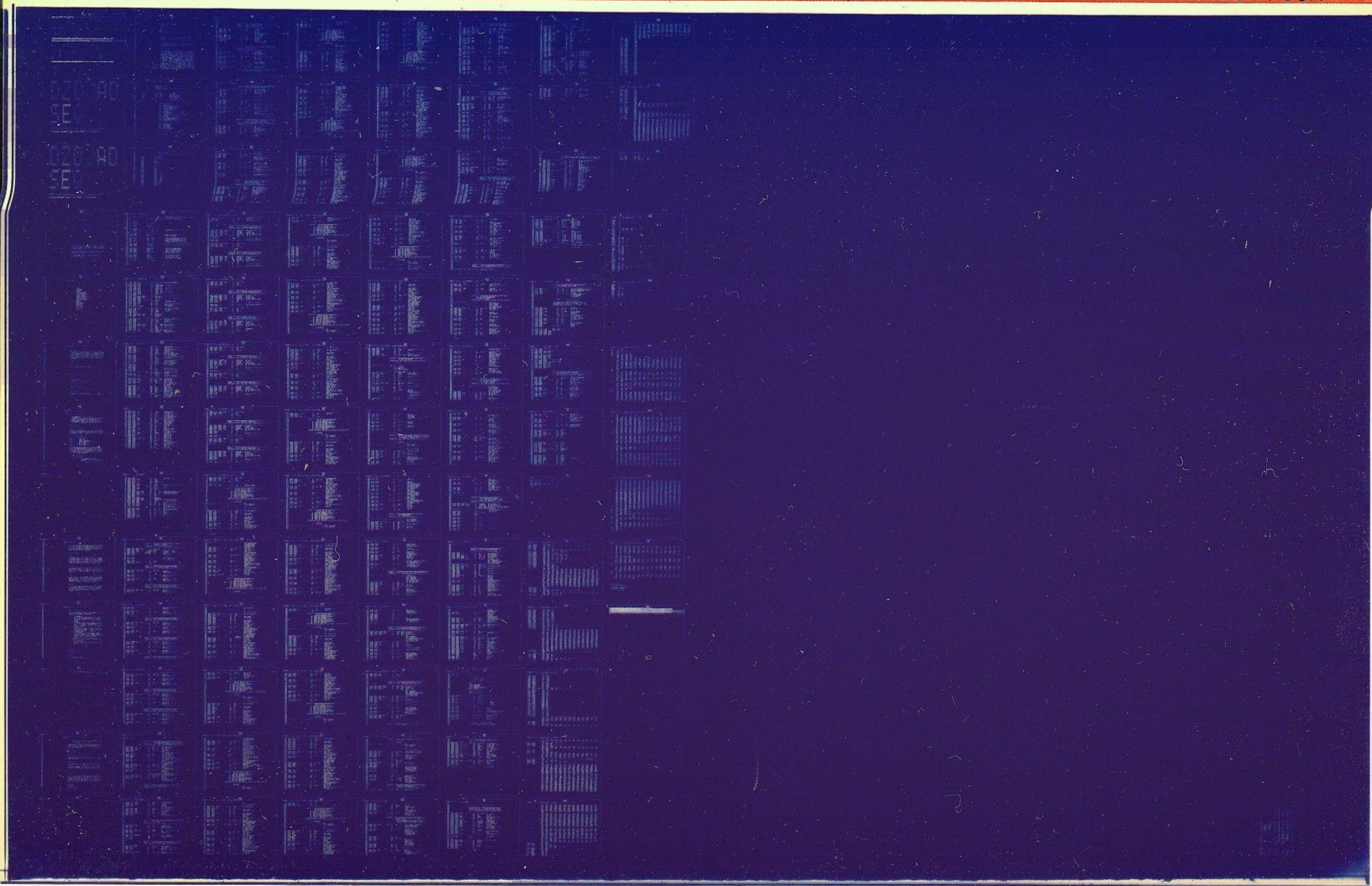


DJ11

LOGIC TEST
MD-11-DZDJA-D

EP-DZDJA-D-DL-A
COPYRIGHT©1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA



SECRET

PTC/NL2-1001233- Distribution to WH-4, also 104

[illegible]

J0867-90A229567E9-T23-5878-8Z089567898023456-906-7245679-018-958789012345-22-0123456-789C-22556789-0123-557897123456789

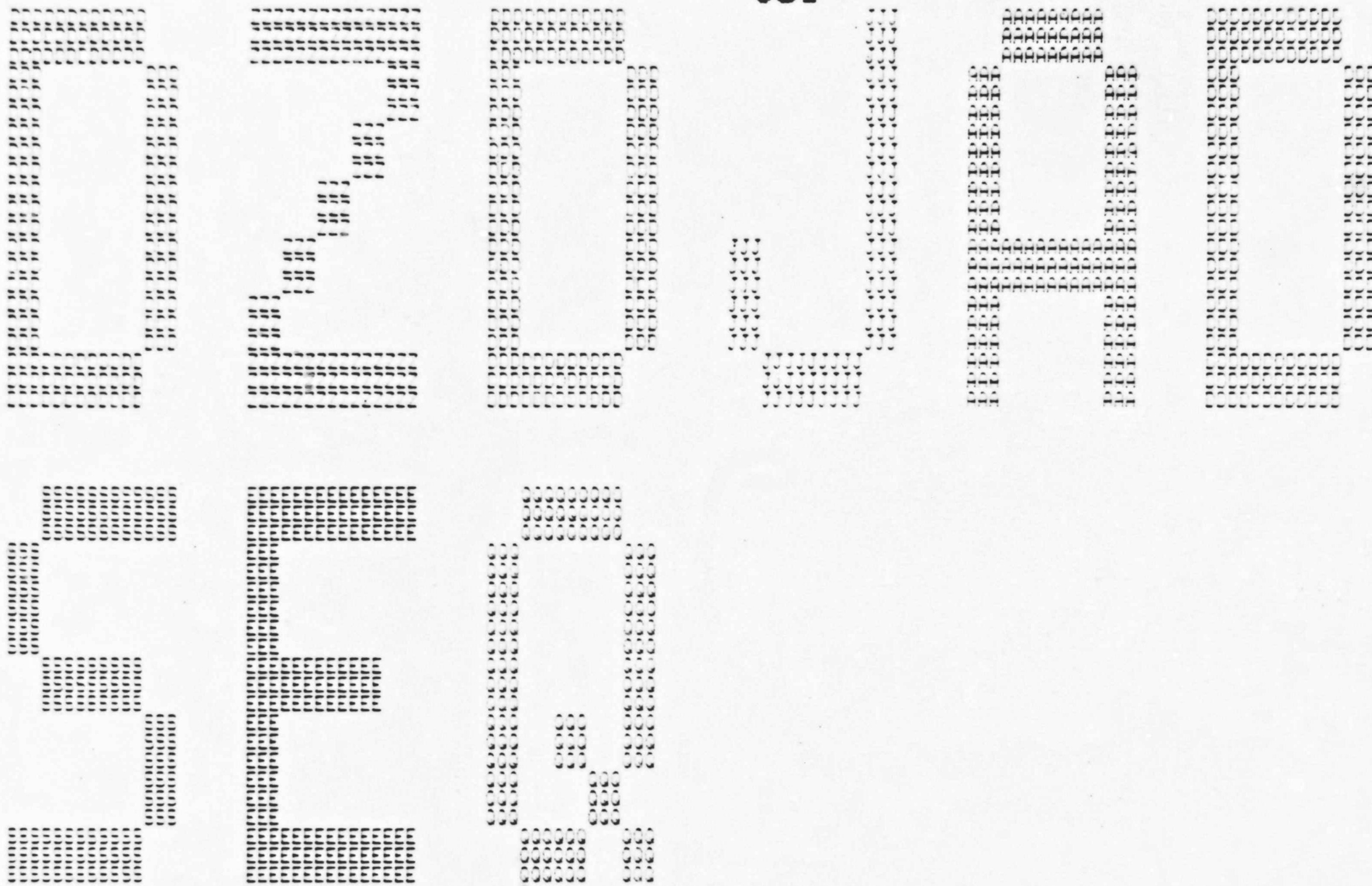
[illegible]

