #2,@%6 ;UPDATE TO NEXT MESSAGE ADDRESS
937 002540 011067 000014 MOV @%0,TYPSB ;ADDRESS OF MESSAGE TO TYPSB
938 002544 022767 177777 000006 CMP #-1,TYPSB ;CHECK FOR TERMINATOR
939 002552 001001 BNE TYP5A ;BRANCH IF NOT TERMINATOR.
940 002554 000002 RTI ;TERMINATOR. EXIT
941 002556 104000 TYP5A: TYPE ;CALL ON TYP SUB TO TYPE MESSAGE
942 002560 000000 TYP5B: OPEN ;ADDRESS OF MESSAGE GOES HERE
943 002562 000763 BR TYP5 ;GO PROCESS NEXT MESSAGE
944
945 002564 012701 000300 OVLAY: MOV #300,%1 ;GET DC11 VECTOR BASE ADDRESS
946 002570 012702 000302 MOV #302,%2 ;GET NEXT ADDRESS
947 002574 010221 OVL5A: MOV %2,(1)+ ;LOAD VECTOR WITH ADDRESS OF NEXT ADDRESS

```

948	002576	005021		CLR	(1)+		;PUT A HALT IN THE NEXT ADDRESS
949	002600	020267	176174	CMP	%2,1000		;ALL VECTORS BEEN LOADED
950	002604	001403		BEQ	OVRLYB		
951	002606	062702	000004	ADD	#4,%2		;GET NEXT VECTOR ADDRESS
952	002612	000770		BR	OVRLYA		
953	002614	000207		OVRLYB: RTS	7		;EXIT
954							
955							;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
956	002616	011667	000040	DLY: MOV	%6,DLCNT		;GET DELAY COUNT ADDRESS.
957	002622	062716	000002	ADD	#2,%6		;SET UP EXIT ADDRESS
958	002626	017746	000030	MOV	%DLCNT,-(6)		;DELAY COUNT TO STACK
959	002632	001411		BEQ	DLYC		
960	002634	005067	175136	CLR	PSW		;SET PRIORITY 0
961	002640	012746	000554	DLYA: MOV	#554,-(6)		;1 MSEC COUNT TO STACK
962	002644	005316		DLYB: DEC	%6		;DECREMENT 1 MSEC COUNT
963	002646	001376		BNE	DLYB		;BRANCH IF NOT 0.
964	002650	005726		POPSP			;ZERO. UNCOVER MSECS. COUNT.
965	002652	005316		DEC	%6		;DECREMENT IT
966	002654	001371		BNE	DLYA		;BR IF NOT DONE DELAYING
967	002656	005726		DLYC: POPSP			;DONE
968	002660	000002		RTI			;EXIT.
969	002662	000000		DLCNT: OPEN			;CONTAINS MILLISECONDS COUNT ADDRESS.
970							;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
971							;DETERMINED BY CONTENTS OF LOC STLMSK.
972	002664	004767	177446	STAL: JSR	%7,RNGEN		;GO GET RANDOM NUMBER.
973	002670	046700	176374	BIC	STLMSK,%0		;# IN RD. APPLY STALL MASK.
974	002674	001404		BEQ	STALB		;BRANCH IF RESULT IS 0.
975	002676	010067	000002	MOV	%0,STALA		
976	002702	104016		DELAY			;DELAY
977	002704	000000		STALA: OPEN			;DELAY COUNT
978	002706	000002		STALB: RTI			;DONE. EXIT.
979							;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
980	002710	004767	177422	GRCNT: JSR	%7,RNGEN		;GET RANDOM NUMBER
981	002714	046700	000010	BIC	RCMSK,%0		;APPLY MASK
982	002720	001773		BEQ	GRCNT		;TRY AGAIN IF RESULT 0
983	002722	010067	000004	MOV	%0,RNCNT		;COUNT TO RNCNT
984	002726	000207		RTS	%7		;EXIT.
985	002730	000000		RCMSK: OPEN			;RANDOM CHARACTER MASK.
986	002732	000000		RNCNT: OPEN			;RANDOM CHARACTER COUNT.
987							;SUBROUTINE TO SELECT LINE AND
988	002734	104000		LINSEL: TYPE			
989	002736	016730		LDLINE			
990	002740	000000		HALT			
991	002742	016701	174622	MOV	SR,%1		
992	002744	042701	177407	BIC	#177407,%1		
993	002752	010167	176306	MOV	%1,TEMP		
994	002756	012702	000770	MOV	#770,%2		
995	002762	012703	001100	MOV	#RXCSR,%3		
996	002766	012704	000004	MOV	#4,%4		
997	002772	040213		BIC	%2,(3)		
998	002774	050123		BIS	%1,(3)+		
999	002776	005304		DEC	%4		
1000	003000	001374		BNE	.-6		
1001	003002	006201		ASR	%1		;POSITION SELECTED LINE
1002	003004	006201		ASR	%1		
1003	003006	016101	015100	MOV	VECTAB(1),%1		;GET LINE VECTOR ADDRESS

1004	003012	010167	176072		MOV	%1,RXVTR	;LOAD INTO PROGRAM RXVTR
1005	003016	022121			CMP	(1)+,(1)+	;ADD +4
1006	003020	010167	176070		MOV	%1,TXVTR	;LOAD INTO PROGRAM TXVTR
1007	003024	006267	176234		ASR	TEMP	
1008	003030	006267	176230		ASR	TEMP	
1009	003034	006267	176224		ASR	TEMP	
1010	003040	004567	000166		JSR	5,OACNV	;TYPE LINE #
1011	003044	001264			TEMP		
1012	003046	017001			SELINE		
1013	003050	000002			2		
1014	003052	104000			TYPE		
1015	003054	016767			ALINE		
1016	003056	000205			RTS	5	
1017					;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS		
1018	003060	012767	177777	000014	INBIN:	MOV	#-1,RIND ;SET ALL VARIABLES
1019	003066	004567	000222			JSR	%5,BMOVE ;TO MINUS 1.
1020	003072	003102				RIND	
1021	003074	003103				RIND+1	
1022	003076	000013				11.	
1023	003100	000207				RTS	%7 ;EXIT
1024	003102	000000			RIND:	OPEN	
1025	003104	000000			PTO:	OPEN	
1026	003106	000000			PT1:	OPEN	
1027	003110	000000			PIND:	OPEN	
1028	003112	000000			PTOP:	OPEN	
1029	003114	000000			PT1P:	OPEN	
1030					;SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN RO		
1031	003116	016767	177762	177762	GTBIN:	MOV	PTO,PT1 ;PREVIOUS BIN CHAR TO PT1
1032	003124	005167	177756			COM	PT1
1033	003130	005167	177746			COM	RIND
1034	003134	001002				BNE	.+6
1035	003136	005267	177744			INC	PT1
1036	003142	042767	177400	177736		BIC	#177400,PT1 ;MASK TO 8 BITS
1037	003150	016767	177732	177726		MOV	PT1,PTO ;SAVE BIN CHAR IN PTO
1038	003156	016700	177724			MOV	PT1,%0 ;BIN CHAR TO RO.
1039	003162	000207				RTS	%7 ;EXIT.
1040	003164	016767	177722	177722	GTBINP:	MOV	PTOP,PT1P ;PREVIOUS BIN CHAR TO PT1P
1041	003172	005167	177716			COM	PT1P
1042	003176	005167	177706			COM	PIND
1043	003202	001002				BNE	.+6
1044	003204	005267	177704			INC	PT1P
1045	003210	042767	177400	177676		BIC	#177400,PT1P ;MASK TO 8 BITS.
1046	003216	016767	177672	177666		MOV	PT1P,PTOP ;SAVE BIN CHAR IN PTO.
1047	003224	016701	177664			MOV	PT1P,%1 ;BIN CHAR TO R1.
1048	003230	000207				RTS	%7 ;EXIT.
1049					;OCTAL TO ASCII CONVERT ROUTINE		
1050	003232	013567	000054		OACNV:	MOV	2(5)+,OACNVX ;GET OCTAL VALUE.
1051	003236	012501				MOV	(5)+,%1 ;GET DESTINATION ADDR.
1052	003240	012502				MOV	(5)+,%2 ;GET CONVERT COUNT.
1053	003242	060201				ADD	%2,%1 ;DEVELOP ADDR TO STORE 1ST CHAR.
1054	003244	016703	000042		OACNVA:	MOV	OACNVX,%3
1055	003250	042703	177770			BIC	#177770,%3 ;ISOLATE LEAST SIGNIFICANT DIGIT.
1056	003254	062703	000060			ADD	#60,%3 ;CONVERT DIGIT TO ASCII.
1057	003260	110341				MOVB	%3,-(1) ;STORE ASCII CHARACTER.
1058	003262	042767	000007	000022		BIC	#7,OACNVX
1059	003270	006067	000016			ROR	OACNVX

1060	003274	006067	000012		ROR	OACNVX	
1061	003300	006067	000006		ROR	OACNVX	
1062	003304	005302			DEC	%2	; DONE ALL DIGITS?
1063	003306	001356			BNE	OACNVA	; BRANCH IF NOT DONE.
1064	003310	000205			RTS	%5	; DONE. EXIT.
1065	003312	000000			OACNVX: OPEN		
1066					; SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.		
1067	003314	104013			BMOVE: SAVREG		; SAVE REGS.
1068	003316	012501			MOV	(5)+,%1	; GET "FROM" ADDRESS
1069	003320	012502			MOV	(5)+,%2	; GET "TO" ADDRESS
1070	003322	012503			MOV	(5)+,%3	; GET COUNT
1071	003324	112122			BMOVA: MOV B	(1)+,(2)+	; MOVE BYTE
1072	003326	005303			DEC	%3	; DECREMENT COUNT
1073	003330	001375			BNE	BMOVA	; BRANCH IF NOT DONE.
1074	003332	104014			RSTREG		; RESTORE REGS.
1075	003334	000205			RTS	%5	; DONE EXIT
1076					; BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.		
1077	003336	012700	003456		BDCNV: MOV	#DECVAL,%0	; SET UP ADDR TO STORE DECIMAL ASCII IN R0
1078	003342	013501			MOV	2(5)+,%1	; BINARY VALUE TO R1.
1079	003344	012702	003444		MOV	#ADTENP,%2	; ADDR OF TEN POWER STRING TO R2.
1080	003350	012767	000005	000060	MOV	#5,CNVCTR	; SET UP FOR 5 POWER CONVERSIONS.
1081	003356	012267	000060		BDCNVA: MOV	(2)+,TENPWR	; MOVE POWER OF TEN VALUE TO TENPWR.
1082	003362	004767	000010		JSR	%7,SUBTEN	; PERFORM CONVERSION
1083	003366	005367	000044		DEC	CNVCTR	; DONE 5 CONVERSIONS?
1084	003372	001371			BNE	BDCNVA	; BRANCH IF NOT YET 5.
1085	003374	000205			RTS	%5	; YES, EXIT.
1086	003376	005067	000036		SUBTEN: CLR	DIGIT	; CLEAR DIGIT
1087	003402	166701	000034		SUBTNA: SUB	TENPWR,%1	; SUBTRACT TEN POWER FROM BINARY VALUE.
1088	003406	103403			BCS	SUBTNB	; BRANCH IF UNSUCCESSFUL SUBTRACTION.
1089	003410	005267	000024		INC	DIGIT	
1090	003414	000772			BR	SUBTNA	
1091	003416	066701	000020		SUBTNB: ADD	TENPWR,%1	; RESTORE SUBTRACTED VALUE.
1092	003422	062767	000060	000010	ADD	#60,DIGIT	; CONVERT (DIGIT) TO ASCII
1093	003430	116720	000004		MOVB	DIGIT,(0)+	; MOVE ASCII CHAR TO DECVAL FIELD.
1094	003434	000207			RTS	%7	; EXIT.
1095	003436	000000			CNVCTR: OPEN		
1096	003440	000000			DIGIT: OPEN		
1097	003442	000000			TENPWR: OPEN		
1098	003444	023420			ADTENP: 10000.		
1099	003446	001750			1000.		
1100	003450	000144			100.		
1101	003452	000012			10.		
1102	003454	000001			1		
1103	003456	040	040	040	DECVAL: .BYTE	040,040,040,040,040,040	
1104	003461	040	040	040			
1105	003464	012567	175576		DATTST: MOV	(5)+,SRT	; GET PARAMETERS
1106	003470	004767	011216		JSR	7,STPARB	; LOAD PARAMETERS
1107	003474	042777	000001	175376	BIC	#BIT0,@RXCSR	; CLEAR DATA TERM. READY
1108	003502	052777	000004	175374	BIS	#BIT2,@TXCSR	; SET MAINTENANCE BIT
1109	003510	012767	000144	175530	MOV	#100,CTRA	; GET CHARACTER COUNT
1110	003516	105777	175362		DATAA: TSTB	@TXCSR	; WAIT FOR
1111	003522	100375			BPL	-4	; READY FLAG
1112	003524	004767	177434		JSR	7,GTBINP	; GET CHARACTER
1113	003530	110167	175476		MOVB	%1,CBUBA	; MOVE CHARACTER
1114	003534	046767	175474	175470	BIC	CARMSK,CBUBA	; MASK OFF NON TRANSMITTED BITS
1115	003542	110177	175340		MOVB	%1,@TXBUF	; TRANSMIT CHARACTER

1116	003546	105777	175326		TSTB	QRXCSR		;WAIT FOR
1117	003552	100375			BPL	.-4		;DONE FLAG
1118	003554	117767	175322	175446	MOVB	QRXBUF,CRBUF		;GET RECEIVED CHARACTER
1119	003562	104004			DATCHK			;CHK DATA
1120	003564	005367	175456		DEC	CTRA		;DECREMENT CHARACTER COUNT
1121	003570	001352			BNE	DATAA		
1122	003572	005726			TST	(6)+		;POP STACK
1123	003574	104012			SCOPE			
1124								
1125	003576	012767	003606	174220	PFAIL: MOV	#PWRUP,24		
1126	003604	000000			HALT			
1127								
1128	003606	000005			PWRUP: RESET			
1129	003610	012706	001076		MOV	#SPBOT,%6		
1130	003614	104003			ERROR			
1131	003616	016700	175316		RESTART: MOV	PRGNUM,%0		;GET PROGRAM NUMBER
1132	003622	006300			ASL	%0		
1133	003624	012767	003576	174172	MOV	#PFAIL,24		;RELOAD POWER FAIL VECTOR
1134	003632	004767	011054		JSR	7,STPARB		;RELOAD LINE PARAMETERS
1135	003636	000170	003642		JMP	QRSTART(0)		;GO RESTART SELECTED PROGRAM
1136								
1137	003642	003700			RSTART: PRG0A			;PROGRAM 0 RESTART ADDRESS
1138	003644	014364			PRG1A			;PROGRAM 1 RESTART ADDRESS
1139	003646	014430			PRG2A			;PROGRAM 2 RESTART ADDRESS
1140	003650	014524			PRG3A			;PROGRAM 3 RESTART ADDRESS
1141	003652	014560			PRG4A			;PROGRAM 4 RESTART ADDRESS

MO2

```

1142
1143
1144 003654 012767 003704 175260 :PRGO - INPUT-OUTPUT LOGIC TESTS
1145 003662 104000 PRGO: MOV #ATO KSTART
1146 003664 015231 TYPE ;TYPE TITLE AND INSTRUCTIONS
1147 003666 000000 POTIT
1148 003670 004567 177040 JSR 5,LINSEL ;GO GET LINE # FROM USER
1149 003674 004767 175372 JSR 7,SETSR
1150 003700 000167 175736 PRGOA: JMP GETRDY ;GET STARTED.
1151 177777 TX=-1
1152
1153 003704 000000 ATO: 0 ;TEST NUMBER 0 *
1154 003706 003736 AT1 ;ADDRESS OF NEXT TEST *
1155 003710 001750 1000. ;TEST ITERATION COUNT *
1156 003712 003714 AAA ;SCOPE ENTRY POINT *
1157
1158 ;*****
1158 ;TEST ABILITY TO REFERENCE RECEIVER CSR WITHOUT TRAPPING
1159 003714 012767 003730 174062 AAA: MOV #AAE,MACHER ;SET UP MACHINE ERROR TRAP.
1160 003722 005777 175152 TST @RXCSR ;REFERENCE RXCSR
1161 003726 104012 AAB: SCOPE ;OK IF NO TRAP. SCOPE
1162 003730 022626 AAE: POPSP2
1163 003732 104003 ERROR ;TRAPPED WHEN REFERENCING RXCSR.
1164 003734 000774 BR AAB
1165
1166 003736 000001 AT1: 1 ;TEST NUMBER 1 *
1167 003740 003770 AT2 ;ADDRESS OF NEXT TEST *
1168 003742 001750 1000. ;TEST ITERATION COUNT *
1169 003744 003746 ABA ;SCOPE ENTRY POINT *
1170
1171 ;*****
1171 ;TEST ABILITY TO REFERENCE RECEIVER BUFFER WITHOUT TRAPPING
1172 003746 012767 003762 174030 ABA: MOV #ABE,MACHER ;SET UP MACHINE ERROR TRAP.
1173 003754 005777 175122 TST @RXBUF ;REFERENCE RXBUF
1174 003760 104012 ABB: SCOPE ;OK IF NO TRAP SCOPE
1175 003762 022626 ABE: POPSP2
1176 003764 104003 ERROR ;TRAPPED WHEN REFERENCING RXBUF
1177 003766 000774 BR ABB
1178
1179 ;*****
1180 003770 000002 AT2: 2 ;TEST NUMBER 2 *
1181 003772 004022 AT3 ;ADDRESS OF NEXT TEST *
1182 003774 001750 1000. ;TEST ITERATION COUNT *
1183 003776 004000 ACA ;SCOPE ENTRY POINT *
1184 000000 TX=TX+1
1185
1186 ;*****
1186 ;TEST ABILITY TO REFERENCE TRANSMITTER CSR WITHOUT TRAPPING.
1187 004000 012767 004014 173776 ACA: MOV #ACE,MACHER ;SET UP MACHINE ERROR TRAP.
1188 004006 005777 175072 TST @TXCSR ;REFERENCE TXCSR
1189 004012 104012 ACB: SCOPE ;SCOPE
1190 004014 022626 ACE: POPSP2
1191 004016 104003 ERROR ;TRAPPED WHEN REFERENCING TXCSR
1192 004020 000774 BR ACB
1193
  
```

```

1194
1195
1196 004022 000003
1197 004024 004054
1198 004026 001750
1199 004030 004032
1200
1201
1202 004032 012767 004046 173744
1203 004040 005777 175042
1204 004044 104012
1205 004046 022626
1206 004050 104003
1207 004052 000774
1208
1209
1210 004054 000004
1211 004056 004154
1212 004060 000144
1213 004062 004064
1214
1215
1216 004064 032777 000001 175012
1217 004072 001402
1218 004074 104003
1219 004076 000421
1220 004100 052777 000001 174776
1221 004106 032777 000001 174770
1222 004114 001002
1223 004116 104003
1224 004120 000410
1225 004122 042777 000001 174754
1226 004130 032777 000001 174746
1227 004136 001401
1228 004140 104003
1229 004142 052777 000001 174734
1230 004150 104011
1231 004152 104012
1232
1233 004154 000005
1234 004156 004252
1235 004160 000144
1236 004162 004164
1237
1238
1239 004164 042777 000001 174706
1240 004172 032777 000002 174704
1241 004200 001402
1242 004202 104003
1243 004204 000421
1244 004206 052777 000001 174664
1245 004214 032777 000002 174662
1246 004222 001002
1247 004224 104003
1248 004226 000410
1249 004230 042777 000001 174642

```

```

;*****
AT3: 3 ;TEST NUMBER 3 *
      AT4 ;ADDRESS OF NEXT TEST *
      1000. ;TEST ITERATION COUNT *
      ADA ;SCOPE ENTRY POINT *
;*****
;TEST ABILITY TO REFERENCE TRANSMITTER BUFFER WITHOUT TRAPPING
ADA: MOV #ADE,MACHER ;SET UP MACHINE ERROR TRAP.
      TST @TXBUF ;REFERENCE TX BUF.
ADB: SCOPE ;SCOPE
ADE: POPSP2
      ERROR ;TRAPPED WHEN REFERENCING TXBUF
      BR ADB
;*****
AT4: 4 ;TEST NUMBER 4 *
      AT5 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AEA ;SCOPE ENTRY POINT *
;*****
;TEST THAT TXCSR BIT0 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AEA: BIT #BIT0,@TXCSR ;SEE IF TXCSR BIT0 IS CLEAR.
      BEQ AEB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT0
      BR AED
AEB: BIS #BIT0,@TXCSR ;SET TXCSR BIT0.
      BIT #BIT0,@TXCSR ;SEE IF BIT IS SET.
      BNE AEC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT0 FAILED TO SET.
      BR AED
AEC: BIC #BIT0,@TXCSR ;CLEAR TXCSR BIT0
      BIT #BIT0,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ AED
      ERROR ;TXCSR BIT0 FAILED TO CLEAR.
AED: BIS #BIT0,@TXCSR ;SET TXCSR BIT0.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE
;*****
AT5: 5 ;TEST NUMBER 5 *
      AT6 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AFA ;SCOPE ENTRY POINT *
;*****
;TEST THAT TXCSR BIT1 (CLEAR TO SEND) CAN BE SET, AND CLEARED
AFA: BIC #BIT0,@TXCSR ;CLEAR DATA TERMINAL READY
      BIT #BIT1,@TXCSR ;SEE IF TXCSR BIT1 IS CLEAR.
      BEQ AFB ;BRANCH IF BIT IS CLEAR.
      ERROR ;TXCSR BIT1 IS NOT CLEAR.
      BR AFD
AFB: BIS #BIT0,@TXCSR ;SET DATA TERM. RDY. (SETS CTS VIA JUMPER)
      BIT #BIT1,@TXCSR ;IS CLEAR TO SEND SET?
      BNE AFC ;BRANCH IF SET
      ERROR ;CTS NOT BEING SET VIA DTR
      BR AFD ;EXIT TEST
AFC: BIC #BIT0,@TXCSR ;CLEAR DATA TERM. RDY.

```

1250	004236	032777	000002	174640
1251	004244	001401		
1252	004246	104003		
1253	004250	104012		

AFD:

BIT
BEQ
ERROR
SCOPE

#BIT1,3TXCSR
AFD

:IS CTS CLEAR?
:CTS FAILED TO CLEAR VIA DTR
:SCOPE