XTOMLBI--HQAIEB-- distribution to MLBI-- slot 134

SECRET

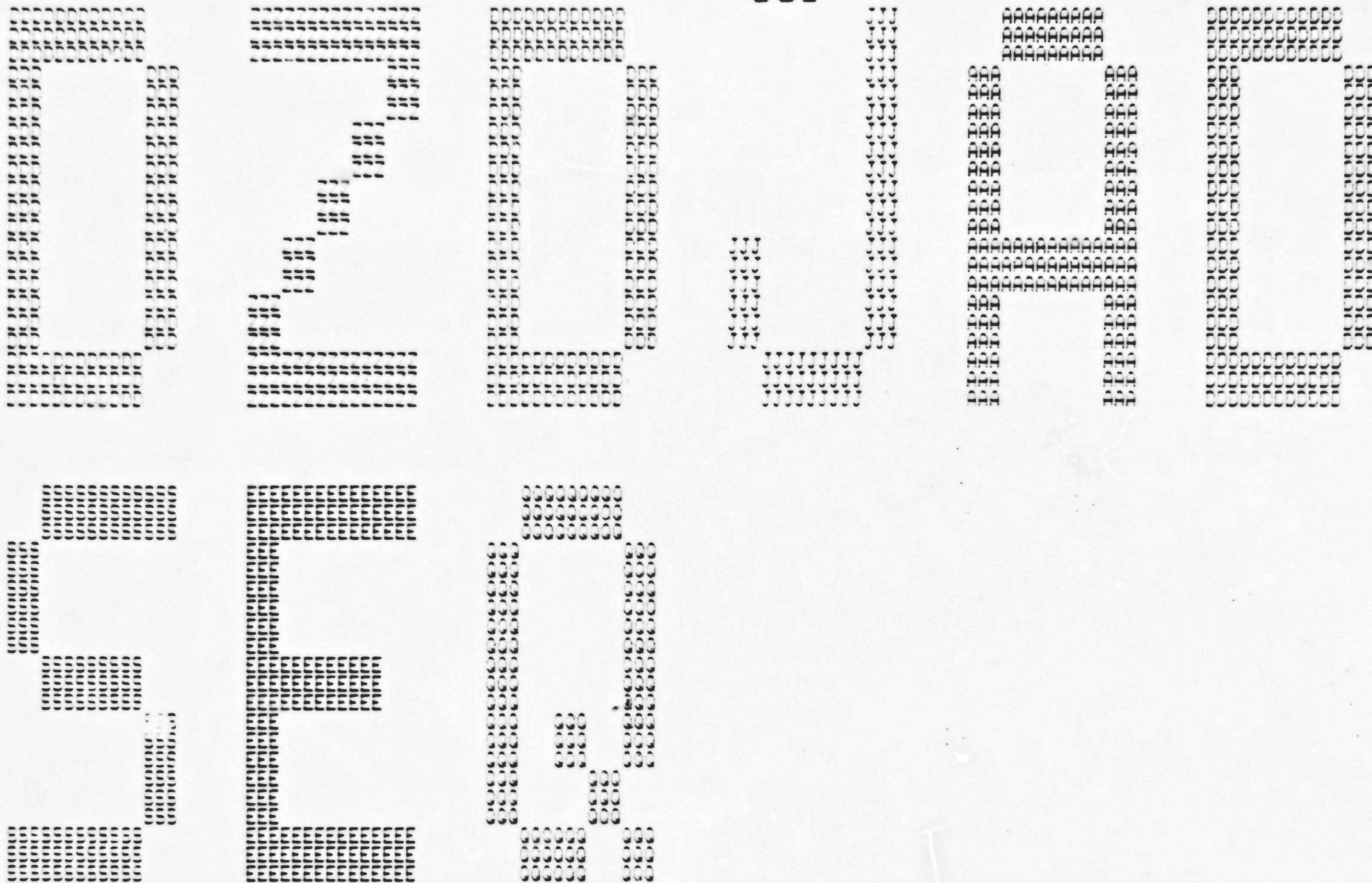
①

C01



LFTR Version 6/100344) Running on MTA1
+CART+ User: DAVID TOM (400 2704) Job 020JAD Sep. 23 Date: 13-Oct-76 15:13:22 Monitor IPC-D 5078 (1A3) *START*
Request created: 13-Oct-76 14:51:19 TO:ML21-4:DAVIES -- Distribution to ML21-4, slot 134
E:10: 55420:022:025:501:08:270:4:ICE:0101 1100:157: 15529100:11221M17:481:450335003476 15:13:26
File will be REFORMED to OS7 protection

D01



LF 1.00, Version 6 (100344) Running on MTA1

START User: DAVIES, TOM (400,2704) Job: DZDJAD Seq. 23 Date: 13-Oct-76 15:13:22 Monitor IPC-D 5078 (1A3) *START*

Request created: 13-Oct-76 14:51:19 /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

E:101 554ZC:32EJAC:SIOT:00:2204:ICE:1111 1:00:11575 16F29100:11321MIW:481.4F0335063F76L15:13:27

File will be RENAMED to (057) protection

E01

MAINDEC-11-DZDJA-D
21-SEP-76

DJ11 LOGIC TESTS

MAY11 27(732) 21-SEP-76 13:43 PAGE 2

.REN :

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDJA-D-D
PRODUCT NAME: DJ11 LOGIC TESTS
PROGRAM DATE: MAY 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 (C) 1972, 1976 BY DIGITAL EQUIPMENT CORPORATION

MAINDOC-11-0203A-0
0203A.F11

NACV11 27(732) 21-SEP-76 13:43 PAGE 3

DJ11 LOGIC TESTS

PAGE 2

[illegible]

1	ABSTRACT
2	REQUIREMENTS
3	EQUIPMENT
4	STORAGE
5	PRELIMINARY PROGRAMS
6	LOADING PROCEDURE
7	STARTING PROCEDURE
8	CONTROL SWITCH SETTINGS
9	STARTING ADDRESS
10	PROGRAM AND OPERATOR ACTION
11	OPERATING PROCEDURE
12	OPERATIONAL SWITCH SETTINGS
13	SUBROUTINE ABSTRACTS
14	PROGRAM AND OPERATOR ACTION
15	ERRORS
16	ERROR PRINTOUT
17	ERROR RECOVERY
18	ERROR COUNTER
19	RESTRICTIONS
20	MISCELLANEOUS
21	EXECUTION TIME
22	STACK POINTER
23	PASS COUNTER
24	POWER FAIL
25	PROGRAM DESCRIPTION

MAINDEC-11-DZDJA-D-D
DESCRIPTION

DJ11 LOGIC TESTS

PAGE 3

1. ABSTRACT

THIS PROGRAM TESTS THE LOGIC OF THE DJ11 ASYNCHRONOUS MULTIPLEXER IN MAINTENANCE MODE. IT CHECKS THAT ALL THE CONTROL REGISTERS FUNCTION PROPERLY, THAT INTERRUPTS OCCUR AT THE RIGHT LEVEL, AND THAT DATA CAN BE TRANSMITTED AND RECEIVED CORRECTLY. THIS PROGRAM DOES NOT TEST THAT THE INPUT AND OUTPUT LEAD CONNECTIONS ARE FUNCTIONAL. (SEE MAINDEC-11-DZDJB, PROGRAMS 2 AND 3 FOR ON-LINE TESTING). THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES DOWN.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER WITH CONSOLE TELETYPE UP TO 16 DJ11 ASYNCHRONOUS MULTIPLEXERS.

2.2 STORAGE

THIS PROGRAM USES ALL OF BK, EXCEPT ABS LOADER.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200. IT MAY BE RESTARTED AT 1000 AFTER ALL PARAMETERS HAVE BEEN SELECTED.

4.3 PROGRAM AND OPERATOR ACTION

- 1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 2) LOAD ADDRESS 200.
- 3) IF HARDWARE SWITCH REGISTER IS AVAILABLE, SET SWITCHES (SEE SEC. 5.1). ALL DOWN FOR WORST CASE, PRESS START.
- 4) IF SWITCH-LESS PROCESSOR SIMPLY PRESS START.
- 5) ENTER PARAMETERS (SEE SEC. 5.3) AS THEY ARE REQUESTED.
- 6) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS.
- 7) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200, ALL SWITCHES DOWN IS WORST CASE TESTING. EACH SUBTEST WILL BE LOOPED UPON UNTIL COMPLETION OF 16 PASSES OF THAT SUBTEST. THE BELL WILL RING UPON COMPLETION OF A PASS OF THE ENTIRE PROGRAM. ALTERNATE PASS WILL RUN WITH THE 1-BIT SET.

THE SWITCH SETTINGS ARE:

```

SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... SCOPE LOOP
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<12> = 1 ..... INHIBIT TRACE TRAPPING
SW<11> = 1 ..... INHIBIT ITERATIONS OF SUBTEST
SW<10> = 1 ..... BELL ON ERROR
          = 0 ..... BELL ON PASS COMPLETE
SW<09> = 1 ..... LOOP ON ERROR
SW<08> = 1 ..... LOOP ON TEST IN SW<7:0>

```

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(I.E.) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE.
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>.
3. 1U TO ALLOW REENTERING VALUE IF ERROR IS

I01

MAINDEX-11-DIDCA-D
DIDCA01.F11

DJ11 LOGIC TESTS

MACY11 27(732) 21-SEP-76 13:43 PAGE 6

192

COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL (VIA A TRAP INSTRUCTION) IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF "LAD" MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 FLT

THIS ROUTINE (CALLED BY AN EMT INSTRUCTION) PRINTS OUT AN ERROR MESSAGE (SEE 6.1). IF SW<9> IS ON A 1 AND A HLT IS EXECUTED, THE SUBTEST WILL BE LOOPED UPON UNTIL 16 CONSECUTIVE GOOD PASSES ARE COMPLETED. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1. TO RING THE BELL ON AN ERROR, PUT SW<10> ON A 1.

5.2.4 TRIRAP

IF SW<12> IS ON A 0, THE T-BIT WILL BE SET ON ALTERNATE PASSES. WHEN THE T-BIT IS SET, THE PROCESSOR TRAPS AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTI" OR "RTT" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS. THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS REACHED.

5.2.5 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 56 TO DETECT ANY UNEXPECTED TRAPS AND A ".+2" - "IOT" SEQUENCE IS REPEATED FROM 60 - 775 TO DETECT ANY UNEXPECTED INTERRUPTS. THUS ANY UNEXPECTED TRAPS WILL HALT AT THE VECTOR + 2. ANY UNEXPECTED INTERRUPTS WILL RESULT IN AN ERROR "HLT" IN "IOTRAP".

[illegible]

THE FOLLOWING REQUESTS ARE MADE TO THE OPERATOR AT THE BEGINNING OF THE PROGRAM. A DETAILED DESCRIPTION OF WHAT IS REQUIRED FOR EACH PARAMETER IS GIVEN BELLOW.

- 1) "FIRST DJ11 ADDRESS: "
THE CSR ADDRESS OF THE FIRST DJ11 YOU WISH TO TEST. MUST BE BETWEEN 160000(8) AND 177777(8). THE DEFAULT (CARRIAGE RETURN) IS TO 160010(8). "P(REVIOUS)" SELECTS THE ADDRESS PREVIOUSLY SELECTED.
- 2) "VECTOR ADDRESS: "
THE RECEIVER INTERRUPT VECTOR ADDRESS OF THE FIRST DJ11 YOU WISH TO TEST. MUST BE BETWEEN 300(8) AND 1000(8). THE DEFAULT (CARRIAGE RETURN) IS TO 300(8). "P(REVIOUS)" SELECTS THE ADDRESS PREVIOUSLY SELECTED.
- 3) "NO. OF DJ11'S: "
THE NUMBER OF DJ11 UNITS YOU WISH TO TEST AT ONE TIME. MUST BE BETWEEN 1 AND 16. THE DEFAULT (CARRIAGE RETURN) IS TO 1. "P(REVIOUS)" SELECTS THE NUMBER OF UNITS PREVIOUSLY SELECTED.
- 4) "STANDARD CONFIGURATION? "
"Y(ES)" OR DEFAULT (CARRIAGE RETURN) SELECTS 8 LEVEL CODE, NO PARITY. "N(O)" CAUSES REQUESTS FOR CODE LEVEL AND PARITY ON ALL REQUESTED LINES IN GROUPS OF FOUR. "P(REVIOUS)" SELECTS THE CODE LEVELS AND PARITIES PREVIOUSLY SELECTED.
- 5) "CHAR LENGTH: "
THE CODE LEVEL FOR THE LINE GROUP SPECIFIED. MUST BE 5, 6, 7, OR 8. THE DEFAULT (CARRIAGE RETURN) IS TO 8 LEVEL CODE.
- 6) "PARITY (NO, ODD, EVEN): "
THE TYPE OF PARITY SELECTED FOR THE LINE GROUP SPECIFIED. THE DEFAULT (CARRIAGE RETURN) IS TO NO PARITY.

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR DJADR (R1) (R2) (R3) (R4)

WHERE:

ADR = ADDRESS OF ERROR HLT
DJADR = CSR ADDRESS OF DJ11 UNDER TEST

L01

MAINDEC-11-DZDJA-D
DZDJAD.F11

DJ11 LOGIC TESTS

MACY11 27(732) 21-SEP-76 13:43 PAGE 9

305
306

(RN) = CONTENTS OF GENERAL REGISTER "N". FROM NONE TO
FOUR OF THESE MAY BE TYPED DEPENDING ON THE NUMBER

MAINDEC-11-DZDJA-D-D
DESCRIPTION

DJ11 LOGIC TESTS

PAGE 7

FOLLOWING THE HLT: E.G.. HLT+3 WOULD TYPE (R1)
THRU (R3); HLT (BY ITSELF) WOULD STOP AFTER TYPING
ADR AND DJADR.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE
ADDRESS TYPED. IN MOST CASES THE COMMENT BESIDE THE HLT
TELLS WHAT WAS BEING CHECKED AND WHAT WAS EXPECTED. ALSO, A
LIST OF THE PROBABLE FAILING LOGIC IS GIVEN IN THE COMMENTS
AT THE BEGINNING OF THE TEST.