```

1254
1255
1256 004252 000006
1257 004254 004352
1258 004256 000144
1259 004260 004262
1260
1261
1262 004262 032777 000004 174614
1263 004270 001402
1264 004272 104003
1265 004274 000421
1266 004276 052777 000004 174600
1267 004304 032777 000004 174572
1268 004312 001002
1269 004314 104003
1270 004316 000410
1271 004320 042777 000004 174556
1272 004326 032777 000004 174550
1273 004334 001401
1274 004336 104003
1275 004340 052777 000004 174536
1276 004346 104011
1277 004350 104012
1278
1279 004352 000007
1280 004354 004452
1281 004356 000144
1282 004360 004362
1283
1284
1285 004362 032777 000010 174514
1286 004370 001402
1287 004372 104003
1288 004374 000421
1289 004376 052777 000010 174500
1290 004404 032777 000010 174472
1291 004412 001002
1292 004414 104003
1293 004416 000410
1294 004420 042777 000010 174456
1295 004426 032777 000010 174450
1296 004434 001401
1297 004436 104003
1298 004440 052777 000010 174436
1299 004446 104011
1300 004450 104012

;*****
AT6: 6 ;TEST NUMBER 6 *
      AT7 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AGA ;SCOPE ENTRY POINT *
;*****
;TEST THAT TXCSR BIT2 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AGA: BIT #BIT2,@TXCSR ;SEE IF TXCSR BIT2 IS CLEAR.
      BEQ AGB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT2
      BR AGD
AGB: BIS #BIT2,@TXCSR ;SET TXCSR BIT2.
      BIT #BIT2,@TXCSR ;SEE IF BIT IS SET.
      BNE AGC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT2 FAILED TO SET.
      BR AGD
AGC: BIC #BIT2,@TXCSR ;CLEAR TXCSR BIT2
      BIT #BIT2,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ AGD
      ERROR ;TXCSR BIT2 FAILED TO CLEAR.
AGD: BIS #BIT2,@TXCSR ;SET TXCSR BIT2.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE
;*****
AT7: 7 ;TEST NUMBER 7 *
      AT10 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AHA ;SCOPE ENTRY POINT *
;*****
;TEST THAT TXCSR BIT3 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AHA: BIT #BIT3,@TXCSR ;SEE IF TXCSR BIT3 IS CLEAR.
      BEQ AHB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT3
      BR AHD
AHB: BIS #BIT3,@TXCSR ;SET TXCSR BIT3.
      BIT #BIT3,@TXCSR ;SEE IF BIT IS SET.
      BNE AHC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT3 FAILED TO SET.
      BR AHD
AHC: BIC #BIT3,@TXCSR ;CLEAR TXCSR BIT3
      BIT #BIT3,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ AHD
      ERROR ;TXCSR BIT3 FAILED TO CLEAR.
AHD: BIS #BIT3,@TXCSR ;SET TXCSR BIT3.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE

```

```

1301
1302
1303 004452 000010
1304 004454 004552
1305 004456 000144
1306 004460 004462
1307
1308
1309 004462 032777 000020 174414
1310 004470 001402
1311 004472 104003
1312 004474 000421
1313 004476 052777 000020 174400
1314 004504 032777 000020 174372
1315 004512 001002
1316 004514 104003
1317 004516 000410
1318 004520 042777 000020 174356
1319 004526 032777 000020 174350
1320 004534 001401
1321 004536 104003
1322 004540 052777 000020 174336
1323 004546 104011
1324 004550 104012
1325
1326 004552 000011
1327 004554 004660
1328 004556 000144
1329 004560 004562
1330
1331
1332 004562 012767 000340 173206
1333 004570 032777 000100 174306
1334 004576 001402
1335 004600 104003
1336 004602 000421
1337 004604 052777 000100 174272
1338 004612 032777 000100 174264
1339 004620 001002
1340 004622 104003
1341 004624 000410
1342 004626 042777 000100 174250
1343 004634 032777 000100 174242
1344 004642 001401
1345 004644 104003
1346 004646 052777 000100 174230
1347 004654 104011
1348 004656 104012
1349

```

```

*****
AT10: 10 ;TEST NUMBER 10 *
      AT11 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AIA ;SCOPE ENTRY POINT *
*****
;TEST THAT TXCSR BIT4 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AIA: BIT #BIT4,@TXCSR ;SEE IF TXCSR BIT4 IS CLEAR.
      BEQ AIB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT4
      BR AID
AIB: BIS #BIT4,@TXCSR ;SET TXCSR BIT4.
      BIT #BIT4,@TXCSR ;SEE IF BIT IS SET.
      BNE AIC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT4 FAILED TO SET.
      BR AID
AIC: BIC #BIT4,@TXCSR ;CLEAR TXCSR BIT4
      BIT #BIT4,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ AID
      ERROR ;TXCSR BIT4 FAILED TO CLEAR.
AID: BIS #BIT4,@TXCSR ;SET TXCSR BIT4.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE
*****
AT11: 11 ;TEST NUMBER 11 *
      AT12 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AJA ;SCOPE ENTRY POINT *
*****
;TEST THAT TXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AJA: MOV #PRTY7,PSW ;SET PRIORITY 7.
      BIT #BIT6,@TXCSR ;SEE IF TXCSR BIT6 IS CLEAR.
      BEQ AJB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT6
      BR AJD
AJB: BIS #BIT6,@TXCSR ;SET TXCSR BIT6.
      BIT #BIT6,@TXCSR ;SEE IF BIT IS SET.
      BNE AJC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT6 FAILED TO SET.
      BR AJD
AJC: BIC #BIT6,@TXCSR ;CLEAR TXCSR BIT6
      BIT #BIT6,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ AJD
      ERROR ;TXCSR BIT6 FAILED TO CLEAR.
AJD: BIS #BIT6,@TXCSR ;SET TXCSR BIT6.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE

```

```

1350
1351
1352 004660 000012
1353 004662 004702
1354 004664 000144
1355 004666 004670
1356
1357
1358
1359 004670 105777 174210
1360 004674 100401
1361 004676 104003
1362 004700 104012
1363
1364 004702 000013
1365 004704 005002
1366 004706 000144
1367 004710 004712
1368
1369
1370 004712 032777 000400 174164
1371 004720 001402
1372 004722 104003
1373 004724 000421
1374 004726 052777 000400 174150
1375 004734 032777 000400 174142
1376 004742 001002
1377 004744 104003
1378 004746 000410
1379 004750 042777 000400 174126
1380 004756 032777 000400 174120
1381 004764 001401
1382 004766 104003
1383 004770 052777 000400 174106
1384 004776 104011
1385 005000 104012
1386
1387
1388
1389 005002 000014
1390 005004 005026
1391 005006 000144
1392 005010 005012
1393
1394
1395 005012 032777 100000 174064
1396 005020 001401
1397 005022 104003
1398 005024 104012
1399

```

```

:*****
AT12: 12 ;TEST NUMBER 12 *
      AT13 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AKA ;SCOPE ENTRY POINT *
:*****
:TEST THAT TXCSR BIT 7 (READY BIT) IS SET UPON ENTERING ROUTINE AND
:THAT IT CAN BE READ RELIABLY.
AKA: TSTB @TXCSR ;SEE IF TXCSR BIT 7 IS SET.
      BMI AKB ;BRANCH IF SET.
      ERROR ;TXCSR BIT 7 NOT SET.
AKB: SCOPE ;SCOPE
:*****
AT13: 13 ;TEST NUMBER 13 *
      AT14 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ALA ;SCOPE ENTRY POINT *
:*****
:TEST THAT TXCSR BIT8 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
ALA: BIT #BIT8,@TXCSR ;SEE IF TXCSR BIT8 IS CLEAR.
      BEQ ALB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT8
      BR ALD
ALB: BIS #BIT8,@TXCSR ;SET TXCSR BIT8.
      BIT #BIT8,@TXCSR ;SEE IF BIT IS SET.
      BNE ALC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT8 FAILED TO SET.
      BR ALD
ALC: BIC #BIT8,@TXCSR ;CLEAR TXCSR BIT8
      BIT #BIT8,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ ALD
      ERROR ;TXCSR BIT8 FAILED TO CLEAR.
ALD: BIS #BIT8,@TXCSR ;SET TXCSR BIT8.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE
:*****
AT14: 14 ;TEST NUMBER 14 *
      AT15 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AMA ;SCOPE ENTRY POINT *
:*****
:TEST THAT TXCSR BIT15 IS CLEAR AND CAN BE READ RELIABLY.
AMA: BIT #BIT15,@TXCSR ;SEE IF TXCSR BIT15 IS CLEAR.
      BEQ AMB ;BRANCH IF BIT IS CLEAR.
      ERROR ;TXCSR BIT15 IS NOT CLEAR.
AMB: SCOPE ;SCOPE

```

```

1400      ;*****
1401 005026 000015 AT15: 15 ;TEST NUMBER 15 *
1402 005030 005120      AT16 ;ADDRESS OF NEXT TEST *
1403 005032 000144      100. ;TEST ITERATION COUNT *
1404 005034 005036      ANA ;SCOPE ENTRY POINT *
1405      ;*****
1406      ;TEST THAT RXCSR BIT 0 (DATA TERMINAL READY) CAN BE SET, NOT CLEARED BY RESET, AND CLEAR
1407 005036 052777 000001 174034 ANA: BIS #BIT0, @RXCSR ;SET RXCSR BIT 0.
1408 005044 032777 000001 174026      BIT #BIT0, @RXCSR ;SEE IF BIT IS SET.
1409 005052 001002      BNE ANB ;BRANCH IF BIT IS SET.
1410 005054 104003      ERROR
1411 005056 000417      BR AND
1412 005060 104011      ANB: SRESET ;ISSUE RESET.
1413 005062 032777 000001 174010      BIT #BIT0, @RXCSR ;SEE IF BIT IS STILL SET.
1414 005070 001002      BNE ANC ;BRANCH IF BIT SET.
1415 005072 104003      ERROR ;RESET CLEARED RXCSR BIT 0.
1416 005074 000410      BR AND
1417 005076 042777 000001 173774 ANC: BIC #BIT0, @RXCSR ;CLEAR RXCSR BIT 0.
1418 005104 032777 000001 173766      BIT #BIT0, @RXCSR ;SEE IF BIT IS CLEAR.
1419 005112 001401      BEQ AND ;BRANCH IF BIT IS CLEAR.
1420 005114 104003      ERROR ;RXCSR BIT 0 FAILED TO CLEAR.
1421 005116 104012      AND: SCOPE ;SCOPE
1422      ;*****
1423 005120 000016 AT16: 16 ;TEST NUMBER 16
1424 005122 005220      AT17 ;ADDRESS OF NEXT TEST
1425 005124 000144      100. ;TEST ITERATION COUNT
1426 005126 005130      ANW ;SCOPE ENTRY POINT
1427      ;*****
1428      ;TEST THAT RXCSR BIT 1 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT
1429
1430 005130 032777 000002 173742 ANW: BIT #BIT1, @RXCSR ;SEE IF BIT 1 IS CLEAR
1431 005136 001402      BEQ ANX ;BRANCH IF CLEAR
1432 005140 104003      ERROR ;RESET DID NOT CLEAR RXCSR BIT 1
1433 005142 000421      BR ANZ
1434 005144 052777 000002 173726 ANX: BIS #BIT1, @RXCSR ;SET RXCSR BIT 1
1435 005152 032777 000002 173720      BIT #BIT1, @RXCSR ;SEE IF BIT IS SET
1436 005160 001002      BNE ANY ;BRANCH IF SET
1437 005162 104003      ERROR ;RXCSR BIT 1 FAILED TO SET
1438 005164 000410      BR ANZ
1439 005166 042777 000002 173704 ANY: BIC #BIT1, @RXCSR ;CLEAR RXCSR BIT 1
1440 005174 032777 000002 173676      BIT #BIT1, @RXCSR ;SEE IF BIT IS CLEAR
1441 005202 001401      BEQ ANZ
1442 005204 104003      ERROR ;RXCSR BIT 1 FAILED TO CLEAR
1443 005206 052777 000002 173664 ANZ: BIS #BIT1, @RXCSR ;SET RXCSR BIT 1
1444 005214 104011      SRESET ;ISSUE RESET TO CLEAR BIT
1445 005216 104012      SCOPE ;SCOPE
1446

```



```

1447
1448
1449
1450 005220 000017
1451 005222 005244
1452 005224 000144
1453 005226 005230
1454
1455
1456 005230 032777 000004 173642
1457 005236 001401
1458 005240 104003
1459 005242 104012
1460
1461 005244 000020
1462 005246 005344
1463 005250 000144
1464 005252 005254
1465
1466
1467 005254 032777 000010 173616
1468 005262 001402
1469 005264 104003
1470 005266 000421
1471 005270 052777 000010 173602
1472 005276 032777 000010 173574
1473 005304 001002
1474 005306 104003
1475 005310 000410
1476 005312 042777 000010 173560
1477 005320 032777 000010 173552
1478 005326 001401
1479 005330 104003
1480 005332 052777 000010 173540
1481 005340 104011
1482 005342 104012
1483
1484 005344 000021
1485 005346 005444
1486 005350 000144
1487 005352 005354
1488
1489
1490 005354 032777 000020 173516
1491 005362 001402
1492 005364 104003
1493 005366 000421
1494 005370 052777 000020 173502
1495 005376 032777 000020 173474
1496 005404 001002
1497 005406 104003
1498 005410 000410
1499 005412 042777 000020 173460
1500 005420 032777 000020 173452
1501 005426 001401
1502 005430 104003

```

```

;*****
AT17: 17 ;TEST NUMBER 17 *
      AT20 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      APA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RXCSR BIT2 IS CLEAR AND CAN BE READ RELIABLY.
APA: BIT #BIT2,ARXCSR ;SEE IF RXCSR BIT2 IS CLEAR.
      BEQ APB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RXCSR BIT2 IS NOT CLEAR.
      SCOPE ;SCOPE
APB:
;*****
AT20: 20 ;TEST NUMBER 20 *
      AT21 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AQA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RXCSR BIT3 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AQA: BIT #BIT3,ARXCSR ;SEE IF RXCSR BIT3 IS CLEAR.
      BEQ AGB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR RXCSR BIT3
      BR AQA
AGB: BIS #BIT3,ARXCSR ;SET RXCSR BIT3.
      BIT #BIT3,ARXCSR ;SEE IF BIT IS SET.
      BNE AQC ;BRANCH IF BIT IS SET.
      ERROR ;RXCSR BIT3 FAILED TO SET.
      BR AQA
AQC: BIC #BIT3,ARXCSR ;CLEAR RXCSR BIT3
      BIT #BIT3,ARXCSR ;SEE IF BIT IS CLEAR.
      BEQ AQD
      ERROR ;RXCSR BIT3 FAILED TO CLEAR.
1480 AQD: BIS #BIT3,ARXCSR ;SET RXCSR BIT3.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE
;*****
AT21: 21 ;TEST NUMBER 21 *
      AT22 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ARA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RXCSR BIT4 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
ARA: BIT #BIT4,ARXCSR ;SEE IF RXCSR BIT4 IS CLEAR.
      BEQ ARB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR RXCSR BIT4
      BR ARD
ARB: BIS #BIT4,ARXCSR ;SET RXCSR BIT4.
      BIT #BIT4,ARXCSR ;SEE IF BIT IS SET.
      BNE ARC ;BRANCH IF BIT IS SET.
      ERROR ;RXCSR BIT4 FAILED TO SET.
      BR ARD
ARC: BIC #BIT4,ARXCSR ;CLEAR RXCSR BIT4
      BIT #BIT4,ARXCSR ;SEE IF BIT IS CLEAR.
      BEQ ARD
      ERROR ;RXCSR BIT4 FAILED TO CLEAR.

```

H03

MAIN. MACY11 27(732) 29-SEP-76 13:35 PAGE 34
DZDCAB.P11

```
1503 005432 052777 000020 173440 ARD:  BIS      #BIT4,ARXCSR  ;SET RXCSR BIT4.
1504 005440 104011          SRESET          ;ISSUE RESET TO CLEAR BIT.
1505 005442 104012          SCOPE              ;SCOPE
1506
1507          ;*****
1508 005444 000022 AT22:  22          ;TEST NUMBER 22
1509 005446 005470          AT23          ;ADDRESS OF NEXT TEST
1510 005450 000144          100.          ;TEST ITERATION COUNT
1511 005452 005454          ARBA          ;SCOPE ENTRY POINT
1512          ;*****
1513          ;TEST THAT PARITY INDICATOR (BITS RXCSR) IS CLEAR
1514          ;AND CAN BE READ RELIABLY.
1515
1516 005454 032777 000040 173416 ARBA:  BIT      #BITS,ARXCSR  ;SEE IF PARITY INDICATOR IS CLEAR
1517 005462 001401          BEQ      ARBB          ;BRANCH IF CLEAR
1518 005464 104003          ERROR          ;IS NOT CLEAR
1519 005466 104012          ARBB:  SCOPE          ;SCOPE
1520
1521          ;*****
1522 005470 000023 AT23:  23          ;TEST NUMBER 23
1523 005472 005576          AT24          ;ADDRESS OF NEXT TEST
1524 005474 000144          100.          ;TEST ITERATION COUNT
1525 005476 005500          ASA          ;SCOPE ENTRY POINT
1526          ;*****
1527          ;TEST THAT RXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1528 005500 012767 000340 172270 ASA:  MOV      #PRTY7,PSW  ;SET PRIORITY 7.
1529 005506 032777 000100 173364          BIT      #BIT6,ARXCSR  ;SEE IF RXCSR BIT6 IS CLEAR.
1530 005514 001402          BEQ      ASB          ;BRANCH IF BIT IS CLEAR.
1531 005516 104003          ERROR          ;RESET DID NOT CLEAR RXCSR BIT6
1532 005520 000421          BR      ASD          ;
1533 005522 052777 000100 173350 ASB:  BIS      #BIT6,ARXCSR  ;SET RXCSR BIT6.
1534 005530 032777 000100 173342          BIT      #BIT6,ARXCSR  ;SEE IF BIT IS SET.
1535 005536 001402          BNE      ASC          ;BRANCH IF BIT IS SET.
1536 005540 104003          ERROR          ;RXCSR BIT6 FAILED TO SET.
1537 005542 000410          BR      ASD          ;
1538 005544 042777 000100 173326 ASC:  BIC      #BIT6,ARXCSR  ;CLEAR RXCSR BIT6
1539 005552 032777 000100 173320          BIT      #BIT6,ARXCSR  ;SEE IF BIT IS CLEAR.
1540 005560 001401          BEQ      ASD          ;
1541 005562 104003          ERROR          ;RXCSR BIT6 FAILED TO CLEAR.
1542 005564 052777 000100 173306 ASD:  BIS      #BIT6,ARXCSR  ;SET RXCSR BIT6.
1543 005572 104011          SRESET          ;ISSUE RESET TO CLEAR BIT.
1544 005574 104012          SCOPE          ;SCOPE
1545          ;*****
1546 005576 000024 AT24:  24          ;TEST NUMBER IS 24
1547 005600 005622          AT25          ;ADDRESS OF NEXT TEST
1548 005602 000144          100.          ;TEST ITERATION COUNT
1549 005604 005606          ATA          ;SCOPE ENTRY POINT
1550          ;*****
1551          ;TEST THAT RXCSR BIT7 IS CLEAR AND CAN BE READ RELIABLY.
1552 005606 032777 000200 173264 ATA:  BIT      #BIT7,ARXCSR  ;SEE IF RXCSR BIT7 IS CLEAR.
1553 005614 001401          BEQ      ATB          ;BRANCH IF BIT IS CLEAR.
1554 005616 104003          ERROR          ;RXCSR BIT7 IS NOT CLEAR.
1555 005620 104012          ATB:  SCOPE          ;SCOPE
1556
1557          ;*****
1558 005622 000025 AT25:  25          ;TEST NUMBER 25
```

```

1559 005624 005722 AT26 ;ADDRESS OF NEXT TEST *
1560 005626 000144 100. ;TEST ITERATION COUNT *
1561 005630 005632 AUA ;SCOPE ENTRY POINT *
1562 ;*****
1563 ;TEST THAT RXCSR BIT8 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1564 005632 032777 000400 173240 AUA: BIT #BIT8, @RXCSR ;SEE IF RXCSR BIT8 IS CLEAR.
1565 005640 001402 BEQ AVB ;BRANCH IF BIT IS CLEAR.
1566 005642 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT8
1567 005644 000421 BR AVD
1568 005646 052777 000400 173224 AUB: BIS #BIT8, @RXCSR ;SET RXCSR BIT8.
1569 005654 032777 000400 173216 BIT #BIT8, @RXCSR ;SEE IF BIT IS SET.
1570 005662 001002 BNE AUC ;BRANCH IF BIT IS SET.
1571 005664 104003 ERROR ;RXCSR BIT8 FAILED TO SET.
1572 005666 000410 BR AVD
1573 005670 042777 000400 173202 AUC: BIC #BIT8, @RXCSR ;CLEAR RXCSR BIT8
1574 005676 032777 000400 173174 BIT #BIT8, @RXCSR ;SEE IF BIT IS CLEAR.
1575 005704 001401 BEQ AVD
1576 005706 104003 ERROR ;RXCSR BIT8 FAILED TO CLEAR.
1577 005710 052777 000400 173162 AUD: BIS #BIT8, @RXCSR ;SET RXCSR BIT8.
1578 005716 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1579 005720 104012 SCOPE ;SCOPE
1580 ;*****
1581 005722 000026 AT26: 26 ;TEST NUMBER 26 *
1582 005724 006022 AT27 ;ADDRESS OF NEXT TEST *
1583 005726 000144 100. ;TEST ITERATION COUNT *
1584 005730 005732 AVA ;SCOPE ENTRY POINT *
1585 ;*****
1586 ;TEST THAT RXCSR BIT9 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1587 005732 032777 001000 173140 AVA: BIT #BIT9, @RXCSR ;SEE IF RXCSR BIT9 IS CLEAR.
1588 005740 001402 BEQ AVB ;BRANCH IF BIT IS CLEAR.
1589 005742 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT9
1590 005744 000421 BR AVD
1591 005746 052777 001000 173124 AVB: BIS #BIT9, @RXCSR ;SET RXCSR BIT9.
1592 005754 032777 001000 173116 BIT #BIT9, @RXCSR ;SEE IF BIT IS SET.
1593 005762 001002 BNE AVC ;BRANCH IF BIT IS SET.
1594 005764 104003 ERROR ;RXCSR BIT9 FAILED TO SET.
1595 005766 000410 BR AVD
1596 005770 042777 001000 173102 AVC: BIC #BIT9, @RXCSR ;CLEAR RXCSR BIT9
1597 005776 032777 001000 173074 BIT #BIT9, @RXCSR ;SEE IF BIT IS CLEAR.
1598 006004 001401 BEQ AVD
1599 006006 104003 ERROR ;RXCSR BIT9 FAILED TO CLEAR.
1600 006010 052777 001000 173062 AVD: BIS #BIT9, @RXCSR ;SET RXCSR BIT9.
1601 006016 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1602 006020 104012 SCOPE ;SCOPE
1603 ;*****
1604 006022 000027 AT27: 27 ;TEST NUMBER 27 *
1605 006024 006122 AT30 ;ADDRESS OF NEXT TEST *
1606 006026 000144 100. ;TEST ITERATION COUNT *
1607 006030 006032 AWA ;SCOPE ENTRY POINT *
1608 ;*****
1609 ;TEST THAT RXCSR BIT10 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1610 006032 032777 002000 173040 AWA: BIT #BIT10, @RXCSR ;SEE IF RXCSR BIT10 IS CLEAR.
1611 006040 001402 BEQ AWB ;BRANCH IF BIT IS CLEAR.
1612 006042 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT10
1613 006044 000421 BR AWB
1614 006046 052777 002000 173024 AWB: BIS #BIT10, @RXCSR ;SET RXCSR BIT10.

```