6.2 ERROR RECOVERY

RESTART AT 200 OR 1000.

6.3 ERROR COUNTER

AN ERROR COUNT IS KEPT IN "ERRORS" (LOC 1202). IT CAN BE
CLEARED FROM THE CONSOLE, BY RESTARTING AT 200, OR BY
RELOADING THE PROGRAM.

7. RESTRICTIONS

IF MORE THAN ONE DJ11 IS TESTED AT A TIME, THE DEVICE
ADDRESSES AND THE VECTOR ADDRESSES MUST ALL BE CONTIGUOUS.

IF THIS PROGRAM IS RUN WITH A MONITOR, I.E. ACT11 OR DDP,
THE DEVICE ADDRESSES MUST FOLLOW THE FLOATING ADDRESS
CONVENTION. DJ11'S WILL BE FIRST, STARTING AT 160010,.

8. MISCELLANEOUS

8.1 EXECUTION TIME

DUE TO THE VARIOUS BAUD RATES AVAILABLE AND THE ABILITY TO
CHECK UP TO 8 DJ11'S AT ONCE, THE EXECUTION TIME CAN BE
ANYWHERE FROM 15 SECONDS TO THREE AND A HALF HOURS. THE
FOLLOWING TYPICAL TIMES ARE FOR ONE DJ11 WITH ALL LINES AT
THE SAME SPEED, 8 LEVEL CODE, 2 STOP BITS, AND NO PARITY ON
A PDP-11/20 WITHOUT TRACE TRAPPING. FOR MULTIPLE DJ11'S,
MULTIPLY THESE TIMES BY THE NUMBER OF UNITS SELECTED FOR
TEST.

BAUD RUN TIME

75 00:26:00
110 00:18:00
134.5 00:15:00

007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
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
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
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
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

NO1

MAINDEC-11-DZDJA-
DZDJAD.P11

DJ11 LOGIC TESTS

MACY11 27(732) 21-SEP-76 13:43 PAGE 11

363
364

150 00:13:00
300 00:05:30

1123

STACK POINTER

(11)

PASS COUNT

(1) 1.

POWER FAIL

9.

PROGRAM DESCRIPTION

1. The first step is to identify the problem. This involves understanding the symptoms and the context in which the problem is occurring.

2. Next, we need to gather information. This can be done through interviews, observations, and reviewing relevant documents.

3. Once we have gathered the information, we need to analyze it. This involves looking for patterns, identifying causes, and determining the scope of the problem.

4. After analysis, we can develop a plan of action. This plan should outline the steps that need to be taken to resolve the problem.

5. Finally, we need to implement the plan. This involves putting the plan into action and monitoring the results to ensure that the problem is resolved.

LOGIC TESTS
SWITCH SETTINGS

TITLE MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS
PROGRAM BY KEY CHAPMAN
COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

SWITCH	USE
00000000	HALT ON ERROR
00000001	LOOP ON TEST
00000010	INHIBIT ERROR TYPEOUTS
00000011	INHIBIT ERROR TRAP
00000100	INHIBIT INTERRUPTS
00000101	0 - WAIT ON PASS COMPLETE
00000110	1 - WAIT ON ERROR
00000111	LOOP ON ERROR
00001000	LOOP ON TEST IN SW 7:0

DJ11 REGISTER BIT ASSIGNMENTS:

CONTROL STATUS REGISTER (CSR) XXXXXX0

00000000	RECEIVER ENABLE (READ/WRITE)
00000001	RECEIVER DUBLEX SELECT (READ/WRITE)
00000010	MAINTENANCE (READ/WRITE)
00000011	CLEAR MOS (WRITE ONLY)
00000100	CLEAR MOS FLAG (READ ONLY)
00000101	NOT USED
00000110	RECEIVER INTERRUPT ENABLE (READ/WRITE)
00000111	DONE (READ ONLY)
00001000	MASTER TRANSMITTER SCAN ENABLE (READ/WRITE)
00001001	NOT USED
00001010	READ/WRITE BREAK REGISTER (READ/WRITE)
00001011	NOT USED
00001100	STATUS ENABLE (READ/WRITE)
00001101	TI/FO OVERRUN (READ ONLY)
00001110	MASTER TRANSMITTER INTERRUPT ENABLE (READ/WRITE)
00001111	TRANSMITTER READY (READ ONLY)

RECEIVER BUFFER REGISTER (RBUF) XXXXX2 (READ ONLY)

BIT0-7	RECEIVED CHARACTER
BIT8-11	LINE NUMBER
BIT12	PARITY ERROR
BIT13	FRAMING ERROR
BIT14	UART OVERRUN ERROR
BIT15	CHARACTER PRESENT

TRANSMITTER CONTROL REGISTER (TCR) XXXXX4 (READ/WRITE)

BIT0-15 STOP THE SCANNER ON CORRESPONDING LINE

TRANSMITTER BUFFER (TBUF) XXXXX6

BIT0-7 TRANSMITTED CHARACTER (WRITE ONLY)

BATS-11 LINE NUMBER (READ ONLY)

BREAK CONTROL STATUS REGISTER (BCSR) XXXXXX (BIT 0 OF CSR SET) (READ/WRITE)

BIT0-15 TRANSMIT A BREAK ON CORRESPONDING LINE!

[illegible][illegible][illegible]

MAX ALLOWED NUMBER OF CALLS


```

:TRAP CATCHER IN LOCATIONS 0 THRU 775
:LOCATIONS 0 AND 2 CONTAIN "HALT" INSTRUCTIONS
+2" AND "HALT" IN EVERY VECTOR
".+2" AND "IOT" IN EVERY VECTOR

:200 ALWAYS IS THE STARTING ADDRESS
:RESTART ADDRESS

:ITERATION COUNT-LH, TEST NO.-RH
:ERROR COUNT REGISTER
:PASS COUNT REGISTER

:CONTROL STATUS REGISTER(DJ UNDER TEST)
:RECEIVER BUFFER REGISTER(DJ UNDER TEST)
:TRANSMITTER CONTROL REGISTER(DJ UNDER TEST)
:BREAK STATUS REGISTER(DJ UNDER TEST)
:TRANSMITTER BUFFER REGISTER(DJ UNDER TEST)

:RECEIVER INTERRUPT VECTOR ADDRESS(DJ UNDER TEST)
:TRANSMITTER INTERRUPT VECTOR ADDRESS(DJ UNDER TEST)

:FIRST DEVICE ADDRESS(SELECTED)
:FIRST VECTOR ADDRESS(SELECTED)
:NUMBER OF UNITS TO BE TESTED
:TABLE OF CHARACTER LENGTHS (MASKS)
:TABLE OF ODD PARITY FLAGS
:USED TO SAVE RELATIVE TIMES

:UNIT NUMBER OF DJ11 UNDER TEST
:CHAR MASK TABLE POINTER
:UNIT FLAG (FOR ODD PARITY FLAG)
:DEFAULT FIRST DEVICE ADDRESS
:DEFAULT FIRST VECTOR ADDRESS