```

1615 006054 032777 002000 173016 BIT #BIT10, @RXCSR ;SEE IF BIT IS SET.
1616 006062 001002 BNE AWC ;BRANCH IF BIT IS SET.
1617 006064 104003 ERROR ;RXCSR BIT10 FAILED TO SET.
1618 006066 000410 BR AWD
1619 006070 042777 002000 173002 AWC: BIC #BIT10, @RXCSR ;CLEAR RXCSR BIT10
1620 006076 032777 002000 172774 BIT #BIT10, @RXCSR ;SEE IF BIT IS CLEAR.
1621 006104 001401 BEQ AWD
1622 006106 104003 ERROR ;RXCSR BIT10 FAILED TO CLEAR.
1623 006110 052777 002000 172762 AWD: BIS #BIT10, @RXCSR ;SET RXCSR BIT10.
1624 006116 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1625 006120 104012 SCOPE ;SCOPE
1626 *****
1627 006122 000030 AT30: 30 ;TEST NUMBER 30 *
1628 006124 006146 AT31 ;ADDRESS OF NEXT TEST *
1629 006126 000144 100. ;TEST ITERATION COUNT *
1630 006130 006132 AXA ;SCOPE ENTRY POINT *
1631 *****
1632 ;TEST THAT RXCSR BIT12 IS CLEAR AND CAN BE READ RELIABLY.
1633 006132 032777 010000 172740 AXA: BIT #BIT12, @RXCSR ;SEE IF RXCSR BIT12 IS CLEAR.
1634 006140 001401 BEQ AXB ;BRANCH IF BIT IS CLEAR.
1635 006142 104003 ERROR ;RXCSR BIT12 IS NOT CLEAR.
1636 006144 104012 AXB: SCOPE ;SCOPE
1637 *****
1638 ;TEST THAT RXCSR BIT13 IS CLEAR AND CAN BE READ RELIABLY.
1639 006146 000031 AT31: 31 ;TEST NUMBER 31 *
1640 006150 006172 AT32 ;ADDRESS OF NEXT TEST *
1641 006152 000144 100. ;TEST ITERATION COUNT *
1642 006154 006156 AYA ;SCOPE ENTRY POINT *
1643 *****
1644 ;TEST THAT RXCSR BIT13 IS CLEAR AND CAN BE READ RELIABLY.
1645 006156 032777 020000 172714 AYA: BIT #BIT13, @RXCSR ;SEE IF RXCSR BIT13 IS CLEAR.
1646 006164 001401 BEQ AYB ;BRANCH IF BIT IS CLEAR.
1647 006166 104003 ERROR ;RXCSR BIT13 IS NOT CLEAR.
1648 006170 104012 AYB: SCOPE ;SCOPE
1649

```

```

1650
1651
1652 006172 000032
1653 006174 006216
1654 006176 000144
1655 006200 006202
1656
1657
1658 006202 032777 040000 172670
1659 006210 001401
1660 006212 104003
1661 006214 104012
1662
1663 006216 000033
1664 006220 006242
1665 006222 000144
1666 006224 006226
1667
1668
1669 006226 032777 100000 172644
1670 006234 001401
1671 006236 104003
1672 006240 104012

```

```

:*****
AT32: 32 ;TEST NUMBER 32 *
      AT33 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AZA ;SCOPE ENTRY POINT *
:*****
:TEST THAT RXCSR BIT14 IS CLEAR AND CAN BE READ RELIABLY.
AZA: BIT #BIT14,@RXCSR ;SEE IF RXCSR BIT14 IS CLEAR.
      BEQ AZB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RXCSR BIT14 IS NOT CLEAR.
AZB: SCOPE ;SCOPE
:*****
AT33: 33 ;TEST NUMBER 33 *
      AT34 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AAAA ;SCOPE ENTRY POINT *
:*****
:TEST THAT RXCSR BIT15 IS CLEAR AND CAN BE READ RELIABLY.
AAAA: BIT #BIT15,@RXCSR ;SEE IF RXCSR BIT15 IS CLEAR.
      BEQ AAAB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RXCSR BIT15 IS NOT CLEAR.
AAAB: SCOPE ;SCOPE

```

1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728

006242 000034
006244 006320
006246 000144
006250 006252

006252 052777 000001 172620
006260 032777 000004 172612

006266 001002
006270 104003
006272 000410
006274 042777 000001 172576
006302 032777 000004 172570

006310 001401
006312 104003
006314 104011
006316 104012

006320 000035
006322 006460
006324 000144
006326 006330

006330 042777 000001 172542
006336 017767 172536 172714
006344 032777 040000 172526

006352 001402
006354 104003
006356 000436
006360 005277 172514
006364 000004

006366 017767 172506 172664
006374 032767 040000 172656

006402 001002
006404 104003
006406 000422
006410 032777 040000 172462

006416 001402
006420 104003
006422 000414

```

;ALL PREVIOUS TESTS MUST HAVE BEEN RUN SUCCESSFULLY PRIOR
;TO RUNNING THE FOLLOWING TESTS. ALSO, THE JUMPER CONNECTOR
;MUST BE INSERTED IN THE DC11 CABLE. TO THE MODEM. COMMENTS
;REFER TO OPERATION WITH JUMPER INSERTED.
;*****
AT34: 34 ;TEST NUMBER 34
      AT35 ;ADDRESS OF NEXT TEST
      100. ;TEST ITERATION COUNT
      AFBA ;SCOPE ENTRY POINT
;*****
;TEST THAT CARRIER DETECT SETS AND CLEARS WHEN DATA TERMINAL
;READY SETS AND CLEARS.
AFBA: BIS #BIT0,QRXCSR ;SET DATA TERMINAL READY
      BIT #BIT2,QRXCSR ;TEST CARRIER DETECT
      BNE AFBB ;SHOULD BE SET
      ERROR ;WASN'T
      BR AFBC
AFBB: BIC #BIT0,QRXCSR ;CLEAR DATA TERMINAL READY
      BIT #BIT2,QRXCSR ;TEST CARRIER DETECT
      BEQ AFBC
      ERROR ;WAS SET, ERROR
AFBC: SRESET
      SCOPE
;*****
AT35: 35 ;TEST NUMBER 35
      AT36 ;ADDRESS OF NEXT TEST
      100. ;TEST ITERATION COUNT
      AGBA ;SCOPE ENTRY POINT
;*****
;TEST THAT CARRIER TRANSITION (BIT 14) SETS WHEN CARRIER DETECT
;CHANGES STATE, AND IS CLEARED WHEN RXCSR IS READ.
AGBA: BIC #BIT0,QRXCSR ;CLEAR DATA TERMINAL READY
      MOV QRXCSR,RXCSTR ;READ RXCSR
      BIT #BIT14,QRXCSR ;TEST CARRIER TRANSITION
      BEQ AGBB ;WAS CLEAR GO TO AGBB
      ERROR ;WASN'T CLEAR
      BR AGBE ;GO TO SCOPE
AGBB: INC QRXCSR ;SETTING DATA TERMINAL READY
      IOT ;CAUSES CARRIER DETECT TO SET
      ;WHICH CAUSES CARRIER TRANSITION
      ;TO SET.
      MOV QRXCSR,RXCSTR ;MOVE RXCSR TO TEMPORARY LOCATION
      BIT #BIT14,RXCSTR ;TEST CARRIER TRANSITION
      BNE AGBC ;SHOULD BE SET GO TO AGBC
      ERROR ;WAS CLEAR
      BR AGBE ;GO TO SCOPE
AGBC: BIT #BIT14,QRXCSR ;CARRIER TRANSITION BIT SHOULD
      ;HAVE BEEN CLEARED
      BEQ AGBD ;IT WAS GO TO AGBD
      ERROR ;IT WASN'T
      BR AGBE ;GO TO SCOPE

```

M03

.MAIN. MACY11 27(732) 28-SEP-76 13:35 PAGE 39
 DZDCAB.P11

```

1729 006424 042777 000001 172446 AGBD: BIC #BIT0, @RXCSR ; CLEARING DATA TERMINAL READY
1730 ; CAUSES CARRIER DETECT TO CLEAR
1731 ; BUT CARRIER TRANSITION
1732 ; WILL NOT SET
1733 006432 017767 172442 172620 MOV @RXCSR, RXCSRT ; MOV RXCSR TO TEMPORARY LOCATION
1734 006440 032767 040000 172612 BIT #BIT14, RXCSRT ; TEST CARRIER TRANSITION
1735 006446 001402 BEQ AGBE ; SHOULD BE CLEAR
1736 006450 104003 ERROR ; IT WASN'T
1737 006452 000400 BR AGBE
1738 006454 104011 AGBE: SRESET ; ISSUE RESET
1739 006456 104012 SCOPE ; SCOPE
1740 ; *****
1741
1742 006460 000036 AT36: 36 ; TEST NUMBER 36
1743 006462 006574 AT37 ; ADDRESS OF NEXT TEST
1744 006464 000144 100. ; TEST ITERATION COUNT
1745 006466 006470 AMBA ; SCOPE ENTRY POINT
1746 ; *****
1747 ; TEST THAT CARRIER TRANSITION SETTING CAUSES ERROR (BIT 15 RXCSR) TO
1748 ; SET AND THAT READING RXCSR CLEARS ERROR.
1749
1750 006470 042777 000001 172402 AMBA: BIC #BIT0, @RXCSR ; CLEAR DATA TERMINAL READY
1751 006476 052777 000001 172374 BIS #BIT0, @RXCSR ; SET DATA TERMINAL READY
1752 006504 017767 172370 172546 MOV @RXCSR, +RXCSRT ; MOVE RXCSR TO TEMPORARY LOCATION
1753 006512 032767 100000 172540 BIT #BIT15, RXCSRT ; TEST ERROR BIT
1754 006520 001002 BNE AMBB ; ERROR BIT SHOULD BE SET
1755 006522 104003 ERROR
1756 006524 000421 BR AMBD
1757 006526 032777 100000 172344 AMBB: BIT #BIT15, @RXCSR ; TEST ERROR BIT
1758 006534 001402 BEQ AMBC ; SHOULD BE CLEAR
1759 006536 104003 ERROR
1760 006540 000413 BR AMBD
1761 006542 042777 000001 172330 AMBC: BIC #BIT0, @RXCSR ; CLEAR DATA TERMINAL READY
1762 006550 017767 172324 172502 MOV @RXCSR, RXCSRT ; MOV RXCSR TO TEMPORARY LOCATION
1763 006556 032767 100000 172474 BIT #BIT15, RXCSRT ; TEST ERROR BIT
1764 006564 001401 BEQ AMBD ; SHOULD BE CLEAR
1765 006566 104003 ERROR
1766 006570 104011 AMBD: SRESET ; ISSUE RESET
1767 006572 104012 SCOPE ; SCOPE

```

```

1768
1769 006574 000037
1770 006576 006666
1771 006600 000144
1772 006602 006604
1773
1774
1775
1776
1777 006604 042777 000001 172266 AJBA: BIC #BIT0, @RXCSR ;CLEAR DATA TERMINAL READY
1778 006612 005277 172266 INC @TXCSR ;TEST CLEAR TO SEND
1779 006616 000004 IOT
1780
1781
1782 006620 052777 000001 172252 AJBB: BIS #BIT0, @RXCSR ;SET DATA TERMINAL READY
1783 006626 032777 000002 172250 BIT #BIT1, @TXCSR ;TEST CLEAR TO SEND
1784 006634 001002 BNE AJBC ;BRANCH IF SET
1785 006636 104003 ERROR ;CLEAR TO SEND SHOULD BE SET
1786 006640 000410 BR AJBD
1787 006642 042777 000001 172230 AJBC: BIC #BIT0, @RXCSR ;CLEAR DATA TERMINAL READY
1788 006650 032777 000002 172226 BIT #BIT1, @TXCSR ;TEST CLEAR TO SEND
1789 006656 001401 BEQ AJBD
1790 006660 104003 ERROR ;CLEAR TO SEND SHOULD BE CLEAR
1791 006662 104011 AJBD: SRESET ;ISSUE RESET
1792 006664 104012 SCOPE ;SCOPE
1793
1794 006666 000040
1795 006670 007004
1796 006672 000144
1797 006674 006676
1798
1799
1800
1801
1802
1803 006676 042777 000001 172200 AKBA: BIC #BIT0, @TXCSR ;CLEAR REQUEST TO SEND
1804 006704 052777 000001 172172 AKBB: BIS #BIT0, @TXCSR ;SET REQUEST TO SEND
1805 006712 042777 000001 172164 BIC #BIT0, @TXCSR
1806 006720 032777 020000 172152 BIT #BIT13, @RXCSR ;TEST RING
1807 006726 001002 BNE AKBC
1808 006730 104003 ERROR ;RING SHOULD BE SET
1809 006732 000422 BR AKBE
1810 006734 032777 020000 172136 AKBC: BIT #BIT13, @RXCSR ;TEST RING
1811 006742 001402 BEQ AKBD
1812 006744 104003 ERROR ;RING SHOULD BE CLEAR
1813 006746 000414 BR AKBE
1814 006750 052777 000001 172126 AKBD: BIS #BIT0, @TXCSR ;SET
1815 006756 042777 000001 172120 BIC #BIT0, @TXCSR ;RING
1816 006764 000005 RESET
1817 006766 032777 020000 172104 BIT #BIT13, @RXCSR ;TEST RING
1818 006774 001401 BEQ AKBE ;BRANCH IF CLEAR
1819 006776 104003 ERROR ;RING SHOULD BE CLEAR AFTER RESET
1820 ;BUT WAS SET
1821 007000 104011 AKBE: SRESET ;ISSUE RESET
1822 007002 104012 SCOPE ;SCOPE
1823

```



```

1824 007004 000041          AT41: 41          : TEST NUMBER 41
1825 007006 007112          AT42          : ADDRESS OF NEXT TEST
1826 007010 000144          100.          : TEST ITERATION COUNT
1827 007012 007014          AOBA          : SCOPE ENTRY POINT
1828                                     :*****
1829                                     :TEST THAT ERROR (BIT 15 RXCSR) SETS WHEN RING SETS.
1830
1831 007014 042777 000001 172062 AOBA: BIC      #BIT0,RTXCSR : SET REQUEST TO SEND
1832 007022 032777 100000 172050      BIT      #BIT15,RTXCSR : TEST ERROR BIT
1833 007030 001402          BEQ      AOBB
1834 007032 104003          ERROR
1835 007034 000424          BR
1836 007036 052777 000001 172040 AOBB: SIS      #BIT0,RTXCSR : SET REQUEST TO SEND
1837 007044 042777 000001 172032      BIC      #BIT0,RTXCSR : CLEAR REQUEST TO SEND
1838 007052 032777 100000 172020      BIT      #BIT15,RTXCSR : TEST ERROR BIT
1839 007060 001002          BNE      AOBC
1840 007062 104003          ERROR
1841 007064 000410          BR
1842 007066 042777 000001 172010 AOBC: BIC      #BIT0,RTXCSR : CLEAR REQUEST TO SEND
1843 007074 032777 100000 171776      BIT      #BIT15,RTXCSR : TEST ERROR BIT
1844 007102 001401          BEQ      AOBD
1845 007104 104003          ERROR
1846 007106 104011          AOBD: SRESET      : ISSUE RESET
1847 007110 104012          SCOPE

```

```

1848
1849
1850 007112 000042
1851 007114 007212
1852 007116 000144
1853 007120 007122
1854
1855
1856
1857
1858 007122 042777 000400 171750 ALBA: BIC #BIT8,@TXCSR ;CLEAR SUPERVISOR XMIT DATA
1859 007130 032777 100000 171746 BIT #BIT15,@TXCSR ;TEST SUPERVISORY RECEIVE DATA.
1860 007136 001402 BEQ ALBB
1861 007140 104003 ERROR ;SHOULD HAVE BEEN CLEAR
1862 007142 000421 BR ALBD
1863 007144 052777 000400 171726 ALBB: BIS #BIT8,@TXCSR ;SET SUPERVISORY XMIT DATA
1864 007152 032777 100000 171724 BIT #BIT15,@TXCSR ;TEST SUPERVISORY RECEIVE DATA
1865 007160 001002 BNE ALBC
1866 007162 104003 ERROR ;SHOULD HAVE BEEN SET
1867 007164 000410 BR ALBD
1868 007166 042777 000400 171704 ALBC: BIC #BIT8,@TXCSR ;CLEAR SUPERVISORY XMIT DATA
1869 007174 032777 100000 171702 JI1 #BIT15,@TXCSR ;TEST SUPERVISORY RECEIVE DATA
1870 007202 001401 BEQ ALBD
1871 007204 104003 ERROR ;SHOULD HAVE BEEN CLEAR
1872 007206 104011 ALBD: SRESET ;ISSUE RESET
1873 007210 104012 SCOPE ;SCOPE
1874
1875 007212 000043
1876 007214 007316
1877 007216 000144
1878 007220 007222
1879
1880
1881 007222 012767 000340 170546 ABAA: MOV #PRTY7,PSW ;SET PRIORITY 7.
1882 007230 012777 177777 171646 MOV #-1,@TXCSR ;SET ALL POSSIBLE BITS IN TXCSR
1883 007236 104011 SRESET ;ISSUE RESET TO CLEAR BITS
1884 007240 022777 000200 171636 CMP #BIT7,@TXCSR ;SEE IF ONLY BIT 7 IS SET.
1885 007246 001422 BEQ ABAB ;BRANCH IF ONLY BIT 7 IS SET
1886 007250 017767 171630 172000 MOV @TXCSR,TXCSRT ;SAVE CONTENTS OF TXCSR
1887 007256 012767 000200 172000 MOV #BIT7,TEMP ;MOVE EXPECTED TXCSR TO TEMP.
1888 007264 004567 173742 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1889 007270 001264 TEMP ;SOURCE ADDR.
1890 007272 015401 ATXSB ;DESTINATION ADDR.
1891 007274 000006 6 ;#OF DIGITS TO CONVERT.
1892 007276 004567 173730 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1893 007302 001256 TXCSRT ;SOURCE ADDR.
1894 007304 015416 ATXWAS ;DESTINATION ADDR.
1895 007306 001206 6 ;#OF DIGITS TO CONVERT.
1896 007310 104015 ERROR1 ;RESET FAILED TO CLEAR ALL BITS EXCEPT
1897 007312 015366 ATXCSR ;BIT 7 - SEE PRINTOUT
1898 007314 104012 ABAB: SCOPE

```

```

1899
1900 007316 000044
1901 007320 007472
1902 007322 000144
1903 007324 007326
1904
1905
1906
1907 007326 012767 000340 170442
1908 007334 042777 000001 171536
1909 007342 012777 177775 171530
1910 007350 052777 000030 171526
1911 007356 005077 171524
1912 007362 105777 171516
1913 007366 100375
1914 007370 012777 000001 171510
1915 007376 105777 171502
1916 007402 100375
1917 007404 104011
1918 007406 017767 171466 171644
1919 007414 022767 000005 171636
1920 007422 001417
1921 007424 012767 000005 171632
1922 007432 004567 173574
1923 007436 001264
1924 007440 015440
1925 007442 000006
1926 007444 004567 173562
1927 007450 001260
1928 007452 015455
1929 007454 000006
1930 007456 104015
1931 007460 015425
1932 007462 042777 000001 171410
1933 007470 104012
1934
1935 007472 000045
1936 007474 007522
1937 007476 000144
1938 007500 007502
1939
1940
1941 007502 005077 171400
1942 007506 105777 171372
1943 007512 100001
1944 007514 104003
1945 007516 104011
1946 007520 104012

:*****
AT44: 44 ;TEST NUMBER 44 *
      AT45 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ACAA ;SCOPE ENTRY POINT *
:*****
:TEST THAT RESET CLEARS ALL RXCSR BITS EXCEPT BIT 0 (DATA TERMINAL READY)
:RING, AND THE BREAK BIT.
ACAA: MOV #PRTY7,PSW ;SET PRIORITY 7
      BIC #BIT0,RXCSR ;CLEAR DATA TERM.READY
      MOV #177775,RXCSR ;SET ALL POSSIBLE BITS IN RXCSR
      BIS #30,ATXCSR ;SET MAINT BIT
      CLR ATXBUF ;TRANSMIT A CHAR
      TSTB ATXCSR ;WAIT FOR
      BPL -4 ;TRANSMITTER TO FINISH
      MOV #1,ATXBUF ;TRANSMIT ANOTHER CHAR.
      TSTB ATXCSR ;WAIT FOR
      BPL -4 ;TRANSMITTER TO FINISH
      SRESET ;ISSUE RESET TO CLEAR BITS.
      MOV RXCSR,RXCST ;MOVE RXCSR CONTENTS TO RXCST
      CMP #5,RXCST ;SEE IF ONLY BIT 0 IS SET
      BEQ ACAB ;BRANCH IF ONLY BIT 0 IS SET.
      MOV #5,TEMP
      JSR %5,0ACNV ;GO TO OCTAL TO ASCII CONVERT.
      TEMP ;SOURCE ADDR.
      ARXSB ;DESTINATION ADDR.
      6 ;#OF DIGITS TO CONVERT.
      JSR %5,0ACNV ;GO TO OCTAL TO ASCII CONVERT.
      RXCST ;SOURCE ADDR.
      ARXWAS ;DESTINATION ADDR.
      6 ;#OF DIGITS TO CONVERT.
      ERROR1 ;RESET FAILED TO CLEAR ALL BITS EXCEPT
      ARXCSR ;BIT 0. SEE ERROR PRINTOUT.
      BIC #BIT0,RXCSR ;CLEAR DATA TERM. READY
      SCOPE ;SCOPE
:*****
AT45: 45 ;TEST NUMBER 45 *
      AT46 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ADAA ;SCOPE ENTRY POINT *
:*****
:TEST THAT LOADING TXBUF (TRANSMITTER BUFFER) CLEARS TXCSR BIT 7 (READY)
ADAA: CLR ATXBUF ;LOAD TXBUF
      TSTB ATXCSR ;TEST TXCSR BIT 7 (READY BIT)
      BPL ADAB ;BRANCH IF BIT NOT SET.
      ERROR ;ERROR. LOADING TXBUF FAILED TO CLEAR READY.
      SRESET ;ISSUE RESET TO SET READY.
      SCOPE ;SCOPE

```