```

```

001324 012706 001200 BEGIN: MOV #ICNT, SP ;SET UP STACK POINTER
001330 004737 017710 JSR PC, SUSWR ;CHECK FOR SWITCH REGISTER
001334 012700 000014 MOV #14, RO
001340 012720 015534 MOV #YESRT, (RO)+ ;TRACE TRAP VECTOR (14)
001344 012720 000340 MOV #340, (RO)+
001350 012720 017104 MOV #IOTRAP, (RO)+ ;IOT VECTOR (20)
001354 012720 000340 MOV #340, (RO)+
001360 012720 016744 MOV #PDOWNS, (RO)+ ;POWER FAIL VECTOR (24)
001364 012720 000340 MOV #340, (RO)+
001370 012720 015774 MOV #EMTS, (RO)+ ;EMT VECTOR (30)
001374 012720 000340 MOV #340, (RO)+
001380 012720 015632 MOV #TRAPS, (RO)+ ;TRAP VECTOR (34)
001384 012720 000340 MOV #340, (RO)+
001390 012737 001446 MOV #15, 3#10
001394 006700 SXT RO ;CHECK FOR PDP-11/40 OR 45
001400 012737 000006 015534 MOV #RTT, 2#YESRT
001404 012737 000006 000006 MOV #RTT, 2#6
001408 012737 000400 177774 MOV #400, 3#177774 ;SET UP STACK LIMIT TO 1000
001412 005037 000006 CLR 3#6
001416 012737 000012 000010 1$: MOV #12, 3#10
001420 005037 001202 CLR ERRORS ;CLEAR ERROR COUNTER
001424 005037 001204 CLR PCNT ;CLEAR PASS COUNTER
001428 005037 001206 CLR PCNT+2
001432 005737 000042 TST 3#42 ;CHECK FOR ACT11 OR DDP PRESENT
001436 001404 BEQ GETADR ;BRANCH IF NONE
001440 004737 JSR PC, AUTO ;GO TO SUBROUTINE TO "MAP" DJ11 ON THE SYS
001444 000137 JMP RESTAR ;SKIP OPERATOR ACTION

001506 000004 017244 GETADR: TYPE, MSGADR ;TYPE "FIRST DJ11 ADDRESS"
001512 004537 016216 JSR R5, READIN ;READ INPUT FROM TTY AND SAVE
001516 001230 .WORD DEVADR ;IN DEVADR
001520 001372 BNE GETADR ;BRANCH IF BAD INPUT
001524 005737 001230 TST DEVADR
001528 001003 BNE 1$
001532 013737 001312 001230 MOV DEFADR, DEVADR
001536 042737 000007 001230 1$: BIC #7, DEVADR
001540 022737 160000 001230 CMP #160000, DEVADR
001544 101355 BHI GETADR

001554 000004 017274 GETVEC: TYPE, MSGVEC ;TYPE "VECTOR ADDRESS:"
001560 004537 016216 JSR R5, READIN ;READ INPUT FROM TTY AND SAVE
001564 001232 .WORD VECADR ;IN VECADR
001568 001372 BNE GETVEC ;BRANCH IF BAD INPUT
001572 005737 001232 TST VECADR ;CHECK FOR CR
001576 001003 BNE 1$
001580 012737 000300 001232 MOV #300, VECADR ;SET TO FIRST FLOATING VECTOR
001584 042737 000007 001232 1$: BIC #7, VECADR ;CLEAR TO MODULO 10
001588 022737 000300 001232 CMP #300, VECADR ;CHECK FOR LOW LIMIT
001592 003355 BGT GETVEC
001596 022737 001000 001232 CMP #1000, VECADR ;CHECK FOR UPPER LIMIT
001600 003751 BLE GETVEC
001604 000004 017320 GETNUM: TYPE, MSGNUM ;TYPE "NUMBER OR UNITS:"
001608 004737 016354 JSR PC, READS ;READ INPUT FROM THE TTY
; CHECK STRING FOR VALID DIGITS, TERMINATOR, AND 'P'
MOV #INPUT, R2 ;POINTS TO INPUT STRING

```


001646	112201		MOV	(R2)+,R1	:LOAD 1ST CHAR
001650	001434		BEQ	1\$	:BRANCH IF IMMEDIATE TERMINATOR
001652	122701	000120	CMPB	#P,R1	:CHECK FOR A P
001656	001443		BEQ	CONFIG	:CONFIGURE MODE IF SO
001660	120127	000071	CMPB	R1,#71	:MUST BE A VALID ASCII NUMBER
001664	101362		BHI	GETNUM	:ELSE RETRY
001666	162701	000060	SUB	#60,R1	:STRIP ASCII CODE
001672	003757		BLE	GETNUM	:SHOULDN'T BE ZERO OR NEG
				:1ST CHAR IS A VALID DIGIT, 1 THRU 9 - CHECK REST OF STRING	
001674	105722		TSTB	(R2)+	:2ND CHAR A TERMINATOR?
001676	001433		BEQ	3\$	:YES - R1 HAS THE FINAL # OF UNITS
001700	105712		TSTB	(R2)	:NO - NXT CHAR MUST BE THE TERMINATOR
001702	001353		BNE	GETNUM	:ELSE RETRY.
001704	124227	000071	CMPB	-(R2),#71	:CHECK 2ND CHAR FOR A VALID NUMBER
001710	101350		BHI	GETNUM	
001712	111303		MOV	(R2),R3	:MAKE IT MORE ACCESSIBLE
001714	162702	000060	SUB	#60,R3	:STRIP ASCII CODE
001720	100744		SMT	GETNUM	:SHOULDN'T BE NEGATIVE
				:BOTH LOW AND HIGH ORDER DIGITS ARE OK - CONVERT DECIMAL VALUE.	
001722	010146		MOV	R1,-(SP)	:SAVE IT FOR LATER
001724	006301		ASL	R1	:MULT HI ORDER BY 8
001726	006301		ASL	R1	
001730	006301		ASL	R1	
001732	006316		ASL	(SP)	:MULTIPLY IT BY 2
001734	062601		ADD	(SP)+,R1	:RESULT IS (HI ORD)*10.
001736	060301		ADD	R3,R1	:NOW ADD IN THE LOW ORDER
001740	000402		BR	3\$	
001742	012701	000001	1\$: MOV	#1,R1	:LOAD DEFAULT VALUE HERE
			3\$: R1 HAS THE CONVERTED VALUE -	COMPARE AGAINST MAX ALLOWED.	
001746	020127	000020	3\$: CMP	R1,#DJMXNO	:TOO BIG?
001750	101327		BHI	GETNUM	:YES - RETRY.
001754	010137	001234	MOV	R1,UNITS	:VALUE OK - STORE RESULT
001760	012737	000100 001304	MOV	#100,DJUUT	:PRIME UNIT UNDER TEST NUMBER
			CONFIG: TYPE,	MSGCON	:TYPE "STANDARD CONFIG?"
001766	000004	017343	JSR	PC	:READ INPUT FROM TTY
001772	004737	016364	CMPB	#P	:CHECK FOR "P"
001776	122737	000120 016512	BEQ	RESTAR	:BRANCH IF PREVIOUS
002004	001536		MOV	#22,R0	:SET UP COUNTER
002006	012700	000022	MOV	#LENGTH,R1	:POINT TO CHAR LEN TABLE
002012	012701	001236	1\$: CLR	(R1)+	:PUT CHAR MASK FOR 9 IN CHAR TABLE
002016	005021				:AND CLR PARITY TABLE
002020	005300		DEC	R0	:COUNT DOWN
002022	001375		BNE	1\$	:BRANCH IF NOT DONE
002024	105737	016512	TSTB	INPUT	:CHECK FOR CR
002030	001524		BEQ	RESTAR	:BRANCH IF DEFAULT
002032	122737	000131 016512	CMPB	#131	:CHECK FOR "Y"
002040	001520		BEQ	RESTAR	:BRANCH IF DEFAULT
002042	122737	000116 016512	CMPB	#116	:CHECK FOR "N"
002050	001346		BNE	CONFIG	:BRANCH IF ILLEGAL ENTRY
002052	005000		CLR	R0	:CLR UNIT COUNTER
002054	005001		CLR	R1	:CLR LINE COUNTER
002056	012702	001236	MOV	#LENGTH,R2	:SET UP POINTER TO CHAR MASK TABLE
002062	012703	001276	MOV	#PARITY,R3	:SET UP POINTER TO PARITY TABLE
002066	012704	000001	MOV	#1,R4	:SET UP MARKER
002072	000004	017377	TYPLIN: TYPE,	MSGLIN	:TYPE "LINES "

600	0000076	010105		MOV	R1,TTY	:TYPE R1 IN OCTAL
601	0000100	004737	016572	JSR	PC,PRINTS	:AND SUPPRESS LEADING ZERO'S
602	0000104	0000004	017410	TYPE,	MSGDAS	:TYPE A DASH (-)
603	0000110	062701	0000003	ADD	#3, R1	:LAST LINE IN GROUP OF 4
604	0000114	010105		MOV	R1,TTY	:TYPE R1 IN OCTAL
605	0000116	004737	016572	JSR	PC,PRINTS	:AND SUPPRESS LEADING ZERO'S
606	0000122	005201		INC	R1	:FIRST LINE NEXT GROUP
607	0000124	0000004	017414	GETLEN: TYPE,	MSGLEN	:TYPE "CHAR LENGTH:"
608	0000120	004737	016364	JSR	PC, READS	:READ FROM THE TTY
609	0000124	105737	016512	TST3	INPUT	:CHECK FOR CR (DEFAULT)
610	0000140	001421		BEQ	GETPAR	:BRANCH IF DEFAULT
611	0000142	105737	016513	TSTB	INPUT+1	:CHECK FOR BAD DATA
612	0000146	001366		BNE	GETLEN	:BRANCH IF BAD
613	0000150	162737	000060	SUB	#60, INPUT	:CONVERT FROM ASCII
614	0000156	112712	177777	MOVB	#-1, (R2)	:SET UP MASK
615	0000162	106312		ASLB	(R2)	:CLEAR MASK BIT BY BIT
616	0000164	103357		BCC	GETLEN	:BAD INPUT, > 8
617	0000166	105337	016512	DECB	INPUT	:COUNT
618	0000172	001373		BNE	2\$	:BRANCH IF NOT DONE
619	0000174	132712	000037	BIT3	#37, (R2)	:CHECK FOR 5 CHAR
620	0000200	001351		BNE	GETLEN	:BAD INPUT, < 5
621	0000202	005202		INC	R2	:INC TO NEXT LINE GROUP
622	0000204	0000004	017435	GETPAR: TYPE,	MSGPAR	:TYPE "PARITY (NO, ODD, EVEN):
623	0000210	004737	016364	JSR	PC,READS	:READ INPUT FROM TTY
624	0000214	005737	016512	TST	INPUT	:CHECK FOR CR
625	0000220	001415		BEQ	3\$	:BRANCH IF DEFAULT
626	0000222	122737	000116	CMPB	#116, INPUT	:CHECK FOR "N"
627	0000230	001411		BEQ	3\$	:BRANCH IF "NO"
628	0000232	122737	000105	CMPB	#105, INPUT	:CHECK FOR "E"
629	0000240	001405		BEQ	3\$	:BRANCH IF "EVEN"
630	0000242	122737	000117	CMPB	#117, INPUT	:CHECK FOR "O"
631	0000250	001355		BNE	GETPAR	:BRANCH IF NONE
632	0000252	150413		BISB	R4, (R3)	:SET THE ODD PARITY FLAG
633	0000254	005203		INC	R3	:UP DATE PARITY TABLE POINTER
634	0000256	032701	000017	BIT	#17,R1	:CHECK FOR NEXT UNIT
635	0000262	001303		BNE	TYPLIN	:BRANCH IF NOT
636	0000264	012703	001276	MOV	#PARITY,R3	:RESET PARITY TABLE POINTER
637	0000270	006304		ASL	R4	:FLAG TO NEXT UNIT
638	0000272	005200		INC	R0	:COUNT UNITS
639	0000274	020037	001234	CMP	R0,UNITS	:CHECK FOR LAST
640	0000300	001274		BNE	TYPLIN	:BRANCH IF MORE

MAINDEC-11-DZDJA-D
DZDJD.P11

DJ11 LOGIC TESTS
SETUP AREA

MACY11 27(732) 21-SEP-76 13:43 PAGE 19

723				RESTAR: RESET		:ISSUE RESET
724	002302	000005		MOV	#ICNT, SP	:SET UP STACK POINTER
725	002304	012706	001200	TST	FTIME	
726	002310	005737	001322	BEG	CLRVEC	
727	002314	001410		CMP	#SWREG, SWR	
728	002316	022737	000176 001316	BNE	CLRVEC	
729	002324	001004		JSR	PC, CNTLU	
730	002326	004737	017770	CLR	FTIME	
731	002332	005037	001322	MOV	#300, R0	:BEGINNING OF FLOATING VECTORS
732	002336	012700	000300	TST	(R0)+	
733	002342	005720		MOV	R0, -2(R0)	:":+2"
734	002344	010060	177776	MOV	#IOT, (R0)+	:":IOT"
735	002350	012720	000004	CMP	#1000, R0	
736	002354	022700	001000	BNE	1\$	
737	002360	001370		ADD	#10, CSR	:UPDATE DEVICE ADDRESS TO NEXT UNIT
738	002362	062737	000010 001210	ADD	#10, RCVVEC	:UPDATE DEVICE VECTOR ADDRESS
739	002370	062737	000010 001220	ADD	#4, DJLEN	:MOVE CHAR TABLE POINTER
740	002376	062737	000004 001306	ASL	DJPAR	:UPDATE PARITY FLAG MARKER
741	002404	006337	001310	CMP	DJUUT, UNITS	
742	002410	023737	001304 001234	BLT	2\$	
743	002416	002416		CLR	DJUUT	
744	002420	005037	001304	MOV	DEVADR, CSR	
745	002424	013737	001230 001210	MOV	VECADR, RCVVEC	
746	002432	013737	001232 001220	MOV	#LENGTH, DJLEN	:INIT CHAR TABLE POINTER
747	002440	012737	001236 001306	MOV	#1, DJPAR	:INIT PARITY FLAG MARKER
748	002446	012737	000001 001310	INC	DJUUT	
749	002454	005237	001304	MOV	CSR, R1	:SET UP ALL THE REGISTER ADDRESSES
750	002460	013701	001210	ADD	#2, R1	:ADD 2
751	002464	062701	000002	MOV	R1, RBUF	:SET UP RECEIVER BUFFER
752	002470	010137	001212	ADD	#2, R1	:ADD 2
753	002474	062701	000002	MOV	R1, TCR	:SET UP TRANSMITTER CONTROL REG
754	002500	010137	001214			:AND BREAK STATUS REG
755				ADD	#2, R1	:ADD 2
756	002504	062701	000002	MOV	R1, TBUF	:SET UP TRANSMITTER BUFFER
757	002510	010137	001216	MOV	RCVVEC, R1	:POINTER FOR VECTOR SETUP
758	002514	013701	001220	TST	(R1)+	:INC R1
759	002520	005721		MOV	R1, RCVLVL	:SET INT LVL
760	002522	010137	001222	TST	(R1)+	:INC R1
761	002526	005721		MOV	R1, XMTVEC	:TRANSMITTER VECTOR
762	002530	010137	001224	TST	(R1)+	:INC R1
763	002534	005721		MOV	R1, XMTLVL	:XMT INT LVL ADR
764	002536	010137	001226	CLR	ICNT	
765	002542	005037	001200	CLR	LAD	
766	002546	005037	015770	SCOPE		
767	002552	104400				
768						
769						