```

1947
1948 007522 000046
1949 007524 007554
1950 007526 000012
1951 007530 007532
1952
1953
1954
1955 007532 005077 171350
1956 007536 104016
1957 007540 001750
1958 007542 105777 171336
1959 007546 100401
1960 007550 104003
1961
1962 007552 104012
1963
1964
1965 007554 000047
1966 007556 007614
1967 007560 000012
1968 007562 007564
1969
1970
1971
1972 007564 052777 000010 171312
1973 007572 005077 171310
1974 007576 104016
1975 007600 000764
1976 007602 105777 171276
1977 007606 100401
1978 007610 104003
1979
1980 007612 104012
1981
1982 007614 000050
1983 007616 007654
1984 007620 000012
1985 007622 007624
1986
1987
1988
1989 007624 052777 000020 171252
1990 007632 005077 171250
1991 007636 104016
1992 007640 000620
1993 007642 105777 171236
1994 007646 100401
1995 007650 104003
1996
1997 007652 104012

```

```

:*****
AT46: 46 ;TEST NUMBER 46 *
      AT47 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AEAA ;SCOPE ENTRY POINT *
:*****
;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 1000 MSECS AFTER
;LOADING TXBUF WITH TRANSMIT SPEED SET TO 00 (TXCSR BITS 3 AND 4)
AEAA: CLR @TXBUF ;LOAD TXBUF
      DELAY ;DELAY 1000 MSECS APPROX.
      1000.
      TSTB @TXCSR ;SEE IF READY BIT IS SET
      BMI AEAB ;BRANCH IF READY IS SET
      ERROR ;READY NOT SET 200 MSECS AFTER BUFFER
AEAB: ;LOAD TX SPEED = 00.
      SCOPE ;SCOPE
:*****
AT47: 47 ;TEST NUMBER 47 *
      AT50 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AFAA ;SCOPE ENTRY POINT *
:*****
;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 500 MSECS AFTER
;LOADING TXBUF, WITH TRANSMIT SPEED SET TO 01 (TXCSR BITS 3 AND 4).
AFAA: BIS #10,@TXCSR ;SET TX SPEED TO 01.
      CLR @TXBUF ;LOAD TXBUF
      DELAY ;DELAY 500 MSECS
      500.
      TSTB @TXCSR ;SEE IF READY BIT IS SET
      BMI AFAB ;BRANCH IF READY IS SET
      ERROR ;READY NOT SET 200 MSECS AFTER BUFFER
AFAB: ;LOAD TX SPEED = 01.
      SCOPE ;SCOPE
:*****
AT50: 50 ;TEST NUMBER 50 *
      AT51 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AGAA ;SCOPE ENTRY POINT *
:*****
;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 400 MSECS AFTER
;LOADING TXBUF, WITH TRANSMIT SPEED SET TO 10 (TXCSR BITS 3 AND 4).
AGAA: BIS #20,@TXCSR ;SET TX SPEED TO 10.
      CLR @TXBUF ;LOAD TXBUF
      DELAY ;DELAY 400 MSECS
      400.
      TSTB @TXCSR ;SEE IF READY BIT IS SET
      BMI AGAB ;BRANCH IF READY BIT IS SET
      ERROR ;READY NOT SET 200 MSECS AFTER BUFFER
AGAB: ;LOAD TX SPEED = 10.
      SCOPE ;SCOPE

```

```

1998
1999 007654 000051
2000 007656 007714
2001 007660 000012
2002 007662 007664
2003
2004
2005
2006 007664 052777 000030 171212
2007 007672 005077 171210
2008 007676 104016
2009 007700 000372
2010 007702 105777 171176
2011 007706 100401
2012 007710 104003
2013
2014 007712 104012
2015
2016
2017 007714 000052
2018 007716 010132
2019 007720 000144
2020 007722 007724
2021
2022
2023
2024 007724 005067 171316
2025 007730 005067 171314
2026 007734 005067 171312
2027 007740 005067 171310
2028 007744 042777 000030 171132
2029 007752 004767 000110
2030 007756 066767 000146 171262
2031 007764 052777 000010 171112
2032 007772 004767 000070
2033 007776 066767 000126 171244
2034 010004 042777 000030 171072
2035 010012 052777 000020 171064
2036 010020 004767 000042
2037 010024 066767 000100 171220
2038 010032 052777 000030 171044
2039 010040 004767 000022
2040 010044 066767 000060 171202
2041 010052 004767 004176
2042 010056 000402
2043 010060 104015
2044 010062 015464
2045 010064 104012
2046 010066 005067 000036
2047 010072 105777 171006
2048 010076 100375
2049 010100 104016
2050 010102 000024
2051 010104 005077 170776
2052 010110 104016
2053 010112 000001

*****
AT51: 51 ;TEST NUMBER 51 *
      AT52 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AHAA ;SCOPE ENTRY POINT *
*****
;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 250 MSECS AFTER
;LOADING TXBUF, WITH TRANSMIT SPEED SET TO 11 (TXCSR BITS 3 AND 4).
AHAA: BIS #30,ATXCSR ;SET TX SPEED TO 30.
      CLR ATXBUF ;LOAD TXBUF
      DELAY 250. ;DELAY 250 MSECS.
      TSTB ATXCSR ;SEE IF READY BIT IS SET.
      BMI AHAB ;BRANCH IF READY BIT IS SET.
      ERROR ;READY NOT SET 200 MSECS AFTER
           ;BUFFER LOAD. TX SPEED = 11.
AHAB: SCOPE ;SCOPE

*****
AT52: 52 ;TEST NUMBER 52 *
      AT53 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AIAA ;SCOPE ENTRY POINT *
*****
;TEST THAT TRANSMIT SPEEDS ARE ARRANGED IN ASCENDING ORDER BY CHECKING THAT TIME
;TO READY BIT (TXCSR BIT 7) DECREASES AS A HIGHER SPEED IS SELECTED.
AIAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD
      CLR CTRB ;(USED TO COUNT ELAPSED TIME.)
      CLR CTRC
      CLR CTRD
      BIC #30,ATXCSR ;SELECT TX SPEED 0
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRA ;ADD ELAPSED TIME TO CTRA.
      BIS #10,ATXCSR ;SELECT TX SPEED 1
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRB ;ADD ELAPSED TIME TO CTRB.
      BIC #30,ATXCSR ;SELECT TX SPEED 2
      BIS #20,ATXCSR
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRC ;ADD ELAPSED TIME TO CTRC.
      BIS #30,ATXCSR ;SELECT TX SPEED 3
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRD ;ADD ELAPSED TIME TO CTRD.
      JSR %7,CMPT ;CHECK THAT CTRA THROUGH CTRD CONTAIN
           ;DESCENDING VALUES
           ;TRANSMIT SPEEDS NOT ARRANGED IN
           ;ASCENDING ORDER.
AIAF: SCOPE ;SCOPE
AIAAS: CLR AIAST ;CLEAR ELAPSED TIME COUNTER.
      TSTB ATXCSR ;WAIT FOR TX READY.
      BPL .-4
           ;WAIT 20 MSECS.
      DELAY 20.
AIASA: CLR ATXBUF ;LOAD TXBUF.
      DELAY 1 ;DELAY 1 MSEC.
      I

```

2054	010114	005267	000010		INC	AJAST		; INCREMENT ELAPSED TIME COUNTER.
2055	010120	105777	170760		TSTB	@TXCSR		; READY SET?
2056	010124	100371			BPL	AJASA		; BRANCH IF READY NOT SET.
2057	010126	000207			RTS	%7		; EXIT.
2058	010130	000000			AJAST:	OPEN		
2059					:*****			
2060	010132	000053			AT53:	53		; TEST NUMBER 53 *
2061	010134	010316				AT54		; ADDRESS OF NEXT TEST *
2062	010136	000144				100.		; TEST ITERATION COUNT *
2063	010140	010142				AJAA		; SCOPE ENTRY POINT *
2064					:*****			
2065					; TEST FOR CORRECT OPERATION OF STOP CODE BIT (TXCSR BIT 8) BY CHECKING THAT TIME.			
2066					; REQUIRED TO COMPLETE TRANSMISSION OF 2 CONSECUTIVE CHARACTERS WITH STOP BIT			
2067					; SET TO 0 IS LONGER THAN TIME REQUIRED WITH STOP CODE BIT SET TO A 1.			
2068	010142	005067	171100		AJAA:	CLR	CTRA	; CLEAR CTRA AND CTRB
2069	010146	005067	171076			CLR	CTRB	; (ELAPSED TIME COUNTERS).
2070	010152	042777	000400	170724		BIC	#BIT8,@TXCSR	; SET STOP CODE TO 0 (2 STOP CODES)
2071	010160	004767	000044			JSR	%7,AJAS	; OUTPUT CHAR AND TIME
2072	010164	066767	000124	171054		ADD	AJAST,CTRA	; ADD ELAPSED TIME TO CTRA
2073	010172	052777	000400	170704		BIS	#BIT8,@TXCSR	; SET STOP CODE TO 1 (1 STOP CODE)
2074	010200	004767	000024			JSR	%7,AJAS	; OUTPUT CHARACTER AND TIME.
2075	010204	066767	000104	171036		ADD	AJAST,CTRB	; ADD ELAPSED TIME TO CTRB
2076	010212	026767	171030	171030		CMP	CTRA,CTRB	; SEE IF CTRA IS GREATER THAN CTRB
2077	010220	101002				BHI	AJAB	; BRANCH IF CTRA IS GREATER.
2078	010222	104015				ERROR1		; ERROR. ELAPSED TIME FOR 2 STOP CODE
2079	010224	015526				ESTPCD		; OPERATION NOT GREATER THAN FOR 1 STOP
2080								; CODE.
2081	010226	104012			AJAB:	SCOPE		; SCOPE
2082	010230	005067	000060		AJAS:	CLR	AJAST	; CLEAR ELAPSED TIME COUNTER AJAST
2083	010234	105777	170644			TSTB	@TXCSR	; WAIT FOR TX READY.
2084	010240	100375				BPL	.-4	
2085	010242	104016				DELAY		; WAIT 20 MSECS.
2086	010244	000024				20.		
2087	010246	005077	170634			CLR	@TXBUF	; LOAD TXBUF
2088	010252	104016			AJASA:	DELAY		; DELAY 1 MSEC
2089	010254	000001				1		
2090	010256	005267	000032			INC	AJAST	; INCREMENT ELAPSED TIME COUNTER
2091	010262	105777	170616			TSTB	@TXCSR	; READY SET?
2092	010266	100371				BPL	AJASA	; BRANCH IF READY NOT SET.
2093	010270	005077	170612			CLR	@TXBUF	; LOAD TXBUF.
2094	010274	104016			AJASB:	DELAY		; DELAY 1 MSEC.
2095	010276	000001				1		
2096	010300	005267	000010			INC	AJAST	; INCR ELAPSED TIME COUNTER.
2097	010304	105777	170574			TSTB	@TXCSR	; READY SET?
2098	010310	100371				BPL	AJASB	; BRANCH IF READY NOT SET.
2099	010312	000207				RTS	%7	; EXIT
2100	010314	000000			AJAST:	OPEN		; ELAPSED TIME COUNTER.

```

2101
2102
2103 010316 000054
2104 010320 010542
2105 010322 000144
2106 010324 010326
2107
2108
2109
2110
2111 010326 005067 170714
2112 010332 005067 170712
2113 010336 005067 170710
2114 010342 005067 170706
2115 010346 042777 003000 170524
2116 010354 004767 000116
2117 010360 066767 000154 170660
2118 010366 042777 003000 170504
2119 010374 052777 001000 170476
2120 010402 004767 000070
2121 010406 066767 000126 170634
2122 010414 042777 003000 170456
2123 010422 052777 002000 170450
2124 010430 004767 000042
2125 010434 066767 000100 170610
2126 010442 052777 003000 170430
2127 010450 004767 000022
2128 010454 066767 000060 170572
2129 010462 004767 003566
2130 010466 000402
2131 010470 104015
2132 010472 015577
2133 010474 104012
2134 010476 005067 000036
2135 010502 105777 170376
2136 010506 100375
2137 010510 104016
2138 010512 000024
2139 010514 005077 170366
2140 010520 104016
2141 010522 000001
2142 010524 005267 000010
2143 010530 105777 170350
2144 010534 100371
2145 010536 000207
2146 010540 000000

```

```

*****
AT54: 54 ;TEST NUMBER 54 *
      AT55 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AKAA ;SCOPE ENTRY POINT *
*****
;TEST FOR CORRECT OPERATION OF CHARACTER LENGTH SELECTION (RXCSR BITS 9 AND 10)
;BY CHECKING THAT TIME REQUIRED FOR OUTPUTTING A CHARACTER IS LONGEST FOR
;8 BIT CODE THAN FOR 7 BIT CODE ETC.
AKAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD.
      CLR CTRB ;(ELAPSED TIME COUNTERS).
      CLR CTRC
      CLR CTRD
      BIC #3000, @RXCSR ;SET CHAR LENGTH TO 00 (8 BIT CODE).
      JSR %7, AKAS ;OUTPUT CHARACTER AND TIME.
      ADD AKAST, CTRA ;ADD ELAPSED TIME TO CTRA
      BIC #3000, @RXCSR ;SET CHAR LENGTH TO 01 (7 BIT CODE).
      BIS #1000, @RXCSR
      JSR %7, AKAS ;OUTPUT CHARACTER AND TIME.
      ADD AKAST, CTRB ;ADD ELAPSED TIME TO CTRB.
      BIC #3000, @RXCSR ;SET CHAR LENGTH TO 10 (6 BIT CODE)
      BIS #2000, @RXCSR
      JSR %7, AKAS ;OUTPUT CHARACTER AND TIME.
      ADD AKAST, CTRC ;ADD ELAPSED TIME TO CTRC
      BIS #3000, @RXCSR ;SET CHAR LENGTH TO 11 (5 BIT CODE)
      JSR %7, AKAS ;OUTPUT CHARACTER AND TIME
      ADD AKAST, CTRD ;ADD ELAPSED TIME TO CTRD
      JSR %7, CMPT ;CHECK THAT CTRA THROUGH CTRD
      BR AKAB ;DESCENDING VALUES.
      ERROR! ;TX CHARACTER LENGTH NOT ARRANGED
      ETCLGT ;IN DESCENDING ORDER.
AKAB: SCOPE
AKAS: CLR AKAST ;CLEAR ELAPSED TIME COUNTER AKAST
      TSTB @TXCSR ;WAIT FOR TX READY.
      BPL .-4
      DELAY 20. ;WAIT 20 MSECS.
AKASA: CLR @TXBUF ;LOAD TXBUF
      DELAY 1 MSEC ;DELAY 1 MSEC
      INC AKAST ;INCREMENT ELAPSED TIME COUNTER
      TSTB @TXCSR ;READY SET?
      BPL AKASA ;BRANCH IF READY NOT SET
      RTS %7 ;EXIT
AKAST: OPEN ;ELAPSED TIME COUNTER

```

```

2147
2148
2149 010542 000055
2150 010544 010616
2151 010546 000144
2152 010550 010552
2153
2154
2155
2156
2157 010552 052777 000004 170324
2158 010560 005077 170322
2159 010564 104016
2160 010566 000310
2161 010570 105777 170304
2162 010574 100402
2163 010576 104003
2164 010600 000405
2165 010602 104011
2166 010604 105777 170270
2167 010610 100001
2168 010612 104003
2169 010614 104012
2170
2171 010616 000056
2172 010620 010664
2173 010622 000144
2174 010624 010626
2175
2176
2177
2178 010626 052777 000004 170250
2179 010634 005077 170246
2180 010640 105777 170234
2181 010644 100375
2182 010646 005777 170230
2183 010652 105777 170222
2184 010656 100001
2185 010660 104003
2186 010662 104012

;*****
AT55: 55 ;TEST NUMBER 55 *
      AT56 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ALAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT OUTPUTTING A CHARACTER WITH THE MAINTENANCE BIT SET (TXCSR BIT 2)
;RESULTS IN DONE BIT SETTING (RXCSR BIT 7) NO LATER THAN 200 MSECS, AND
;THAT RESET INSTRUCTION CLEARS THE DONE BIT
ALAA: BIS #BIT2,@TXCSR ;SET MAINTENANCE (TXCSR BIT 2)
      CLR @TXBUF ;LOAD TXBUF
      DELAY ;WAIT 200 MSECS.
      200.
      TSTB @RXCSR ;SEE IF DONE BIT IS SET
      BMI ALAB ;BRANCH IF DONE BIT IS SET
      ERROR ;DONE BIT FAILED TO SET
      BR ALAC
ALAB: SRESET ;ISSUE RESET TO CLEAR DONE BIT
      TSTB @RXCSR ;SEE IF DONE BIT IS CLEARED
      BPL ALAC ;BRANCH IF DONE BIT IS CLEARED
      ERROR ;RESET FAILED TO CLEAR DONE BIT
      SCOPE ;SCOPE
ALAC: SCOPE
;*****
AT56: 56 ;TEST NUMBER 56 *
      AT57 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AMAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT DONE BIT (RXCSR BIT 7) IS CLEARED BY READING RXBUF.
;DONE SET BY OUTPUTTING CHARACTER WITH MAINTENANCE BIT SET (TXCSR BIT 2)
AMAA: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT (TXCSR BIT 2)
      CLR @TXBUF ;LOAD TXBUF
      TSTB @RXCSR ;WAIT FOR DONE BIT TO SET.
      BPL AMAB
      TST @RXBUF ;READ RXBUF TO CLEAR DONE BIT
      TSTB @RXCSR ;SEE IF DONE BIT IS CLEAR
      BPL AMAC ;BRANCH IF DONE BIT IS CLEAR
      ERROR ;READING RXBUF FAILED TO CLEAR DONE BIT
      SCOPE ;SCOPE
AMAB:
AMAC:

```



```

2187
2188
2189 010664 000057
2190 010666 010752
2191 010670 000012
2192 010672 010674
2193
2194
2195
2196
2197 010674 042777 000030 170202
2198 010702 052777 000014 170174
2199 010710 042777 000030 170162
2200 010716 052777 000010 170154
2201 010724 005077 170156
2202 010730 104016
2203 010732 000764
2204 010734 105777 170140
2205 010740 100401
2206 010742 104003
2207 010744 005777 170132
2208 010750 104012
2209
2210
2211 010752 000060
2212 010754 011040
2213 010756 000012
2214 010760 010762
2215
2216
2217
2218
2219 010762 042777 000030 170114
2220 010770 052777 000024 170106
2221 010776 042777 000030 170074
2222 011004 052777 000020 170066
2223 011012 005077 170070
2224 011016 104016
2225 011020 000620
2226 011022 105777 170052
2227 011026 100401
2228 011030 104003
2229 011032 005777 170044
2230 011036 104012
2231
2232 011040 000061
2233 011042 011112
2234 011044 000012
2235 011046 011050
2236
2237
2238
2239
2240 011050 052777 000034 170026
2241 011056 052777 000030 170014
2242 011064 005077 170016

```

```

*****
AT57: 57 ;TEST NUMBER 57 *
      AT60 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      ANAA ;SCOPE ENTRY POINT *
*****
;TEST THAT DONE BIT (RXCSR BIT 7) SETS NO LATER THAN 500 MSECS AFTER OUTPUTTING
;CHARACTER WITH MAINTENANCE BIT SET AND RECEIVE SPEED SET TO 01 (TRANSMIT
;SPEED ALSO SET TO 01
ANAA: BIC #30,@TXCSR ;SET MAINTENANCE BIT AND SET
      BIS #14,@TXCSR ;TX SPEED=01
      BIC #30,@RXCSR ;SET RX SPEED =01
      BIS #10,@RXCSR
      CLR @TXBUF ;LOAD TXBUF
      DELAY 500. ;DELAY 500 MSECS.
      TSTB @RXCSR ;SEE IF DONE BIT IS SET.
      BMI ANAB ;BRANCH IF DONE IS SET.
      ERROR ;DONE FAILED TO SET WITH RX SPEED=01.
ANAB: TST @RXBUF ;CLEAR DONE BIT IF SET.
      SCOPE ;SCOPE
*****
AT60: 60 ;TEST NUMBER 60 *
      AT61 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AOAA ;SCOPE ENTRY POINT *
*****
;TEST THAT DONE BIT (RXCSR BIT 7) SETS NO LATER THAN 400 MSECS AFTER OUTPUTTING
;CHARACTER WITH MAINTENANCE BIT SET AND RECEIVE SPEED SET TO 10 (TRANSMIT
;SPEED ALSO SET TO 10).
AOAA: BIC #30,@TXCSR ;SET MAINTENANCE BIT AND SET
      BIS #24,@TXCSR ;TX SPEED=10.
      BIC #30,@RXCSR ;SET RX SPEED=10.
      BIS #20,@RXCSR
      CLR @TXBUF ;LOAD TXBUF
      DELAY 400. ;DELAY 400 MSECS
      TSTB @RXCSR ;SEE IF DONE BIT IS SET.
      BMI AOAB ;BRANCH IF DONE BIT IS SET.
      ERROR ;DONE FAILED TO SET WITH RX SPEED=10.
AOAB: TST @RXBUF ;CLEAR DONE BIT IF SET
      SCOPE ;SCOPE
*****
AT61: 61 ;TEST NUMBER 61 *
      AT62 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      APAA ;SCOPE ENTRY POINT *
*****
;TEST THAT DONE BIT (RXCSR BIT 7) SETS NO LATER THAN 250 MSECS AFTER OUTPUTTING
;CHARACTER WITH MAINTENANCE BIT SET AND RECEIVE SPEED SET 11 (TRANSMIT SPEED
;ALSO SET TO 11).
APAA: BIS #34,@TXCSR ;SET MAINT BIT AND TX SPEED=11
      BIS #30,@RXCSR ;SET RX SPEED=11
      CLR @TXBUF ;LOAD TXBUF

```

K04

.MAIN. MACY11 27(732) 28-SEP-76 13:35 PAGE 50
DZDCAB.P11

2243	011070	104016		DELAY					
2244	011072	000372		250.					;DELAY 250 MSECS.
2245	011074	105777	170000	TSTB	@RXCSR				;SEE IF DONE BIT IS SET.
2246	011100	100401		BMI	APAB				;BRANCH IF DONE BIT IS SET.
2247	011102	104003		ERROR					;DONE FAILED TO SET WITH RX SPEED=11
2248	011104	005777	167772	TST	@RXBUF				;CLEAR DONE BIT IF SET.
2249	011110	104012		SCOPE					;SCOPE