770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825

: TEST 1: TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING
: AND ABILITY TO CLR CSR.
: PROBABLE FAULTY LOGIC: M105; M7285 (D2-6) E29
: *****

TST1: MOV #1\$, @#4 ;SET UP TIME-OUT TRAP VECTOR
MOV #LEVEL7, @#6
TST @CSR ;REFERENCE CSR (READ)
CLR @CSR ;CLEAR CSR (WRITE)
ADC @CSR ;CHECK CSR (READ AND WRITE)
BEQ 2\$;BRANCH IF OK
HLT ;CSR NOT CLEARED

BR 2\$;SKIP ISR

1\$: MOV (SP)+, R1 ;SAVE RTI ADR FOR TYPING
HLT+1 ;CAN'T REFERENCE CSR!

TST (SP)+ ;FINISH CLEARING STACK
2\$: MOV #5, @#4
CLR @#6
SCOPE

: TEST 2: TEST THAT CSR BIT1 CAN BE SET AND CLEARED
: PROBABLE FAULTY LOGIC: M7285 (D2-2) E25, E7, (D2-4) E47, E64
: *****

TST2: MOV #BIT1, @CSR ;SET BIT1
BIT #BIT1, @CSR ;CHECK THAT BIT1 IS SET
BNE .+4 ;BRANCH IF OK
HLT ;CSR BIT1 FAILED TO SET

BIT #177775, @CSR ;CHECK THAT NO OTHER BIT SET
BEQ .+4 ;BRANCH IF OK
HLT ;EXTRA BIT SET IN CSR

CLR @CSR ;CLEAR BIT1
BIT #BIT1, @CSR ;CHECK THAT BIT1 IS CLEARED
BEQ .+4 ;BRANCH IF OK
HLT ;CSR BIT1 FAILED TO CLEAR

SCOPE

: TEST 3: TEST THAT CSR BIT2 CAN BE SET AND CLEARED
: PROBABLE FAULTY LOGIC: M7285 (D2-2) E25, E7, (D2-4) E47, E64
: *****

TST3: MOV #BIT2, @CSR ;SET BIT2
BIT #BIT2, @CSR ;CHECK THAT BIT2 IS SET
BNE .+4 ;BRANCH IF OK
HLT ;CSR BIT2 FAILED TO SET

K02

MAINDEC-11-DZDJA-D
DZDJA.D.P11TST3: DJ11 LOGIC TESTS
TEST BIT2 OF CSR

MACY11 27(732) 21-SEP-76 13:43 PAGE 21

002726	032777	177773	176254	BIT	#177773, CSR	:CHECK THAT NO OTHER BIT SET
002734	001401			BEG	.+4	:BRANCH IF OK
002736	104000			HLT		:EXTRA BIT SET IN CSR
002740	005077	176244		CLR	CSR	:CLEAR BIT2
002744	032777	000004	176236	BIT	#BIT2, CSR	:CHECK THAT BIT2 IS CLEARED
002752	001401			BEG	.+4	:BRANCH IF OK
002754	104000			HLT		:CSR BIT2 FAILED TO CLEAR
002756	104400			SCOPE		

 :TEST 4: TEST THAT CSR BIT6 CAN BE SET AND CLEARED
 :PROBABLE FAULTY LOGIC: M7285 (D2-2) E4, E18, (D2-4) E47, E64
 :*****

002760	012777	000100	176222	TST4: MOV	#BIT6, CSR	:SET BIT6
002766	032777	000100	176214	BIT	#BIT6, CSR	:CHECK THAT BIT6 IS SET
002774	001001			BNE	.+4	:BRANCH IF OK
002776	104000			HLT		:CSR BIT6 FAILED TO SET
003000	032777	177677	176202	BIT	#177677, CSR	:CHECK THAT NO OTHER BIT SET
003006	001401			BEG	.+4	:BRANCH IF OK
003010	104000			HLT		:EXTRA BIT SET IN CSR
003012	005077	176172		CLR	CSR	:CLEAR BIT6
003016	032777	000100	176164	BIT	#BIT6, CSR	:CHECK THAT BIT6 IS CLEARED
003024	001401			BEG	.+4	:BRANCH IF OK
003026	104000			HLT		:CSR BIT6 FAILED TO CLEAR
003030	104400			SCOPE		

 :TEST 5: TEST THAT CSR BIT8 CAN BE SET AND CLEARED
 :PROBABLE FAULTY LOGIC: M7285 (D2-2) E6, E18, (D2-4) E47, E31
 :*****

003032	012777	000400	176150	TST5: MOV	#BIT8, CSR	:SET BIT8
003040	032777	000400	176142	BIT	#BIT8, CSR	:CHECK THAT BIT8 IS SET
003046	001001			BNE	.+4	:BRANCH IF OK
003050	104000			HLT		:CSR BIT8 FAILED TO SET
003052	032777	177377	176130	BIT	#177377, CSR	:CHECK THAT NO OTHER BIT SET
003060	001401			BEG	.+4	:BRANCH IF OK
003062	104000			HLT		:EXTRA BIT SET IN CSR
003064	005077	176120		CLR	CSR	:CLEAR BIT8
003070	032777	000400	176112	BIT	#BIT8, CSR	:CHECK THAT BIT8 IS CLEARED
003076	001401			BEG	.+4	:BRANCH IF OK
003100	104000			HLT		:CSR BIT8 FAILED TO CLEAR
003102	104400			SCOPE		

 :TEST 6: TEST THAT CSR BIT10 CAN BE SET AND CLEARED
 :PROBABLE FAULTY LOGIC: M7285 (D2-2) E30, E24, (D2-4) E47, E31
 :*****