```

2250
2251
2252 011112 000062
2253 011114 011414
2254 011116 000144
2255 011120 011122
2256
2257
2258
2259
2260 011122 005067 170120
2261 011126 005067 170116
2262 011132 005067 170114
2263 011136 005067 170112
2264 011142 042777 000030 167734
2265 011150 052777 000004 167726
2266 011156 042777 000030 167714
2267 011164 004767 000154
2268 011170 066767 000216 170050
2269 011176 042777 000030 167700
2270 011204 052777 000010 167672
2271 011212 042777 000030 167660
2272 011220 052777 000010 167652
2273 011226 004767 000112
2274 011232 066767 000154 170010
2275 011240 042777 000030 167636
2276 011246 052777 000020 167630
2277 011254 042777 000030 167616
2278 011262 052777 000020 167610
2279 011270 004767 000050
2280 011274 066767 000112 167750
2281 011302 052777 000030 167574
2282 011310 052777 000030 167562
2283 011316 004767 000022
2284 011322 066767 000064 167724
2285 011330 004767 002720
2286 011334 000402
2287 011336 104015
2288 011340 015650
2289 011342 104012
2290 011344 005067 000042
2291 011350 105777 167530
2292 011354 100375
2293 011356 104016
2294 011360 000024
2295 011362 005777 167514
2296 011366 005077 167514
2297 011372 104016
2298 011374 000001
2299 011376 005267 000010
2300 011402 105777 167472
2301 011406 100371
2302 011410 000207
2303 011412 000000

```

```

*****
AT62: 62 ;TEST NUMBER 62 *
      AT63 ;ADDRESS OF NEXT TEST *
      100 ;TEST ITERATION COUNT *
      AQWA ;SCOPE ENTRY POINT *
*****
;TEST THAT RECEIVE SPEEDS ARE ARRANGED IN ASCENDING ORDER BY CHECKING THAT TIME
;ELAPSED TO DONE BIT SETTING (RXCSR BIT 7) DECREASES AS A HIGHER SPEED
;IS SELECTED. THE TRANSMIT SPEED SELECTED WILL CORRESPOND TO THE SELECTED RECEIVE SPEED
AQAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD
      CLR CTRB ;(ELAPSED TIME COUNTERS)
      CLR CTCR
      CLR CTRD
      PIC #30,ATXCSR ;SELECT TX SPEED 00
      BIS #BIT2,ATXCSR ;SET MAINTENANCE BIT
      BIC #30,ARXCSR ;SELECT RX SPEED 00
      JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
      ADD AQAST,CTRA ;ADD ELAPSED TIME TO CTRA
      BIC #30,ATXCSR ;SELECT TX SPEED 01
      BIS #10,ATXCSR
      BIC #30,ARXCSR ;SELECT RX SPEED 01.
      BIS #10,ARXCSR
      JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
      ADD AQAST,CTRB ;ADD ELAPSED TIME TO CTRB
      BIC #30,ATXCSR ;SELECT TX SPEED 10
      BIS #20,ATXCSR
      BIC #30,ARXCSR ;SELECT RX SPEED 10
      BIS #20,ARXCSR
      JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT.
      ADD AQAST,CTRC ;ADD ELAPSED TIME TO CTCR.
      BIS #30,ATXCSR ;SELECT TX SPEED 11
      BIS #30,ARXCSR ;SELECT RX SPEED 11
      JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
      ADD AQAST,CTRD ;ADD ELAPSED TIME TO CTRD.
      JSR %7,CMPT ;CHECK THAT CTRA THROUGH CTRD CONTAIN
      BR AQAB ;DESCENDING VALUES.
      ERROR1 ;RECEIVE SPEEDS NOT ARRANGED IN
      ERXTIM ;ASCENDING ORDER.
      AQAB: SCOPE ;SCOPE
      AQAS: CLR AQAST ;CLEAR ELAPSED TIME COUNTER AQAST
           TSTB ATXCSR ;WAIT FOR TX READY.
           BPL -.4
           DELAY 20 ;WAIT 20 MSECS.
           TST ARXBUF ;CLEAR DONE BIT IF SET
           CLR ATXBUF ;LOAD TXBUF
      AQASA: DELAY 1 ;DELAY 1 MSEC
           INC AQAST ;INCREMENT ELAPSED TIME COUNTER
           TSTB ARXCSR ;DONE SET?
           BPL AQASA ;BRANCH IF DONE NOT SET
           RTS %7 ;EXIT
      AQAST: OPEN ;ELAPSED TIME COUNTER

```

```

2304
2305
2306 011414 000063
2307 011416 011652
2308 011420 000144
2309 011422 011424
2310
2311
2312
2313
2314 011424 005067 167616
2315 011430 005067 167614
2316 011434 005067 167612
2317 011440 005067 167610
2318 011444 042777 003000 167426
2319 011452 004767 000116
2320 011456 066767 000166 167562
2321 011464 042777 003000 167406
2322 011472 052777 001000 167400
2323 011500 004767 000070
2324 011504 066767 000140 167536
2325 011512 042777 003000 167360
2326 011520 052777 002000 167352
2327 011526 004767 000042
2328 011532 066767 000112 167512
2329 011540 052777 003000 167332
2330 011546 004767 000022
2331 011552 066767 000072 167474
2332 011560 004767 002470
2333 011564 000402
2334 011566 104015
2335 011570 015712
2336 011572 104012
2337 011574 005067 000050
2338 011600 105777 167300
2339 011604 100375
2340 011606 104016
2341 011610 000024
2342 011612 005777 167264
2343 011616 052777 000004 167260
2344 011624 005077 167256
2345
2346 011630 104016
2347 011632 000001
2348 011634 005267 000010
2349 011640 105777 167234
2350 011644 100371
2351 011646 000207
2352 011650 000000

```

```

*****
AT63: 63 ;TEST NUMBER 63 *
      AT64 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ARAA ;SCOPE ENTRY POINT *
*****
;TEST FOR CORRECT OPERATION OF CHARACTER LENGTH SELECTION DURING RECEIVE
;(RXCSR BITS 9 AND 10) BY CHECKING THAT TIME REQUIRED TO RECEIVE A CHARACTER
;IS LONGEST FOR 8 BIT CODE THAN FOR 7 BIT CODE ETC.
ARAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD
      CLR CTRB ;(ELAPSED TIME COUNTERS)
      CLR CTCR
      CLR CTRD
      BIC #3000, @RXCSR ;SET CHAR LENGTH TO 00 (8 BIT CODE)
      JSR %7, ARAS ;OUTPUT CHAR AND TIME DONE BIT.
      ADD ARAS, CTRA ;ADD ELAPSED TIME TO CTRA
      BIC #3000, @RXCSR ;SET CHAR LENGTH TO 01 (7 BIT CODE)
      BIS #1000, @RXCSR
      JSR %7, ARAS ;OUTPUT CHAR AND TIME DONE BIT
      ADD ARAS, CTRB ;ADD ELAPSED TIME TO CTRB
      BIC #3000, @RXCSR ;SET CHAR LENGTH TO 10 (6 BIT CODE)
      BIS #2000, @RXCSR
      JSR %7, ARAS ;OUTPUT CHAR AND TIME DONE BIT
      ADD ARAS, CTCR ;ADD ELAPSED TIME TO CTCR
      BIS #3000, @RXCSR ;SET CHAR LENGTH TO 11 (5 BIT CODE)
      JSR %7, ARAS ;OUTPUT CHAR AND TIME DONE BIT
      ADD ARAS, CTRD ;ADD ELAPSED TIME TO CTRD
      JSR %7, CMPT ;CHECK THAT CTRA THROUGH CTRD
      BR ARAB ;CONTAIN DESCENDING VALUES
      ERROR1 ;RECEIVE CHARACTER LENGTHS NOT ARRANGED
      ERCLGT ;IN DESCENDING ORDER
ARAB: SCOPE
ARAS: CLR ARAS ;CLEAR ELAPSED TIME COUNTER ARAS
      TSTB @TXCSR ;WAIT FOR TX READY.
      BPL .-4
      DELAY ;WAIT 20 MSECS.
      20.
      TST @RXBUF ;CLEAR DONE BIT IF SET
      BIS #BIT2, @TXCSR ;SET MAINTENANCE BIT
      CLR @TXBUF ;LOAD TXBUF
ARASA: DELAY ;DELAY 1 MSEC.
      1
      INC ARAS ;INCREMENT ELAPSED TIME COUNTER
      TSTB @RXCSR ;SEE IF DONE BIT IS SET.
      BPL ARASA ;BRANCH IF NOT SET
      RTS %7 ;EXIT
ARAST: OPEN ;ELAPSED TIME COUNTER

```

```

2353
2354
2355 011652 000064
2356 011654 011766
2357 011656 000144
2358 011660 011662
2359
2360
2361 011662 004767 000060
2362 011666 004767 000054
2363 011672 017767 167202 167360
2364 011700 032767 010000 167352
2365 011706 001002
2366 011710 104003
2367 011712 000412
2368 011714 005767 167340
2369 011720 100402
2370 011722 104003
2371
2372 011724 000405
2373 011726 032777 010000 167144
2374
2375 011734 001401
2376 011736 104003
2377
2378 011740 005777 167136
2379 011744 104012
2380 011746 052777 000004 167130
2381 011754 005077 167126
2382 011760 104016
2383 011762 000310
2384 011764 000207
2385
2386
2387 011766 000065
2388 011770 012040
2389 011772 000012
2390 011774 012006
2391
2392
2393
2394 011776 004767 170562
2395 012002 104007
2396 012004 012034
2397 012006 042777 000100 167070
2398 012014 005067 165756
2399 012020 052777 000104 167056
2400 012026 000240
2401 012030 104003
2402 012032 104012
2403 012034 022111
2404 012036 000775
2405

*****
AT64: 64 ;TEST NUMBER 64 *
      AT65 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ASAA ;SCOPE ENTRY POINT *
*****
;TEST CORRECT OPERATION OF DATA OVERRUN BIT (RXCSR BIT 12)
ASAA: JSR %7,ASAS ;OUTPUT CHARACTER AND WAIT 200 MSECS
      JSR %7,ASAS ;OUTPUT CHARACTER AND WAIT 200 MSECS
      MOV @RXCSR,RXCSR ;SAVE RXCSR CONTENTS
      BIT #BIT12,RXCSR ;SEE IF DATA OVERRUN BIT WAS SET
      BNE ASAB ;BRANCH IF BIT WAS SET
      ERROR
      BR ASAD
ASAB: TST RXCSR ;SEE IF ERROR BIT WAS SET (RXCSR BIT 15)
      BMI ASAC
      ERROR ;ERROR BIT FAILED TO SET
      ;WHEN DATA OVERRUN SET
      BR ASAD
ASAC: BIT #BIT12,@RXCSR ;SEE IF DATA OVERRUN WAS
      ;CLEARED WHEN RXCSR WAS READ
      BEQ ASAD ;BRANCH IF CLEAR
      ERROR ;READING RXCSR FAILED
      ;TO CLEAR DATA OVERRUN
ASAD: TST @RXBUF ;CLEAR DONE BIT (RXCSR BIT 7)
      SCOPE ;SCOPE
ASAS: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
      CLR @TXBUF ;LOAD TXBUF
      DELAY 200. ;DELAY 200 MSECS
      RTS %7 ;EXIT
*****
AT65: 65 ;TEST NUMBER 65 *
      AT66 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AT67 ;SCOPE ENTRY POINT *
*****
;TEST THAT TRANSMITTER IS ABLE TO INTERRUPT. IF THE INTERRUPT IS SERVICED,
;IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
      JSR 7,OVRLAY ;GO TO OVER LAY ROUTINE
      STTXV ;SET TX INTERRUPT SERVICE
      ATAC ;TO ATAC
ATAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPT
      CLR PSW ;SET PROCESSOR PRIORITY TO 0
      BIS #104,@TXCSR ;ENABLE TX INTERRUPT
      NOP
      ERROR ;READY DID NOT CAUSE AN INTERRUPT
ATAB: SCOPE ;SCOPE
ATAC: POPSP2 ;HERE IF INTERRUPT IS SERVICED. POP
      BR ATAB ;THE STACK TWICE

```

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

012040 000066
012042 012116
012044 001750
012046 012054

012050 104007
012052 012106
012054 016767
012062 042777
012070 052777
012076 000240

012100 042777
012106 104012
012110 022626
012112 104003
012114 000774

012116 000067
012120 012202
012122 000012
012124 012132

012126 104007
012130 012170
012132 042777
012140 016767
012146 162767
012154 052777
012162 000240
012164 104003
012166 000401
012170 022626
012172 042777
012200 104012

167036 165714
000100 167014
000104 167006

000100 166776

000100 166744
166752 165630
000040 165622
000104 166722

000100 166704

```
*****  
AT66: 66 ;TEST NUMBER 66 *  
      AT67 ;ADDRESS OF NEXT TEST *  
      1000. ;TEST ITERATION COUNT *  
      AUA ;SCOPE ENTRY POINT *  
*****  
;TEST THAT READY DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS  
;AT THE SAME PRIORITY AS THE TRANSMITTER INTERRUPT REQUEST LEVEL  
      STTXV ;SET TX INTERRUPT SERVICE TO  
      AUAC  
AUAA: MOV TXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS TX PRIORITY  
      BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS  
      BIS #104,@TXCSR ;ENABLE TX INTERRUPTS  
      NOP  
  
AUAB: BIC #BIT6,@TXCSR ;OK IF NO INTERRUPT OCCURS. DISABLE INTERRUPTS  
AUAC: SCOPE ;SCOPE  
      POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE  
      ERROR ;TX INTERRUPTED WITH PROCESSOR AT SAME  
      BR AUAC ;PRIORITY AS THE TRANSMITTER  
  
*****  
AT67: 67 ;TEST NUMBER 67 *  
      AT70 ;ADDRESS OF NEXT TEST *  
      10. ;TEST ITERATION COUNT *  
      AVAA ;SCOPE ENTRY POINT *  
*****  
;TEST THAT TRANSMITTER INTERRUPTS WHEN PROCESSOR IS AT PRIORITY ONE LEVEL  
;LOWER THAN THE TRANSMITTER INTERRUPT PRIORITY.  
      STTXV ;SET TX INTERRUPT SERVICE TO AVAB  
      AVAB  
AVAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS  
      MOV TXLVL,PSW ;SET PROCESSOR PRIORITY TO ONE LEVEL  
      SUB #40,PSW ;LOWER THAN TX PRIORITY  
      BIS #104,@TXCSR ;ENABLE TX INTERRUPTS  
      NOP  
      ERROR ;TX FAILED TO INTERRUPT  
      BR AVAC  
AVAB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE  
AVAC: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS  
      SCOPE ;SCOPE
```

```

2450
2451
2452 012202 000070
2453 012204 012300
2454 012206 000144
2455 012210 012212
2456
2457
2458
2459 012212 104007
2460 012214 012252
2461 012216 042777 000100 166660
2462 012224 005067 165546
2463 012230 052777 000104 166646
2464 012236 000240
2465 012240 104003
2466 012242 042777 000100 166634 AWAB: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2467 012250 104012 AWAC: SCOPE ;SCOPE
2468 012252 012777 012272 166634 AWAC: MOV #AWAE,@TXVTR ;HERE IF INTERRUPT OCCURS. CHANGE EXIT
2469 012260 012716 012266 MOV #AWAD,@%6 ;POINTER TO AWAD AND EXIT INTERRUPT
2470 012264 000002
2471 012266 000240 AWAD: NOP ;OK IF NO INTERRUPT REOCCURS.
2472 012270 000764 BR AWAB
2473 012272 022626 AWAE: POPSP2 ;HERE IF INTERRUPT REOCCURS
2474 012274 104003 ERROR ;TX REINTERRUPTED AFTER RTI
2475 012276 000761 BR AWAB
2476
2477
2478 012300 000071
2479 012302 012356
2480 012304 000012
2481 012306 012324
2482
2483
2484
2485 012310 004767 170250
2486 012314 104006
2487 012316 012352
2488 012320 004767 001710
2489 012324 042777 000100 166546 AXAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2490 012332 005067 165440 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2491 012336 052777 000100 166534 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2492 012344 000240
2493 012346 104003
2494 012350 000401
2495 012352 022626 AXAB: POPSP2 ;HERE IF INTERRUPT OCCURS
2496 012354 104012 AXAC: SCOPE ;SCOPE
2497

```

```

2498
2499
2500 012356 000072
2501 012360 012440
2502 012362 001750
2503 012364 012376
2504
2505
2506
2507 012366 104006
2508 012370 012432
2509 012372 004767 001636
2510 012376 042777 000100 166474 AYAA:
2511 012404 016767 166502 165364
2512 012412 052777 000100 166460
2513 012420 000240
2514 012422 042777 000100 166450 AYAB:
2515 012430 104012
2516 012432 022626 AYAC:
2517 012434 104003
2518 012436 000771
2519
2520
2521 012440 000073
2522 012442 012530
2523 012444 000012
2524 012446 012460
2525
2526
2527
2528 012450 104006
2529 012452 012516
2530 012454 004767 001554
2531 012460 042777 000100 166412 AZAA:
2532 012466 016767 166420 165302
2533 012474 162767 000040 165274
2534 012502 052777 000100 166370
2535 012510 000240
2536 012512 104003
2537 012514 000401
2538
2539 012516 022626 AZAB:
2540 012520 042777 000100 166352 AZAC:
2541 012526 104012

```

```

*****
AT72: 72 ;TEST NUMBER 72 *
      AT73 ;ADDRESS OF NEXT TEST *
      1000. ;TEST ITERATION COUNT *
      AYAA ;SCOPE ENTRY POINT *
*****
;TEST THAT RECEIVER DONE BIT DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR
;IS AT THE SAME PRIORITY LEVEL AS THE RECEIVER INTERRUPT REQUEST LEVEL
      STRXV ;SET RX INTERRUPT SERVICE TO AYAC
      AYAC
      JSR %7,STRXD ;SET RX DONE BIT
      BIC #BIT6,ARXCSR ;DISABLE RX INTERRUPTS
      MOV RXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS RECEIVER'S
      BIS #BIT6,ARXCSR ;ENABLE RX INTERRUPTS
      NOP
      BIC #BIT6,ARXCSR ;OK IF NO INTERRUPT. DISABLE RX INTERRUPTS
      SCOPE ;SCOPE
      POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
      ERROR ;RX INTERRUPTED WITH PROCESOR AT SAME
      BR AYAB ;PRIORITY AS THE RECEIVER
*****
AT73: 73 ;TEST NUMBER 73 *
      AT74 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AZAA ;SCOPE ENTRY POINT *
*****
;TEST THAT RECEIVER DONE BIT CAUSES INTERRUPT WHEN PROCESSOR IS AT PRIORITY
;ONE LEVEL LOWER THAN THE RECEIVER'S INTERRUPT REQUEST LEVEL
      STRXV ;SET RX INTERRUPT TO AZAB
      AZAB
      JSR %7,STRXD ;SET RX DONE BIT
      BIC #BIT6,ARXCSR ;DISABLE RX INTERRUPTS
      MOV RXLVL,PSW ;SET PROCESSOR PRIORITY ONE LEVEL
      SUB #40,PSW ;LOWER THAN RECEIVER'S PRIORITY
      BIS #BIT6,ARXCSR ;ENABLE RX INTERRUPTS
      NOP
      ERROR ;RX FAILED TO INTERRUPT WITH PROCESSOR AT
      BR AZAC ;PRIORITY ONE LEVEL LOWER THAN RECEIVER'S
*****
AZAB: POPSP2 ;HERE IF INTERRUPT OCCURS
AZAC: BIC #BIT6,ARXCSR ;DISABLE RX INTERRUPTS
      SCOPE ;SCOPE

```



```

2542
2543
2544 012530 000074
2545 012532 012626
2546 012534 000144
2547 012536 012544
2548
2549
2550
2551 012540 004767 001470
2552 012544 104006
2553 012546 012600
2554 012550 042777 000100 166322
2555 012556 052777 000100 166314
2556 012564 000240
2557 012566 104003
2558 012570 042777 000100 166302
2559 012576 104012
2560 012600 012777 012620 166302
2561 012606 012716 012614
2562 012612 000002
2563 012614 000240
2564 012616 000764
2565 012620 022626
2566 012622 104003
2567 012624 000761
2568
2569
2570 012626 000075
2571 012630 012702
2572 012632 000144
2573 012634 012636
2574
2575
2576 012636 004767 001372
2577 012642 005077 166240
2578 012646 104016
2579 012650 000024
2580 012652 017767 166222 166400
2581 012660 105777 166214
2582 012664 100001
2583 012666 104003
2584 012670 104016
2585 012672 000310
2586 012674 005777 166202
2587 012700 104012
2588

;*****
AT74: 74 ;TEST NUMBER 74 *
      AT75 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AABA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RECEIVER DOES NOT INTERRUPT AFTER THE INITIAL INTERRUPT HAS
;OCCURED AND DONE BIT HAS NOT BEEN CLEARED
AABA: JSR %7,STRXD ;SET RX DONE BIT
      STRXV ;SET RX INTERRUPT SERVICE TO AABC
      AABC
      BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
      BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
      NOP
      ERROR ;RX FAILED TO INTERRUPT
AABB: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
      SCOPE ;SCOPE
AABC: MOV #AABE,@RXVTR ;HERE IF INTERRUPT OCCURS. CHANGE SERVICE TO
      MOV #AABD,@%6 ;AABE, SET EXIT POINTER TO AABD
      RTI ;EXIT INTERRUPT SERVICE
AABD: NOP ;OK IF NO INTERRUPT REOCCURS
      BR AABB
AABE: POPSP2 ;HERE IF INTERRUPT REOCCURS
      ERROR ;RX REINTERRUPTED AFTER RTI
      BR AABB

;*****
AT75: 75 ;TEST NUMBER 75 *
      AT76 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ABBA ;SCOPE ENTRY POINT *
;*****
;TEST THAT DATA OVERRUN (RXCSR BIT 12) CLEARS THE DONE BIT (RXCSR BIT 7)
ABBA: JSR %7,STRXD ;SET RX DONE BIT
      CLR @TXBUF ;LOAD TXBUF
      DELAY ;WAIT 20 MSECS.
      20.
      MOV @RXCSR,RXCST ;SAVE CONTENT OF RXCSR
      TSTB @RXCSR ;SEE IF DONE BIT IS CLEAR
      BPL ABBB ;BRANCH IF DONE BIT IS CLEAR
      ERROR
ABBB: DELAY ;WAIT FOR RX DONE TO SET.
      200.
      TST @RXBUF ;CLEAR DONE BIT IF SET
      SCOPE ;SCOPE

```

```

2589
2590
2591 012702 000076
2592 012704 012760
2593 012706 000144
2594 012710 012716
2595
2596
2597 012712 104006
2598 012714 012754
2599 012716 004767 001312
2600 012722 004767 001306
2601 012726 042777 000100 166144
2602 012734 005057 165036
2603 012740 052777 000100 166132
2604 012746 000240
2605 012750 104003
2606 012752 000401
2607 012754 022626
2608 012756 104012
2609
2610
2611
2612
2613 012760 000077
2614 012762 013142
2615 012764 000144
2616 012766 013002
2617
2618
2619 012770 004567 170064
2620 012774 052777 000004 166102
2621 013002 112767 000144 166236
2622 013010 112767 000010 166231
2623 013016 004567 170142
2624 013022 105777 166056
2625 013026 100375
2626 013030 010177 166052
2627 013034 105777 166040
2628 013040 100375
2629 013042 017767 166034 166162
2630 013050 005000
2631 013052 006067 166154
2632 013056 103001
2633 013060 005100
2634 013062 105367 166161
2635 013066 001371
2636 013070 032777 000040 166002
2637 013076 001403
2638 013100 005700
2639 013102 001403
2640 013104 000412
2641 013106 005700
2642 013110 001410
2643 013112 104003
2644 013114 004567 170112

;*****
AT76: 76 ;TEST NUMBER 76 *
      AT77 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ACBA ;SCOPE ENTRY POINT *
;*****
;TEST THAT ERROR BIT (RXCSR BIT 15) IS ABLE TO CAUSE AN INTERRUPT
      STRXV ;SET RX INTERRUPT SERVICE TO ACBB.
      ACBB
ACBA: JSR %7,STRXD ;SET RX DONE BIT
      JSR %7,STRXD ;SET RX DATA OFLOW
      BIC #BIT6,ARXCSR ;DISABLE RX INTERRUPTS
      CLR PSW ;SET PROCESSOR PRIORITY TO 0
      BIS #BIT6,ARXCSR ;ENABLE RX INTERRUPTS
      NOP
      ERROR ;RX ERROR BIT FAILED TO CAUSE INTERRUPT
      BR ACBC
ACBB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
ACBC: SCOPE ;SCOPE

;*****
AT77: 77 ;TEST NUMBER 77
      AT100 ;ADDRESS OF NEXT TEST
      100. ;TEST ITERATION COUNT
      ANBB ;SCOPE ENTRY POINT
;*****
;TEST THAT PARITY INDICATOR OPERATES CORRECT.
ANBA: JSR 5,INBIN ;INITIALIZE BINARY COUNT PATTERN
      BIS #BIT2,ARXCSR ;SET MAINTENANCE BIT
ANBB: MOVB #100,CTRA ;GET CHARACTER COUNT
ANBC: MOVB #8,CTRA+1 ;GET CHARACTER BIT COUNT
      JSR 5,GTBINP ;GET A CHARACTER (IN R1)
      TSTB ARXCSR ;WAIT FOR
      BPL -4 ;TRANSMITTER READY FLAG
      MOV %1,ARXBUF ;LOAD TRANSMITTER BUFFER
      TSTB ARXCSR ;WAIT FOR
      BPL -4 ;RECEIVER READY FLAG
      MOV ARXBUF,CRBUFA ;GET RECEIVED CHARACTER
      CLR %0 ;CLEAR WORKING REGISTER
ANBD: ROR CRBUFA ;LOOK AT CHARACTER BITS
      BCC +4 ;AND COMPLEMENT R0 WHEN
      COM %0 ;A 1 IS RECEIVED
      DECB CTRA+1 ;IF R0=1'S, ODD#1'S RECEIVED
      BNE ANBD ;IF R0=0'S, EVEN #1'S RECEIVED
      BIT #BIT5,ARXCSR ;TEST PARITY INDICATOR
      BEQ ANBE ;BRANCH IF INDICATES EVEN
      TST %0 ;TEST RECEIVED PARITY (IN R0)
      BEQ ANBF ;ERROR BRANCH
      BR ANBG ;OK BRANCH
ANBE: TST %0 ;TEST RECEIVED PARITY (IN R0)
      BEQ ANBG ;OK BRANCH
ANBF: ERROR ;TYPE PC
      JSR 5,OACNV ;GO TO OCTAL

```

2645 013120 001232
 2646 013122 016344
 2647 013124 000003
 2648 013126 104015
 2649 013130 016344
 2650 013132 105367
 2651 013136 001324
 2652 013140 104012
 2653 000077
 2654 000000
 2655
 2656 013142 000100
 2657 013144 013160
 2658 013146 000003
 2659 013150 013152
 2660 000100
 2661
 2662 013152 004567
 2663 013156 000000
 2664 000001
 2665
 2666 013160 000101
 2667 013162 013176
 2668 013164 000003
 2669 013166 013170
 2670 000101
 2671
 2672 013170 004567
 2673 013174 000001
 2674 000002
 2675
 2676 013176 000102
 2677 013200 013214
 2678 013202 000003
 2679 013204 013206
 2680 000102
 2681
 2682 013206 004567
 2683 013212 000002
 2684 000003
 2685
 2686 013214 000103
 2687 013216 013232
 2688 013220 000003
 2689 013222 013224
 2690 000103
 2691
 2692 013224 004567
 2693 013230 000003
 2694 000004
 2695
 2696 013232 000104
 2697 013234 013250
 2698 013236 000003
 2699 013240 013242
 2700 000104

166110

170306

170270

170252

170234

ANBG: CKBUFA
 AWAS
 3
 ERROR1
 AWAS
 DECB CTRA
 BNE ANBC
 SCOPE
 X=77
 Y=0

;TO ASCII
 ;ROUTINE AND
 ;CONVERT DATA
 ;TYPE
 ;DATA
 ;DECREMENT CHARACTER COUNT

 AT100: 100 ;ROUTINE #100 *
 AT101 ;ADDRESS OF NEXT TEST *
 3. ;ITERATION COUNT *
 DAT0 ;SCOPE ENTRY POINT *
 X=X+1

 DAT0: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
 0 ;SEE NOTE 0 FOR DATA TEST PARAMETERS
 Y=Y+1

 AT101: 101 ;ROUTINE #101 *
 AT102 ;ADDRESS OF NEXT TEST *
 3. ;ITERATION COUNT *
 DAT1 ;SCOPE ENTRY POINT *
 X=X+1

 DAT1: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
 1 ;SEE NOTE 1 FOR DATA TEST PARAMETERS
 Y=Y+1

 AT102: 102 ;ROUTINE #102 *
 AT103 ;ADDRESS OF NEXT TEST *
 3. ;ITERATION COUNT *
 DAT2 ;SCOPE ENTRY POINT *
 X=X+1

 DAT2: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
 2 ;SEE NOTE 2 FOR DATA TEST PARAMETERS
 Y=Y+1

 AT103: 103 ;ROUTINE #103 *
 AT104 ;ADDRESS OF NEXT TEST *
 3. ;ITERATION COUNT *
 DAT3 ;SCOPE ENTRY POINT *
 X=X+1

 DAT3: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
 3 ;SEE NOTE 3 FOR DATA TEST PARAMETERS
 Y=Y+1

 AT104: 104 ;ROUTINE #104 *
 AT105 ;ADDRESS OF NEXT TEST *
 3. ;ITERATION COUNT *
 DAT4 ;SCOPE ENTRY POINT *
 X=X+1

```

2701
2702 013242 004567 170216
2703 013246 000004
2704 000005
2705
2706 013250 000105
2707 013252 013266
2708 013254 000003
2709 013256 013260
2710 000105
2711
2712 013260 004567 170200
2713 013264 000005
2714 000006
2715
2716 013266 000106
2717 013270 013304
2718 013272 000003
2719 013274 013276
2720 000106
2721
2722 013276 004567 170162
2723 013302 000006
2724 000007
2725
2726 013304 000107
2727 013306 013322
2728 013310 000003
2729 013312 013314
2730 000107
2731
2732 013314 004567 170144
2733 013320 000007
2734 000010
2735
2736 013322 000110
2737 013324 013340
2738 013326 000003
2739 013330 013332
2740 000110
2741
2742 013332 004567 170126
2743 013336 000010
2744 000011
2745
2746 013340 000111
2747 013342 013356
2748 013344 000003
2749 013346 013350
2750 000111
2751
2752 013350 004567 170110
2753 013354 000011
2754 000012
2755
2756 013356 000112

```

```

*****
DAT4: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      4 ;SEE NOTE 4 FOR DATA TEST PARAMETERS
      Y=Y+1
*****
AT105: 105 ;ROUTINE #105 *
      AT106 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT5 ;SCOPE ENTRY POINT *
      X=X+1
*****
DAT5: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      5 ;SEE NOTE 5 FOR DATA TEST PARAMETERS
      Y=Y+1
*****
AT106: 106 ;ROUTINE #106 *
      AT107 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT6 ;SCOPE ENTRY POINT *
      X=X+1
*****
DAT6: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      6 ;SEE NOTE 6 FOR DATA TEST PARAMETERS
      Y=Y+1
*****
AT107: 107 ;ROUTINE #107 *
      AT110 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT7 ;SCOPE ENTRY POINT *
      X=X+1
*****
DAT7: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      7 ;SEE NOTE 7 FOR DATA TEST PARAMETERS
      Y=Y+1
*****
AT110: 110 ;ROUTINE #110 *
      AT111 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT10 ;SCOPE ENTRY POINT *
      X=X+1
*****
DAT10: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      10 ;SEE NOTE 10 FOR DATA TEST PARAMETERS
      Y=Y+1
*****
AT111: 111 ;ROUTINE #111 *
      AT112 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT11 ;SCOPE ENTRY POINT *
      X=X+1
*****
DAT11: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      11 ;SEE NOTE 11 FOR DATA TEST PARAMETERS
      Y=Y+1
*****
AT112: 112 ;ROUTINE #112 *

```

```

2757 013360 013374          AT113          ;ADDRESS OF NEXT TEST          *
2758 013362 000003          3.          ;ITERATION COUNT              *
2759 013364 013366          DAT12          ;SCOPE ENTRY POINT            *
2760          000112          X=X+1
2761          *****
2762 013366 004567 170072 DAT12: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2763 013372 000012          12          ;SEE NOTE 12 FOR DATA TEST PARAMETERS
2764          000013          Y=Y+1
2765          *****
2766 013374 000113          AT113: 113          ;ROUTINE #113                  *
2767 013376 013412          AT114          ;ADDRESS OF NEXT TEST          *
2768 013400 000003          3.          ;ITERATION COUNT              *
2769 013402 013404          DAT13          ;SCOPE ENTRY POINT            *
2770          000113          X=X+1
2771          *****
2772 013404 004567 170054 DAT13: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2773 013410 000013          13          ;SEE NOTE 13 FOR DATA TEST PARAMETERS
2774          000014          Y=Y+1
2775          *****
2776 013412 000114          AT114: 114          ;ROUTINE #114                  *
2777 013414 013430          AT115          ;ADDRESS OF NEXT TEST          *
2778 013416 000003          3.          ;ITERATION COUNT              *
2779 013420 013422          DAT14          ;SCOPE ENTRY POINT            *
2780          000114          X=X+1
2781          *****
2782 013422 004567 170036 DAT14: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2783 013426 000014          14          ;SEE NOTE 14 FOR DATA TEST PARAMETERS
2784          000015          Y=Y+1
2785          *****
2786 013430 000115          AT115: 115          ;ROUTINE #115                  *
2787 013432 013446          AT116          ;ADDRESS OF NEXT TEST          *
2788 013434 000003          3.          ;ITERATION COUNT              *
2789 013436 013440          DAT15          ;SCOPE ENTRY POINT            *
2790          000115          X=X+1
2791          *****
2792 013440 004567 170020 DAT15: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2793 013444 000015          15          ;SEE NOTE 15 FOR DATA TEST PARAMETERS
2794          000016          Y=Y+1
2795          *****
2796 013446 000116          AT116: 116          ;ROUTINE #116                  *
2797 013450 013464          AT117          ;ADDRESS OF NEXT TEST          *
2798 013452 000003          3.          ;ITERATION COUNT              *
2799 013454 013456          DAT16          ;SCOPE ENTRY POINT            *
2800          000116          X=X+1
2801          *****
2802 013456 004567 170002 DAT16: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2803 013462 000016          16          ;SEE NOTE 16 FOR DATA TEST PARAMETERS
2804          000017          Y=Y+1
2805          *****
2806 013464 000117          AT117: 117          ;ROUTINE #117                  *
2807 013466 013502          AT120          ;ADDRESS OF NEXT TEST          *
2808 013470 000003          3.          ;ITERATION COUNT              *
2809 013472 013474          DAT17          ;SCOPE ENTRY POINT            *
2810          000117          X=X+1
2811          *****
2812 013474 004567 167764 DAT17: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST

```



```

2869 013616 013620          DAT25          ;SCOPE ENTRY POINT          *
2870          000125          X=X+1
2871          :*****
2872 013620 004567 167640  DAT25: JSR      5,DATTST      ;LOAD PARAMETERS & RUN TEST
2873 013624 000025          25          ;SEE NOTE 25 FOR DATA TEST PARAMETERS
2874          000026          Y=Y+1
2875          :*****
2876 013626 000126          AT126: 126          ;ROUTINE #126          *
2877 013630 013644          AT127          ;ADDRESS OF NEXT TEST  *
2878 013632 000003          3.          ;ITERATION COUNT      *
2879 013634 013636          DAT26          ;SCOPE ENTRY POINT    *
2880          000126          X=X+1
2881          :*****
2882 013636 004567 167622  DAT26: JSR      5,DATTST      ;LOAD PARAMETERS & RUN TEST
2883 013642 000026          26          ;SEE NOTE 26 FOR DATA TEST PARAMETERS
2884          000027          Y=Y+1
2885          :*****
2886 013644 000127          AT127: 127          ;ROUTINE #127          *
2887 013646 013662          AT130          ;ADDRESS OF NEXT TEST  *
2888 013650 000003          3.          ;ITERATION COUNT      *
2889 013652 013654          DAT27          ;SCOPE ENTRY POINT    *
2890          000127          X=X+1
2891          :*****
2892 013654 004567 167604  DAT27: JSR      5,DATTST      ;LOAD PARAMETERS & RUN TEST
2893 013660 000027          27          ;SEE NOTE 27 FOR DATA TEST PARAMETERS
2894          000030          Y=Y+1
2895          :*****
2896 013662 000130          AT130: 130          ;ROUTINE #130          *
2897 013664 013700          AT131          ;ADDRESS OF NEXT TEST  *
2898 013666 000003          3.          ;ITERATION COUNT      *
2899 013670 013672          DAT30          ;SCOPE ENTRY POINT    *
2900          000130          X=X+1
2901          :*****
2902 013672 004567 167566  DAT30: JSR      5,DATTST      ;LOAD PARAMETERS & RUN TEST
2903 013676 000030          30          ;SEE NOTE 30 FOR DATA TEST PARAMETERS
2904          000031          Y=Y+1
2905          :*****
2906 013700 000131          AT131: 131          ;ROUTINE #131          *
2907 013702 013716          AT132          ;ADDRESS OF NEXT TEST  *
2908 013704 000003          3.          ;ITERATION COUNT      *
2909 013706 013710          DAT31          ;SCOPE ENTRY POINT    *
2910          000131          X=X+1
2911          :*****
2912 013710 004567 167550  DAT31: JSR      5,DATTST      ;LOAD PARAMETERS & RUN TEST
2913 013714 000031          31          ;SEE NOTE 31 FOR DATA TEST PARAMETERS
2914          000032          Y=Y+1
2915          :*****
2916 013716 000132          AT132: 132          ;ROUTINE #132          *
2917 013720 013734          AT133          ;ADDRESS OF NEXT TEST  *
2918 013722 000003          3.          ;ITERATION COUNT      *
2919 013724 013726          DAT32          ;SCOPE ENTRY POINT    *
2920          000132          X=X+1
2921          :*****
2922 013726 004567 167532  DAT32: JSR      5,DATTST      ;LOAD PARAMETERS & RUN TEST
2923 013732 000032          32          ;SEE NOTE 32 FOR DATA TEST PARAMETERS
2924          000033          Y=Y+1

```

```

2925 :*****
2926 013734 000133 AT133: 133 ;ROUTINE #133 *
2927 013736 013752 AT134 ;ADDRESS OF NEXT TEST *
2928 013740 000003 3. ;ITERATION COUNT *
2929 013742 013744 DAT33 ;SCOPE ENTRY POINT *
2930 000133 X=X+1
2931 :*****
2932 013744 004567 167514 DAT33: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
2933 013750 000033 33 ;SEE NOTE 33 FOR DATA TEST PARAMETERS
2934 000034 Y=Y+1
2935 :*****
2936 013752 000134 AT134: 134 ;ROUTINE #134 *
2937 013754 013770 AT135 ;ADDRESS OF NEXT TEST *
2938 013756 000003 3. ;ITERATION COUNT *
2939 013760 013762 DAT34 ;SCOPE ENTRY POINT *
2940 000134 X=X+1
2941 :*****
2942 013762 004567 167476 DAT34: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
2943 013766 000034 34 ;SEE NOTE 34 FOR DATA TEST PARAMETERS
2944 000035 Y=Y+1
2945 :*****
2946 013770 000135 AT135: 135 ;ROUTINE #135 *
2947 013772 014006 AT136 ;ADDRESS OF NEXT TEST *
2948 013774 000003 3. ;ITERATION COUNT *
2949 013776 014000 DAT35 ;SCOPE ENTRY POINT *
2950 000135 X=X+1
2951 :*****
2952 014000 004567 167460 DAT35: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
2953 014004 000035 35 ;SEE NOTE 35 FOR DATA TEST PARAMETERS
2954 000036 Y=Y+1
2955 :*****
2956 014006 000136 AT136: 136 ;ROUTINE #136 *
2957 014010 014024 AT137 ;ADDRESS OF NEXT TEST *
2958 014012 000003 3. ;ITERATION COUNT *
2959 014014 014016 DAT36 ;SCOPE ENTRY POINT *
2960 000136 X=X+1
2961 :*****
2962 014016 004567 167442 DAT36: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
2963 014022 000036 36 ;SEE NOTE 36 FOR DATA TEST PARAMETERS
2964 000037 Y=Y+1
2965 :*****
2966 014024 000137 AT137: 137 ;ROUTINE #137 *
2967 014026 014042 AT140 ;ADDRESS OF NEXT TEST *
2968 014030 000003 3. ;ITERATION COUNT *
2969 014032 014034 DAT37 ;SCOPE ENTRY POINT *
2970 000137 X=X+1
2971 :*****
2972 014034 004567 167424 DAT37: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
2973 014040 000037 37 ;SEE NOTE 37 FOR DATA TEST PARAMETERS
2974 000040 Y=Y+1

```



```

2975
2976 014042 000140
2977 014044 014156
2978 014046 000012
2979 014050 014072
2980
2981
2982
2983
2984
2985 014052 012777 000430 165024
2986 014060 012777 003031 165012
2987 014066 004767 166766
2988 014072 012767 001750 165146
2989 014100 105777 165000
2990 014104 100375
2991 014106 004767 167052
2992 014112 110167 167114
2993 014116 042767 177740 165106
2994 014124 110177 164756
2995 014130 105777 164744
2996 014134 100375
2997 014136 117767 164740 165064
2998 014144 104004
2999 014146 005367 165074
3000 014152 001352
3001 014154 104012
3002
3003 014156 000141
3004 014160 177777
3005 014162 000144
3006 014164 014166
3007
3008
3009
3010
3011 014166 012777 000004 164710
3012 014174 012777 000002 164676
3013 014202 012777 000377 164676
3014 014210 105777 164664
3015 014214 100375
3016 014216 027727 164660 000000
3017 014224 001401
3018 014226 104003
3019 014230 104011
3020 014232 104012
3021
3022 014234 052777 000004 164642
3023 014242 005077 164640
3024 014246 104016
3025 014250 000310
3026 014252 000207
3027
3028 014254 026767 164766 164766
3029 014262 101424
3030 014264 026767 164756 164760

```

```

AT140: 140 ;TEST NUMBER 140
        AT141 ;ADDRESS OF NEXT TEST
        10. ;TEST ITERATION COUNT
        APBA ;SCOPE ENTRY POINT
;*****
;DATA TEST USING JUMPER CONNECTOR. TX SPEED = 11, RX SPEED = 11.
;CHAR LENGTH = 11, STOP CODE = 1. USES SPECIAL BINARY COUNT PATTERN
;FOR DATA. NO INTERRUPT.
;*****
APBA: MOV #430, @TXCSR ;SET TX SPEED = 11, STOP CODE = 1
      MOV #3031, @RXCSR ;SET RX SPEED = 11, CHAR. LENGTH = 11
      JSR 7, INBIN ;INITIALIZE BINARY COUNT PATTERN
      MOV #1000, CTRA ;SET CHARACTER COUNT TO 1000
APBB: TSTB @TXCSR ;WAIT FOR TX READY
      BPL -4
      JSR 7, GTBINP ;GET BINARY CHARACTER
      MOVB %1, CRBUFA ;SAVE CHAR IN CRBUFA AND
      BIC #177740, CRBUFA ;MASK OFF ALL BUT 5 LSB.
      MOVB %1, @TXBUF ;LOAD CHAR.
      TSTB @RXCSR ;WAIT FOR RECEIVER
      BPL -4 ;TO RECEIVE CHARACTER
      MOVB @RXBUF, CRBUF ;LOAD RECEIVED DATA INTO CRBUF
      DATCHK ;CHECK DATA
      DEC CTRA ;TESTED 1000 CHARACTERS
      BNE APBB ;BRANCH IF NOT
      SCOPE ;YES. SCOPE
;*****
AT141: 141 ;TEST NUMBER 141
        ATLAST ;ADDRESS OF NEXT TEST
        100. ;TEST ITERATION COUNT
        AQBA ;SCOPE ENTRY POINT
;*****
;TEST THAT WHEN RXCSR BIT 1 IS SET THAT THE OUTPUT DATA LINE
;IS PULLED TO A SPACE.
;*****
AQBA: MOV #BIT2, @TXCSR ;SET MAINTENANCE BIT IN TXCSR
      MOV #BIT1, @RXCSR
      MOV #377, @TXBUF ;LOAD BUFFER
      TSTB @RXCSR ;WAIT FOR RECEIVER
      BPL -4 ;TO RECEIVE CHARACTER
      CMP @RXBUF, #0 ;CHARACTER RECEIVED SHOULD BE 0
      BEQ .+4
      ERROR ;CHARACTER OTHER THAN 0
      SRESET ;ISSUE RESET
      SCOPE
;SUBROUTINE TO SET RXCSR DONE BIT.
STRXD: BIS #BIT2, @TXCSR ;SET MAINTENANCE BIT.
      CLR @TXBUF ;LOAD TXBUF.
      DELAY ;DELAY 200 MSECS.
      200.
      RTS %7 ;EXIT.
;SUBROUTINE TO CHECK THAT CTRA THROUGH CTRD CONTAIN DESCENDING VALUES.
CMPT: CMP CTRA, CTRE
      BLOS CMPTA
      CMP CTRA, CTRC

```

3031	014272	101420			BLOS	CMPTA
3032	014274	026767	164746	164752	CMP	CTRA,CTRD
3033	014302	101414			BLOS	CMPTA
3034	014304	026767	164740	164740	CMP	CTRB,CTRC
3035	014212	101410			BLOS	CMPTA
3036	014314	026767	164730	164732	CMP	CTRB,CTRD
3037	014322	101404			BLOS	CMPTA
3038	014324	026767	164722	164722	CMP	CTRC,CTRD
3039	014332	101002			BHI	CMPTB
3040	014334	062716	000002		ADD	#2,3%6
3041	014340	000207			RTS	%7

CMPTA:
CMPTB:

3043						*****		
3044						:PRG1 - TRANSMITTER SCOPE LOOP		
3045						*****		
3046	014342	104000				PRG1: TYPE		;TYPE PROGRAM TITLE.
3047	014344	015763				PITIT		
3048	014346	004567	166362			JSR 5,LINSEL		;GO GET LINE # FROM USER
3049	014352	004767	000302			JSR %7,SETPAR		;GO SET PARAMETERS.
3050	014356	104000				TYPE		;TYPE SELECT CHAR AND DELAY.
3051	014360	016215				SELCAD		
3052	014362	000000				HALT		;WAIT FOR USER.
3053	014364	116767	163200	000010	PRG1A:	MOVB SR,PRG1B		;DELAY COUNT TO PRG1B.
3054	014372	116777	163173	164506		MOVB SR+1,@TXBUF		;LOAD TXBUF.
3055	014400	104016				DELAY		;DELAY # OF MSECS. SET AT SR.
3056	014402	000000			PRG1B:	OPEN		
3057	014404	000767				BR PRG1A		;REPEAT.
3058						*****		
3059						:PRG2 - RECEIVER SCOPE LOOP.		
3060						*****		
3061	014406	104000				PRG2: TYPE		;TYPE PROGRAM TITLE.
3062	014410	016023				P2TIT		
3063	014412	004567	166316			JSR 5,LINSEL		;GO GET LINE # FROM USER
3064	014416	004767	000236			JSR %7,SETPAR		;GO SET PARAMETERS.
3065	014422	104000				TYPE		;TYPE SELECT CHAR AND DELAY.
3066	014424	016215				SELCAD		
3067	014426	000000				HALT		;WAIT FOR USER.
3068	014430	004767	000256		PRG2A:	JSR 7,STPPRB		;RELOAD PARAMETERS
3069	014434	052777	000004	164442		BIS #BIT2,@TXCSR		;SET MAINTENANCE BIT.
3070	014442	116767	163122	000010		MOVB SR,PRG2B		;DELAY COUNT TO PRG2B.
3071	014450	116777	163115	164430		MOVB SR+1,@TXBUF		;LOAD TXBUF.
3072	014456	104016				DELAY		;DELAY # OF MSECS. SET IN SR.
3073	014460	000000			PRG2B:	OPEN		
3074	014462	017700	164414			MOV @RXBUF,%0		;RXBUF CONTENTS TO R0.
3075	014466	000005				RESET		;DISPLAY CONTENTS OF RXBUF (IN R0).
3076	014470	000005				RESET		;BY ISSUING 5 RESET INSTRUCTIONS
3077	014472	000005				RESET		
3078	014474	000005				RESET		
3079	014476	000005				RESET		
3080	014500	000753				BR PRG2A		
3081						*****		
3082						:PRG3 - SINGLE CHARACTER MAINTENANCE MODE DATA TEST.		
3083						*****		
3084	014502	104000				PRG3: TYPE		;TYPE PROGRAM TITLE.
3085	014504	016524				P3TIT		
3086	014506	004567	166222			JSR 5,LINSEL		;GO GET LINE # FROM USER
3087	014512	004767	000142			JSR %7,SETPAR		;SET PARAMETERS.
3088	014516	104000				TYPE		;TYPE: SELECT CHARACTER.
3089	014520	016670				SELCAR		
3090	014522	000000				HALT		
3091	014524	116767	163040	164500	PRG3A:	MOVB SR,CBUBA		;MOVE DATA CHAR TO CBUBA.
3092	014532	004767	000040			JSR %7,MOUTIN		;GO OUTPUT, RECEIVE, AND CHECK DATA.
3093	014536	000772				BR PRG3A		
3094						*****		
3095						:PRG4 - SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST.		
3096						*****		
3097	014540	104000				PRG4: TYPE		;TYPE PROGRAM TITLE.

3098	014542	016574			P4TIT		
3099	014544	004567	166164		JSR	5,LINSEL	GO GET LINE # FROM USER
3100	014550	004767	000104		JSR	%7,SETPAR	:SET PARAMETERS.
3101	014554	004767	166300		JSR	%7,INBIN	:INITIALIZE BINARY COUNT.
3102	014560	004767	166400		JSR	%7,GTBINP	:GET BINARY CHARACTER.
3103	014564	110167	164442		MOVB	%1,CRBUFA	:SAVE AT CRBUFA.
3104	014570	004767	000002		JSR	%7,MOUTIN	:GO OUTPUT, RECEIVE, AND CHECK DATA.
3105	014574	000771			BR	PRG4A	:REPEAT.
3106							
3107	014576	032767	000400	162764	:SUBROUTINE TO	OUTPUT, RECEIVE,	AND CHECK DATA WITH MAINTENANCE BIT SET.
3108	014604	001001			MOUTIN: BIT	#BIT8,SR	:SEE IF BIT 8 IS SET.
3109	014606	104002			BNE	.-4	:BRANCH IF SET.
3110	014610	105777	164270		STALL		:SET. DO A RANDOM STALL.
3111	014614	100375			TSTB	@TXCSR	:WAIT FOR TX READY.
3112	014616	052777	000004	164260	BPL	.-4	
3113	014624	016777	164402	164254	BIS	#BIT2,@TXCSR	:SET MAINTENANCE BIT.
3114	014632	046767	164376	164372	MOV	CRBUFA,@TXBUF	:LOAD TXBUF.
					BIC	CARMSK,CRBUFA	:MASK OFF NON-EXPECTED BITS.

3115	014640	105777	164234	
3116	014644	100375		
3117	014646	017767	164230	164354
3118	014654	104004		
3119	014656	000207		

TSTB	DRXCSR
BPL	-4
MOV	DRXBUF,CRBUF
DATCHK	
RTS	%7

;WAIT FOR RECEIVER DONE BIT.
;MOVE CHAR IN RX BUFFER TO CRBUF.
;COMPARE EXPECTED AND RECEIVED DATA
;EXIT.

```

3120
3121
3122
3123 014660 104000
3124 014662 016060
3125 014664 000000
3126 014666 016767 162676 164372
3127 014674 004567 166332
3128 014700 001266
3129 014702 016665
3130 014704 000002
3131 014706 104000
3132 014710 016647
3133 014712 032767 000020 164346 STPARB:
3134 014720 001403
3135 014722 052777 000400 164154
3136 014730 032767 000010 164330
3137 014736 001406
3138 014740 052777 000020 164136
3139 014746 052777 000020 164124
3140 014754 032767 000004 164304
3141 014762 001406
3142 014764 052777 000010 164112
3143 014772 052777 000010 164100
3144 015000 012767 177400 164226
3145 015006 032767 000002 164252
3146 015014 001421
3147 015016 012767 177700 164210
3148 015024 052777 002000 164046
3149 015032 032767 000001 164226
3150 015040 001406
3151 015042 012767 177740 164164
3152 015050 052777 001000 164022
3153 015056 000207
3154 015060 032767 000001 164200 STPARA:
3155 015066 001773
3156 015070 012767 177600 164136
3157 015076 000764
3158
3159
3160 015100 000300
3161 015102 000310
3162 015104 000320
3163 015106 000330
3164 015110 000340
3165 015112 000350
3166 015114 000360
3167 015116 000370
3168 015120 000400
3169 015122 000410
3170 015124 000420
3171 015126 000430
3172 015130 000440
3173 015132 000450
3174 015134 000460
3175 015136 000470

```

```

; SUBROUTINE TO SET STOP CODE, SPEED, AND CHARACTER LENGTH PARAMETERS SET
; IN SR INTO TXCSR AND RXCSR.

```

```

SETPAR: TYPE ;TYPE: SELECT PARAMETERS.
SELPAR
MALT ;WAIT FOR USER.
MOV SR, SRT ;SR TO SRT.
JSR 5, OACNV
SRT
APARM
2
TYPE
PARMTS
BIT #BIT4, SRT ;SEE IF SR BIT 4 IS SET.
BEQ .+10 ;BRANCH IF NOT SET.
BIS #BIT8, TXCSR ;SET. SET STOP CODE TO A 1.
BIT #BIT3, SRT ;SEE IF SR BIT 3 IS SET.
BEQ .+16 ;BRANCH IF NOT SET.
BIS #BIT4, TXCSR ;SET BIT4 IN TXCSR AND RXCSR
BIS #BIT4, RXCSR ;(MSB OF SPEED SELECT BITS.)
BIT #BIT2, SRT ;SEE IF SR BIT 2 IS SET.
BEQ .+16 ;BRANCH IF NOT SET.
BIS #BIT3, TXCSR ;SET BIT3 IN TXCSR AND RXCSR
BIS #BIT3, RXCSR ;(LSB OF SPEED SELECT BITS.)
MOV #177400, CARMSK ;SET CHARACTER MASK TO 8 BITS.
BIT #BIT1, SRT ;SEE IF SR BIT 1 IS SET.
BEQ STPARA ;BRANCH IF NOT SET.
MOV #177700, CARMSK ;CHANGE CHAR MASK TO 6 BITS.
BIS #BIT10, RXCSR ;SET RXCSR BIT 10(MSB OF CHAR LENGTH BITS.)
BIT #BIT0, SRT ;SEE IF SR BIT0 IS SET.
BEQ .+16 ;BRANCH IF NOT SET.
MOV #177740, CARMSK ;CHANGE CHAR MASK TO 5 BITS.
BIS #BIT9, RXCSR ;SET RXCSR BIT9 (LSB OF CHAR LENGTH BITS.)
RTS %7 ;EXIT.
BIT #BIT0, SRT ;SEE IF SR BIT0 IS SET.
BEQ STPARA-2 ;BRANCH IF NOT SET.
MOV #177600, CARMSK ;CHANGE CHAR MASK TO 7 BITS.
BR STPARA-10

```

; VECTOR ASSIGNMENT TABLE

VECTAB:	VECTOR	LINE	VECTOR
	300	0	VECTOR
	310	1	VECTOR
	320	2	"
	330	3	"
	340	4	"
	350	5	"
	360	6	"
	370	7	"
	400	10	"
	410	11	"
	420	12	"
	430	13	"
	440	14	"
	450	15	"
	460	16	"
	470	17	"

3176	015140	000500	500	..	20	..
3177	015142	000510	510	..	21	..
3178	015144	000520	520	..	22	..
3179	015146	000530	530	..	23	..
3180	015150	000540	540	..	24	..
3181	015152	000550	550	..	25	..
3182	015154	000560	560	..	26	..
3183	015156	000570	570	..	27	..
3184	015160	000600	600	..	30	..
3185	015162	000610	610	..	31	..
3186	015164	000620	620	..	32	..
3187	015166	000630	630	..	33	..
3188	015170	000640	640	..	34	..
3189	015172	000650	650	..	35	..
3190	015174	000660	660	..	36	..
3191	015176	000670	670	..	37	..
3192			

3193										
3194										.ASCII MESSAGES
3195	015200	050045	040							EMO: .ASCII '%P'
3196	015203	040	020040	052040						APNUMB: .ASCII ' . T '
3197	015210	040								
3198	015211	040	020040	050040						ATNUMB: .ASCII ' PC '
3199	015216	020103								
3200	015220	020040	020040	020040						APC: .ASCII ' @'
3201	015226	020040	100							
3202	015231	045	050045	043522						POTIT: .ASCII '%XPRGO - INPUT-OUTPUT LOGIC TESTS.'
3203	015236	020060	020055	047111						
3204	015244	052520	026524	052517						
3205	015252	050124	052125	046040						
3206	015260	043517	041511	052040						
3207	015266	051505	051524	020056						
3208	015274	044504	041523	047117						.ASCII 'DISCONNECT DC11 FROM MODEM.%'
3209	015302	042516	052103	042040						
3210	015310	030503	020061	051106						
3211	015316	046517	046440	042117						
3212	015324	046505	022456							
3213	015330	047101	020104	047503						.ASCII 'AND CONNECT JUMPER TO CABLE.%'
3214	015336	047116	041505	020124						
3215	015344	052512	050115	051105						
3216	015352	052040	020117	040503						
3217	015360	046102	027105	040045						
3218	015366	054124	051503	020122						ATXCSR: .ASCII 'TXCSR S/B: '
3219	015374	027523	035102	040						
3220	015401	040	020040	020040						ATXSB: .ASCII ' WAS: '
3221	015406	020040	053440	051501						
3222	015414	020072								
3223	015416	020040	020040	020040						ATXWAS: .ASCII ' @'
3224	015424	100								
3225	015425	122	041530	051123						ARXCSR: .ASCII 'RXCSR S/B: '
3226	015432	051440	041057	020072						
3227	015440	020040	020040	020040						ARXSB: .ASCII ' WAS: '
3228	015446	020040	040527	035123						
3229	015454	040								
3230	015455	040	020040	020040						ARXWAS: .ASCII ' @'
3231	015462	040040								
3232	015464	054124	051440	042520						ETXTIM: .ASCII 'TX SPEEDS NOT IN ASCENDING ORDER.@'
3233	015472	042105	020123	047516						
3234	015500	020124	047111	040440						
3235	015506	041523	047105	044504						
3236	015514	043516	047440	042122						
3237	015522	051105	040056							
3238	015526	044524	042515	043040						ESTPCD: .ASCII 'TIME FOR 2 STOP CODE OP LESS THAN FOR 1.@'
3239	015534	051117	031040	051440						
3240	015542	047524	020120	047503						
3241	015550	042504	047440	020120						
3242	015556	042514	051523	052040						
3243	015564	040510	020116	047506						
3244	015572	020122	027061	100						
3245	015577	124	020130	044103						ETCLGT: .ASCII 'TX CHAR LENGTHS NOT IN DESCENDING ORDER.@'
3246	015604	051101	046040	047105						
3247	015612	052107	051510	047040						
3248	015620	052117	044440	020116						

3249	015626	042504	041523	047105	
3250	015634	044504	043516	047440	
3251	015642	042122	051105	040056	
3252	015650	054122	051440	042520	ERXTIM: .ASCII 'RX SPEEDS NOT IN ASCENDING ORDER.␣'
3253	015656	042105	020123	047516	
3254	015664	020124	047111	040440	
3255	015672	041523	047105	044504	
3256	015700	043516	047440	042122	
3257	015706	051105	040056		
3258	015712	054122	041440	040510	ERCLGT: .ASCII 'RX CHAR LENGTHS NOT IN DESCENDING ORDER.␣'
3259	015720	020122	042514	043516	
3260	015726	044124	020123	047516	
3261	015734	020124	047111	042040	
3262	015742	051505	042503	042116	
3263	015750	047111	020107	051117	
3264	015756	042504	027122	100	
3265	015763	045	050045	043522	PITIT: .ASCII '%PRG1 - TRANSMITTER SCOPE LOOP␣'
3266	015770	020061	020055	051124	
3267	015776	047101	046523	052111	
3268	016004	042524	020122	041523	
3269	016012	050117	020105	047514	
3270	016020	050117	100		
3271	016023	045	050045	043522	P2TIT: .ASCII '%PRG2 - RECEIVER SCOPE LOOP␣'
3272	016030	020062	020055	042522	
3273	016036	042503	053111	051105	
3274	016044	051440	047503	042520	
3275	016052	046040	047517	040120	
3276	016060	051445	052105	050040	SELPAR: .ASCII '%SET PARAMETERS IN SR AS FOLLOWS:'
3277	016066	051101	046501	052105	
3278	016074	051105	020123	047111	
3279	016102	051440	020122	051501	
3280	016110	043040	046117	047514	
3281	016116	051527	072		
3282	016121	045	051123	020064	.ASCII '%SR4 = STOP CODE%SR3 AND 2 = SPEED'
3283	016126	020075	052123	050117	
3284	016134	041440	042117	022505	
3285	016142	051123	020063	047101	
3286	016150	020104	020062	020075	
3287	016156	050123	042505	104	
3288	016163	045	051123	020061	.ASCII '%SR1 AND 0 = CHAR LENGTH%␣'
3289	016170	047101	020104	020060	
3290	016176	020075	044103	051101	
3291	016204	046040	047105	052107	
3292	016212	022510	100		
3293	016215	045	042523	020124	SELCAD: .ASCII '%SET TEST CHAR CODE IN SR15-SR8, SET DELAY TIME IN SR7-SR0.␣'
3294	016222	042524	052123	041440	
3295	016230	040510	020122	047503	
3296	016236	042504	044440	020116	
3297	016244	051123	032461	051455	
3298	016252	034122	020054	042523	
3299	016260	020124	042504	040514	
3300	016266	020131	044524	042515	
3301	016274	044440	020116	051123	
3302	016302	026467	051123	027060	
3303	016310	100			
3304	016311	040	042040	052101	ERDAT: .ASCII ' DATA ERR S/B: '

.MAIN. MACY11 27(732) 29-SEP-76 13:35 PAGE 74
 DZDCAB.P11

3305	016316	020101	051105	020122	
3306	016324	051440	041057	020072	
3307	016332	020040	020040	053440	AASB: .ASCII ' WAS: '
3308	016340	051501	020072		
3309	016344	020040	040040		AWAS: .ASCII ' @'
3310	016350	037445	100		AINPRG: .ASCII '%?@'
3311	016353	045	042523	020124	ASETSR: .ASCII '%SET DESIRED SR OPTIONS. NORMAL OPERATION '
3312	016360	042504	044523	042522	
3313	016366	020104	051123	047440	
3314	016374	052120	047511	051516	
3315	016402	020056	047516	046522	
3316	016410	046101	047440	042520	
3317	016416	040522	044524	047117	
3318	016424	040			
3319	016425	111	020123	044527	.ASCII 'IS WITH SR = 000000@'
3320	016432	044124	051440	020122	
3321	016440	020075	030060	030060	
3322	016446	030060	100		
3323	016451	045	047111	047503	AINCRT: .ASCII '%INCORRECT ROUTINE SELECTED.@'
3324	016456	051122	041505	020124	
3325	016464	047522	052125	047111	
3326	016472	020105	042523	042514	
3327	016500	052103	042105	040056	
3328	016506	050045	047522	051107	APCEND: .ASCII '%PROGRAM END.@'
3329	016514	046501	042440	042116	
3330	016522	040056			
3331	016524	022445	051120	031507	P3TIT: .ASCII '%%PRG3-SINGLE CHAR MAINT MODE DATA TEST@'
3332	016532	051455	047111	046107	
3333	016540	020105	044103	051101	
3334	016546	046440	044501	052116	
3335	016554	046440	042117	020105	
3336	016562	040504	040524	052040	
3337	016570	051505	040124		
3338	016574	022445	051120	032107	P4TIT: .ASCII '%%PRG4-SPEC BIN COUNT MAINT MODE DATA TEST@'
3339	016602	051455	042520	020103	
3340	016610	044502	020116	047503	
3341	016616	047125	020124	040515	
3342	016624	047111	020124	047515	
3343	016632	042504	042040	052101	
3344	016640	020101	042524	052123	
3345	016646	100			
3346	016647	045	040520	040522	PARMTS: .ASCII '%PARAMETERS = '
3347	016654	042515	042524	051522	
3348	016662	036440	040		
3349	016665	040	040040		APARM: .ASCII ' @'
3350	016670	051445	052105	052040	SELCAR: .ASCII '%SET TEST CHAR CODE IN SR7-SRD.@'
3351	016676	051505	020124	044103	
3352	016704	051101	041440	042117	
3353	016712	020105	047111	051440	
3354	016720	033522	051455	030122	
3355	016726	040056			
3356	016730	046045	040517	020104	LDLINE: .ASCII '%LOAD LINE NO. (8) INTO SR 3-7@'
3357	016736	044514	042516	047040	
3358	016744	027117	024040	024470	
3359	016752	044440	052116	020117	
3360	016760	051123	031440	033455	

3361	016766	100				
3362	016767	040	046040	047111	ALINE: .ASCII ' LINE NO.'	
3363	016774	020105	047516	056		
3364	017001	040	020040	040527	SELINE: .ASCII ' WAS SELECTED@'	
3365	017006	020123	042523	042514		
3366	017014	052103	042105	100		
3367	017021	000001			DEND: .END	

.MAIN. MACY11 27(732) 28-SEP-76 13:35 PAGE 77
 DZDCAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

AAA	003714	1156	1159#				
AAAA	006226	1666	1669#				
AAAB	006240	1670	1672#				
AAB	003726	1161#	1164				
AABA	012544	2547	2552#				
AABB	012570	2558#	2564	2567			
AABC	012600	2553	2560#				
AABD	012614	2561	2563#				
AABE	012620	2560	2565#				
AAE	003730	1159	1162#				
AASB	016332	740	3307#				
ABA	003746	1159	1172#				
ABAA	007222	1878	1881#				
ABAB	007314	1885	1898#				
ABB	003760	1174#	1177				
ABBA	012636	2573	2576#				
ABBB	012670	2582	2584#				
ABE	003762	1172	1175#				
ACA	004000	1183	1187#				
ACAA	007326	1903	1907#				
ACAB	007462	1920	1932#				
ACB	004012	1189#	1192				
ACBA	012716	2594	2599#				
ACBB	012754	2598	2607#				
ACBC	012756	2606	2608#				
ACE	004014	1187	1190#				
ADA	004032	1199	1202#				
ADAA	007502	1938	1941#				
ADAB	007516	1943	1945#				
ADB	004044	1204#	1207				
ADE	004046	1202	1205#				
ADTEMP	003444	1079	1098#				
AEA	004064	1213	1216#				
AEAA	007532	1951	1955#				
AEAB	007552	1959	1962#				
AEB	004100	1217	1220#				
AEC	004122	1222	1225#				
AED	004142	1219	1224	1227	1229#		
AFA	004164	1236	1239#				
AFAA	007564	1968	1972#				
AFAB	007612	1977	1980#				
AFB	004206	1241	1244#				
AFBA	006252	1683	1688#				
AFBB	006274	1690	1693#				
AFBC	006314	1692	1695	1697#			
AFC	004230	1246	1249#				
AFD	004250	1243	1248	1251	1253#		
AGA	004262	1259	1262#				
AGAA	007624	1985	1989#				
AGAB	007652	1994	1997#				
AGB	004276	1263	1266#				
AGBA	006330	1703	1708#				
AGBB	006360	1711	1714#				
AGBC	006410	1720	1723#				
AGBD	006424	1725	1729#				
AGBE	006454	1713	1722	1727	1735	1737	1738#

.MAIN. MACY11 27(732) 28-SEP-76 13:35 PAGE 78
 DZDCAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

AGC	004320	1268	1271#					
AGD	004340	1265	1270	1273	1275#			
AHA	004362	1282	1285#					
AHAA	007664	2002	2006#					
AHAB	007712	2011	2014#					
AHB	004376	1286	1289#					
AHC	004420	1291	1294#					
AHD	004440	1288	1293	1296	1298#			
AIA	004462	1306	1309#					
AIAA	007724	2020	2024#					
AIAF	010064	2042	2045#					
AIAS	010066	2029	2032	2036	2039	2046#		
AIASA	010110	2052#	2056					
AIAST	010130	2030	2033	2037	2040	2046*	2054*	2058#
AIB	004476	1310	1313#					
AIC	004520	1315	1318#					
AID	004540	1312	1317	1320	1322#			
AINCRT	016451	715	3323#					
AINPRG	016350	3310#						
AJA	004562	1329	1332#					
AJAA	010142	2063	2068#					
AJAB	010226	2077	2081#					
AJAS	010230	2071	2074	2082#				
AJASA	010252	2088#	2092					
AJASB	010274	2094#	2098					
AJAST	010314	2072	2075	2082*	2090*	2096*	2100#	
AJB	004604	1334	1337#					
AJBA	006604	1772	1777#					
AJBB	006620	1782#						
AJBC	006642	1784	1787#					
AJBD	006662	1786	1789	1791#				
AJC	004626	1339	1342#					
AJD	004646	1336	1341	1344	1346#			
AKA	004670	1355	1359#					
AKAA	010326	2106	2111#					
AKAB	010474	2130	2133#					
AKAS	010476	2110	2120	2124	2127	2134#		
AKASA	010520	2140#	2144					
AKAST	010540	2117	2121	2125	2128	2134*	2142*	2146#
AKB	004700	1360	1362#					
AKBA	006676	1797	1803#					
AKBB	006704	1804#						
AKBC	006734	1807	1810#					
AKBD	006750	1811	1814#					
AKBE	007000	1809	1813	1818	1821#			
ALA	004712	1367	1370#					
ALAA	010552	2152	2157#					
ALAB	010602	2162	2165#					
ALAC	010614	2164	2167	2169#				
ALB	004726	1371	1374#					
ALBA	007122	1853	1858#					
ALBB	007144	1860	1863#					
ALBC	007166	1865	1868#					
ALBD	007206	1862	1867	1870	1872#			
ALC	004750	1376	1379#					
ALD	004770	1373	1378	1381	1383#			

ALINE	016767	1015	3362#					
AMA	005012	1392	1395#					
AMAA	010626	2174	2178#					
AMAB	010640	2180#	2181					
AMAC	010662	2184	2186#					
AMB	005024	1396	1398#					
AMBA	006470	1745	1750#					
AMBB	006526	1754	1757#					
AMBC	006542	1758	1761#					
AMBU	006570	1756	1760	1764		1766#		
ANA	005036	1404	1407#					
ANAA	010674	2192	2197#					
ANAB	010744	2205	2207#					
ANB	005060	1409	1412#					
ANBA	012770	2619#						
ANBB	013002	2616	2621#					
ANBC	013010	2622#	2651					
ANBD	013052	2631#	2635					
ANBE	013106	2637	2641#					
ANBF	013112	2639	2643#					
ANBG	013132	2640	2642	2650#				
ANC	005076	1414	1417#					
AND	005116	1411	1416	1419		1421#		
ANW	005130	1426	1430#					
ANX	005144	1431	1434#					
PNY	005166	1436	1439#					
ANZ	005206	1433	1438	1441		1443#		
AOAA	010762	2214	2219#					
AOAB	011032	2227	2229#					
AOBA	007014	1827	1831#					
AOBB	007036	1833	1836#					
AOBC	007066	1839	1842#					
AOBD	007106	1835	1841	1844		1846#		
APA	005230	1453	1456#					
APAA	011050	2235	2240#					
APAB	011104	2246	2248#					
APARM	016665	3129	3349#					
APB	005242	1457	1459#					
APBA	014072	2979	2988#					
APBB	014100	2985#	3000					
APC	015220	761	3200#					
APGEND	016506	719	3328#					
APNUMB	015203	765	3196#					
AQA	005254	1464	1467#					
AQAA	011122	2255	2260#					
AQAB	011342	2286	2289#					
AQAS	011344	2267	2273	2279		2283	2290#	
AQASA	011372	2297#	2301					
AQAST	011412	2268	2274	2280		2284	2290*	2299* 2303#
AQB	005270	1468	1471#					
AQBA	014166	3006	3011#					
AQC	005312	1473	1476#					
AQD	005332	1470	1475	1478		1480#		
ARA	005354	1487	1490#					
ARAA	011424	2309	2314#					
ARAB	011572	2333	2336#					

ARAS	011574	2319	2323	2327	2330	2337#		
ARASA	011630	2346#	2350					
ARAST	011650	2320	2324	2328	2331	2337*	2348*	2352#
ARB	005370	1491	1494#					
ARBA	005454	1511	1516#					
ARBB	005466	1517	1519#					
ARC	005412	1496	1499#					
ARD	005432	1493	1498	1501	1503#			
ARXCSR	015425	1931	3225#					
ARXSB	015440	1924	3227#					
ARXWAS	015455	1928	3230#					
ASA	005500	1525	1528#					
ASAA	011662	2358	2361#					
ASAB	011714	2365	2368#					
ASAC	011726	2369	2373#					
ASAD	011740	2367	2372	2375	2378#			
ASAS	011746	2361	2362	2380#				
ASB	005522	1530	1533#					
ASC	005544	1535	1538#					
ASD	005564	1532	1537	1540	1542#			
ASETSR	016353	711	3311#					
ATA	005606	1549	1552#					
ATAA	012006	2390	2397#					
ATAB	012032	2402#	2404					
ATAC	012034	2396	2403#					
ATB	005620	1553	1555#					
ATLAST=	177777	840#	3004					
ATNUMB	015211	769	3198#					
ATXCSR	015366	1897	3218#					
ATXSB	015401	1890	3220#					
ATXWAS	015416	1894	3223#					
ATO	003704	1144	1153#					
AT1	003736	1154	1166#					
AT10	004452	1280	1303#					
AT100	013142	2614	2656#					
AT101	013160	2657	2666#					
AT102	013176	2667	2676#					
AT103	013214	2677	2686#					
AT104	013232	2687	2696#					
AT105	013250	2697	2706#					
AT106	013266	2707	2716#					
AT107	013304	2717	2726#					
AT11	004552	1304	1326#					
AT110	013322	2727	2736#					
AT111	013340	2737	2746#					
AT112	013356	2747	2756#					
AT113	013374	2757	2766#					
AT114	013412	2767	2776#					
AT115	013430	2777	2786#					
AT116	013446	2777	2796#					
AT117	013464	2777	2806#					
AT12	004660	1327	1352#					
AT120	013502	2807	2816#					
AT121	013520	2817	2826#					
AT122	013536	2827	2836#					
AT123	013554	2837	2846#					

.MAIN. MACY11 27(732) 28-SEP-76 13:35 PAGE 81
 DZDCAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

AT124	013572	2847	2856
AT125	013610	2857	2866
AT126	013626	2867	2876
AT127	013644	2877	2886
AT13	004702	1353	1364
AT130	013662	2887	2896
AT131	013700	2897	2906
AT132	013716	2907	2916
AT133	013734	2917	2926
AT134	013752	2927	2936
AT135	013770	2937	2946
AT136	014006	2947	2956
AT137	014024	2957	2966
AT14	005002	1365	1389
AT140	014042	2967	2976
AT141	014156	2977	3003
AT15	005026	1390	1401
AT16	005120	1402	1423
AT17	005220	1424	1450
AT2	003770	1167	1180
AT20	005244	1451	1461
AT21	005344	1462	1484
AT22	005444	1485	1508
AT23	005470	1509	1522
AT24	005576	1523	1546
AT25	005622	1547	1558
AT26	005722	1559	1581
AT27	006022	1582	1604
AT3	004022	1181	1196
AT30	006122	1605	1627
AT31	006146	1628	1639
AT32	006172	1640	1652
AT33	006216	1653	1663
AT34	006242	1664	1680
AT35	006320	1681	1700
AT36	006460	1701	1742
AT37	006574	1743	1769
AT4	004054	1197	1210
AT40	006666	1770	1794
AT41	007004	1795	1824
AT42	007112	1825	1850
AT43	007212	1851	1875
AT44	007316	1876	1900
AT45	007472	1901	1935
AT46	007522	1936	1948
AT47	007554	1949	1965
AT5	004154	1211	1233
AT50	007614	1966	1982
AT51	007654	1983	1999
AT52	007714	2000	2017
AT53	010132	2018	2060
AT54	010316	2061	2103
AT55	010542	2104	2149
AT56	010616	2150	2171
AT57	010664	2172	2189
AT6	004252	1234	1256

.MAIN. MACY11 27(732) 29-SEP-76 13:35 PAGE 82
 DZDCAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

AT60	010752	2190	2211#		
AT61	011040	2212	2232#		
AT62	011112	2233	2252#		
AT63	011414	2253	2306#		
AT64	011652	2307	2355#		
AT65	011766	2356	2387#		
AT66	012040	2388	2408#		
AT67	012116	2409	2430#		
AT7	004352	1257	1279#		
AT70	012202	2431	2452#		
AT71	012300	2453	2478#		
AT72	012356	2479	2500#		
AT73	012440	2501	2521#		
AT74	012530	2522	2544#		
AT75	012626	2545	2570#		
AT76	012702	2571	2591#		
AT77	012760	2592	2613#		
AUA	005632	1561	1564#		
AUAA	012054	2411	2417#		
AUAB	012100	2423#			
AUAC	012106	2416	2424#	2427	
AUB	005646	1565	1568#		
AUC	005670	1570	1573#		
AUD	005710	1567	1572	1575	1577#
AVA	005732	1584	1587#		
AVAA	012132	2433	2439#		
AVAB	012170	2438	2446#		
AVAC	012172	2445	2447#		
AVB	005746	1588	1591#		
AVC	005770	1593	1596#		
AVD	006010	1590	1595	1598	1600#
AWA	006032	1607	1610#		
AWAA	012212	2455	2459#		
AWAB	012242	2466#	2472	2475	
AWAC	012252	2460	2468#		
AWAD	012266	2469	2471#		
AWAE	012272	2468	2473#		
AWAS	016344	736	2646	2649	3309#
AWB	006046	1611	1614#		
AWC	006070	1616	1619#		
AWD	006110	1613	1618	1621	1623#
AXA	006132	1630	1633#		
AXAA	012324	2481	2489#		
AXAB	012352	2487	2495#		
AXAC	012354	2494	2496#		
AXB	006144	1634	1636#		
AYA	006156	1642	1645#		
AYAA	012376	2503	2510#		
AYAB	012422	2514#	2518		
AYAC	012432	2508	2516#		
AYB	006170	1646	1648#		
AZA	006202	1655	1658#		
AZAA	012460	2524	2531#		
AZAB	012516	2529	2539#		
AZAC	012520	2537	2540#		
AZB	006214	1659	1661#		

RTNNO	001146	664#	768	784*	804	828*								
RXBUF	001102	646#	1118	1173	2182	2207	2229	2248	2295	2342	2378	2586	2629	2997
		3016	3074	3117										
RXBUFT	001262	706#												
RXCSR	001100	645#	995	1107*	1116	1160	1239*	1244*	1249*	1407*	1408	1413	1417*	1418
		1430	1434*	1435	1439*	1440	1443*	1456	1467	1471*	1472	1476*	1477	1480*
		1490	1494*	1495	1499*	1500	1503*	1516	1529	1533*	1534	1538*	1539	1542*
		1552	1564	1568*	1569	1573*	1574	1577*	1587	1591*	1592	1596*	1597	1600*
		1610	1614*	1615	1619*	1620	1623*	1633	1645	1658	1669	1688*	1689	1693*
		1694	1708*	1709	1710	1714*	1718	1723	1729*	1733	1750*	1751*	1752	1757
		1761*	1762	1777*	1782*	1787*	1806	1810	1817	1832	1838	1843	1858*	1863*
		1868*	1908*	1909*	1918	1932*	2115*	2118*	2119*	2122*	2123*	2126*	2161	2166
		2180	2183	2199*	2200*	2204	2221*	2222*	2226	2241*	2245	2266*	2271*	2272*
		2277*	2278*	2282*	2300	2318*	2321*	2322*	2325*	2326*	2329*	2349	2363	2373
		2489*	2491*	2510*	2512*	2514*	2531*	2534*	2540*	2554*	2555*	2558*	2580	2581
		2601*	2603*	2627	2636	2986*	2995	3012*	3014	3115	3139*	3143*	3148*	3152*
RXCSRT	001260	705#	1709*	1718*	1719	1733*	1734	1752*	1753	1762*	1763	1918*	1919	1927
		2363*	2364	2368	2580*									
RXLVL	001112	650#	878	2511	2532									
RXVTR	001110	649#	876	1004*	2560*									
SAVREG=	104013	635#	1067											
SAVRG	002140	684	847#											
SCOPE =	104012	634#	1123	1161	1174	1189	1204	1231	1253	1277	1300	1324	1348	1362
		1385	1398	1421	1445	1459	1482	1505	1519	1544	1555	1579	1602	1625
		1636	1648	1661	1672	1698	1739	1767	1792	1822	1847	1873	1898	1933
		1946	1962	1980	1997	2014	2045	2081	2133	2169	2186	2208	2230	2249
		2289	2336	2379	2402	2424	2448	2467	2496	2515	2541	2559	2587	2608
		2652	3001	3020										
SCOPTR	001154	667#	813	831*	834*									
SELCAD	016215	3051	3066	3293#										
SELCAR	016670	3089	3350#											
SELINE	017001	1012	3364#											
SELPAR	016060	3124	3276#											
SETPAR	014660	3049	3064	3087	3100	3123#								
SETSR	001272	710#	1149											
SPBOT =	001076	594#	781	795	1129									
SR =	177570	591#	720	726	755	785	798	802	811	815	821	991	3053	3054
		3070	3071	3091	3107	3126								
SRESET=	104011	633#	796	1230	1276	1299	1323	1347	1384	1412	1444	1481	1504	1543
		1578	1601	1624	1697	1738	1766	1791	1821	1846	1872	1883	1917	1945
		2165	3019											
SRSETT	002320	682	888#	890*										
SRT	001266	708#	1105*	3126*	3128	3133	3136	3140	3145	3149	3154			
STAL	002664	675	972#											
STALA	002704	975*	977#											
STALB	002706	974	978#											
STALL =	104002	626#	3109											
START	001562	643	781#											
STLMSK	001270	709#	973											
STLSPV	002270	680	881#											
STLSRV	002240	679	874#											
STPARA	015060	3146	3154#	3155	3157									
STPARB	014712	1106	1134	3068	3133#									
STPPA	002306	881*	884#											
STPRA	002256	874*	877#											
STRXD	014234	2488	2509	2530	2551	2576	2599	2600	3022#					

STRXV =	104006	630#	2486	2507	2528	2552	2597							
STTXV =	104007	631#	2395	2415	2437	2459								
SUBTEN	003376	1082	1086#											
SUBTNA	003402	1087#	1090											
SUBTNB	003416	1088	1091#											
SVRPC	002174	847*	855	857#										
SVRPSW	002176	848*	854	858#										
TEMP	001264	707#	993*	1007*	1008*	1009*	1011	1887*	1889	1921*	1923			
TENPWR	003442	1081*	1087	1091	1097#									
TKB	001122	654#												
TKLVL	001132	658#												
TKS	001120	653#												
TKVTR	001130	657#												
TPB	001126	656#	924*											
TPLVL	001136	660#												
TPS	001124	655#	925											
TPVTR	001134	659#												
TX =	000000	1151#	1184#											
TXBUF	001106	648#	1115*	1203	1911*	1914*	1941*	1955*	1973*	1990*	2007*	2051*	2087*	2093*
		2139*	2158*	2179*	2201*	2223*	2242*	2296*	2344*	2381*	2577*	2626*	2994*	3013*
		3023*	3054*	3071*	3113*									
TXCSR	001104	647#	1108*	1110	1188	1216	1220*	1221	1225*	1226	1229*	1240	1245	1250
		1262	1266*	1267	1271*	1272	1275*	1285	1289*	1290	1294*	1295	1298*	1309
		1313*	1314	1318*	1319	1322*	1333	1337*	1338	1342*	1343	1346*	1359	1370
		1374*	1375	1379*	1380	1383*	1395	1778*	1783	1788	1803*	1804*	1805*	1814*
		1815*	1831*	1836*	1837*	1842*	1859	1864	1869	1882*	1884	1886	1910*	1912
		1915	1942	1958	1972*	1976	1989*	1993	2006*	2010	2028*	2031*	2034*	2035*
		2038*	2047	2055	2070*	2073*	2083	2091	2097	2135	2143	2157*	2178*	2197*
		2198*	2219*	2220*	2240*	2264*	2265*	2269*	2270*	2275*	2276*	2281*	2291	2338
		2343*	2380*	2397*	2399*	2418*	2419*	2423*	2439*	2442*	2447*	2461*	2463*	2466*
		2620*	2624	2985*	2989	3011*	3022*	3069*	3110	3112*	3135*	3138*	3142*	
		704#	1886*	1893										
TXCSRT	001256	652#	885	2417	2440									
TXLVL	001116	651#	883	1006*	2468*									
TXVTR	001114	673	911#											
TYP	002410	914#	923	932										
TYPA	002420	916	918#											
TYPB	002436	922	924#	929	931									
TYPD	002464	914*	915	918	920	924	928*	930*	933#					
TYPDAT	002530	624#	710	714	718	941	988	1014	1145	3046	3050	3061	3065	3084
TYPE =	104000	3088	3097	3123	3131									
		625#	771											
TYPES =	104001	919	928#											
TYPF	002502	921	930#											
TYPG	002514	674	935#	943										
TYPS	002532	939	941#											
TYPSA	002556	937*	938	942#										
TYPSB	002560	1003	3160#											
VECTAB	015100	2653#	2655	2660#	2665	2670#	2675	2680#	2685	2690#	2695	2700#	2705	2710#
X =	000137	2715	2720#	2725	2730#	2735	2740#	2745	2750#	2755	2760#	2765	2770#	2775
		2780#	2785	2790#	2795	2800#	2805	2810#	2815	2820#	2825	2830#	2835	2840#
		2845	2850#	2855	2860#	2865	2870#	2875	2880#	2885	2890#	2895	2900#	2905
		2910#	2915	2920#	2925	2930#	2935	2940#	2945	2950#	2955	2960#	2965	2970#
Y	= 000040	2654#	2655	2664#	2665	2674#	2675	2684#	2685	2694#	2695	2704#	2705	2714#
		2715	2724#	2725	2734#	2735	2744#	2745	2754#	2755	2764#	2765	2774#	2775
		2784#	2785	2794#	2795	2804#	2805	2814#	2815	2824#	2825	2834#	2835	2844#

.MAIN. MACY11 27(732) 29-SEP-76 13:35 PAGE 92
DZDCAB.P11 CROSS REFERENCE TABLE -- MACRO NAMES

.SPOWE	1*
.SRAND	1*
.SRDDE	1*
.SRDOC	1*
.SREAD	1*
.SR2AZ	1*
.SSAVE	1*
.SSB2D	1*
.SSB20	1*
.SSCOP	1*
.SSIZE	1*
.SSUPR	1*
.STRAP	1*
.STYPB	1*
.STYPD	1*
.STYPE	1*
.STYPO	1*
.S4OCA	1*
.117D	1*

.MAIN. MACY11 27(732) 29-SEP-76 13:35 PAGE 96
DZDCAB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DZDCAB, DZDCZB, SEQ/SOL/CRF/PAGNUM/NL: TOC=SYSMAC.SML(400, 1066), DZDCAB(400, 4571)
RUN-TIME: 30 43 5 SECONDS
RUN-TIME RATIO: 314/80=3.9
CORE USED: 34K (67 PAGES)