MAINDEC-11-DZDJA-D
DZDJA.D.P11

DJ11 LOGIC TESTS
TEST BIT10 OF CSR

MACY11 27(732) 21-SEP-76 13:43 PAGE 22

```

*****
0002          ;*****
0003          ;
0004 003104 012777 002000 176076 TST6:  MOV  #BIT10,BCSR  ;SET BIT10
0005 003112 032777 002000 176070      BIT  #BIT10,BCSR  ;CHECK THAT BIT10 IS SET
0006 003120 001001          .+4      BNE      ;BRANCH IF OK
0007 003122 104000          HLT          ;CSR BIT10 FAILED TO SET
0008
0009 003124 032777 175777 176056      BIT  #175777,BCSR  ;CHECK THAT NO OTHER BIT SET
0010 003132 001401          .+4      BEQ      ;BRANCH IF OK
0011 003134 104000          HLT          ;EXTRA BIT SET IN CSR
0012
0013 003136 005077 176046          CLR  BCSR          ;CLEAR BIT10
0014 003142 032777 002000 176040      BIT  #BIT10,BCSR  ;CHECK THAT BIT10 IS CLEARED
0015 003150 001401          .+4      BEQ      ;BRANCH IF OK
0016 003152 104000          HLT          ;CSR BIT10 FAILED TO CLEAR
0017
0018 003154 104400          SCOPE
0019
0020          ;*****
0021          ;TEST 7:      TEST THAT CSR BIT12 CAN BE SET AND CLEARED
0022          ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5,E1, (D2-4) E47,E31
0023          ;*****
0024
0025 003156 012777 010000 176024 TST7:  MOV  #BIT12,BCSR  ;SET BIT12
0026 003164 032777 010000 176016      BIT  #BIT12,BCSR  ;CHECK THAT BIT12 IS SET
0027 003172 001001          .+4      BNE      ;BRANCH IF OK
0028 003174 104000          HLT          ;CSR BIT12 FAILED TO SET
0029
0030 003176 032777 167777 176004      BIT  #167777,BCSR  ;CHECK THAT NO OTHER BIT SET
0031 003204 001401          .+4      BEQ      ;BRANCH IF OK
0032 003206 104000          HLT          ;EXTRA BIT SET IN CSR
0033
0034 003210 005077 175774          CLR  BCSR          ;CLEAR BIT12
0035 003214 032777 010000 175766      BIT  #BIT12,BCSR  ;CHECK THAT BIT12 IS CLEARED
0036 003222 001401          .+4      BEQ      ;BRANCH IF OK
0037 003224 104000          HLT          ;CSR BIT12 FAILED TO CLEAR
0038
0039 003226 104400          SCOPE
0040
0041          ;*****
0042          ;TEST 10:     TEST THAT CSR BIT14 CAN BE SET AND CLEARED
0043          ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E3,E1, (D2-4) E47,E31
0044          ;*****
0045
0046 003230 012777 040000 175752 TST10: MOV  #BIT14,BCSR  ;SET BIT14
0047 003236 032777 040000 175744      BIT  #BIT14,BCSR  ;CHECK THAT BIT14 IS SET
0048 003244 001001          .+4      BNE      ;BRANCH IF OK
0049 003246 104000          HLT          ;CSR BIT14 FAILED TO SET
0050
0051 003250 032777 137777 175732      BIT  #137777,BCSR  ;CHECK THAT NO OTHER BIT SET
0052 003256 001401          .+4      BEQ      ;BRANCH IF OK
0053 003260 104000          HLT          ;EXTRA BIT SET IN CSR
0054
0055 003262 005077 175722          CLR  BCSR          ;CLEAR BIT14
0056 003266 032777 040000 175714      BIT  #BIT14,BCSR  ;CHECK THAT BIT14 IS CLEARED
0057 003274 001401          .+4      BEQ      ;BRANCH IF OK

```


MO2

MAINDEC-11-DZDJA-D
DZDJA0.P11

DJ11 LOGIC TESTS
TEST BIT14 OF CSR

MACY11 27(732) 21-SEP-76 13:43 PAGE 23

938 003276 104000

HLT

;CSR BIT14 FAILED TO CLEAR

940 003300 104400

SCOPE

:TEST 11: TEST THAT RECEIVER ENABLE (BIT0 OF THE CSR) CAN BE SET
:AND CLEARED, AND THAT CLEAR MOS (BIT3) IS WRITE ONLY.
:PROBABLE FAULTY LOGIC: M7285 (D2-2) E26,E36,E7,E24, (D2-8) E17,E14,E15
:*****

948 003302 012777 000004 175700

TST11: MOV #BIT2, @CSR

;SET MAINTENANCE MODE (BIT2)

949 003310 005005

CLR R5

;SET UP COUNTER

950 003312 052777 000010 175670

1\$: BIS #BIT3, @CSR

;SET CLEAR MOS (BIT3)

951 003320 017701 175664

MOV @CSR, R1

;SAVE CSR

952 003324 032701 000010

BIT #BIT3, R1

;CHECK CLEAR MOS (BIT3)

953 003330 001401

BEQ .+4

;BRANCH IF OK

954 003332 104001

HLT+1

;CLEAR MOS (BIT3) SET (WRITE-ONLY)

955 003334 032701 000020

BIT #BIT4, R1

;R1 = CONTENTS OF CSR

956 003340 001403

BEQ 2\$

;CHECK CLEAR MOS FLAG

957 003342 105305

DECB R5

;BRANCH IF CLEARED

958 003344 001365

BNE 1\$

;WAIT FOR MOS TO CLEAR

959 003346 104001

HLT+1

;BRANCH IF MORE TIME

960 003350 022701 000004

2\$: CMP #BIT2, R1

;CLEAR MOS FLAG (BIT4) FAILED TO CLEAR

961 003354 001401

BEQ .+4

;R1 = CONTENTS OF CSR

962 003356 104001

HLT+1

;CHECK THAT ONLY MAINTENANCE BIT SET

963 003360 052777 000001 175622

BIS #BIT0, @CSR

;CLEAR MOS CLEARED MAINTENANCE

964 003366 017701 175616

MOV @CSR, R1

;OR SET OTHER CSR BITS

965 003372 032777 000001 175610

BIT #BIT0, @CSR

;R1 = CONTENTS OF CSR

966 003400 001001

BNE .+4

;SET RECEIVER ENABLE

967 003402 104001

HLT+1

;SAVE CSR

968 003404 022777 000005 175576

CMP #5, @CSR

;CHECK THAT RECEIVER ENABLE SET

969 003412 001401

BEQ .+4

;BRANCH IF OK

970 003414 104001

HLT+1

;RECEIVER ENABLE FAILED TO SET

971 003416 042777 000001 175564

BIC #BIT0, @CSR

;R1 = CONTENTS OF CSR

972 003424 017701 175560

MOV @CSR, R1

;CHECK REST OF CSR

973 003430 022777 000004 175552

CMP #BIT2, @CSR

;BRANCH IF OK

974 003436 001401

BEQ .+4

;CSR ERROR

975 003440 104001

HLT+1

;R1 = CONTENTS OF CSR

;NOTE: IF THE TTY MODULE IS BEING USED AND DONE (BIT7) IS SET,
;THE ERROR COULD BE DUE TO MAINTENANCE OR CLEAR MOS NOT WORKING.

976 003442 104400

SCOPE

:TEST 12: TEST THAT CSR RESPONDS PROPERLY TO BYTE COMMANDS
:PROBABLE FAULTY LOGIC: M7285 (D2-4) E47
:*****

N02

MAINDEC-11-DDDJA-D
DDDDAD.F11

DJ11 LOGIC TESTS
TEST BYTE ADDRESSING OF CSR

MACY11 27(732) 21-SEP-76 13:43 PAGE 24

TST12:

003444 012777
003445 105077
003446 017701
003447 022701
003448 001401
003470 104001

052506 175536
175532
175526
052400

TST12:

MOV #052506, @CSR
CLRB @CSR
MOV @CSR, R1
CMP #052400, R1
BEQ .+4
HLT+1

;SET TEST NUMBER IN CSR
;CLR EVEN BYTE
;SAVE CSR
;CHECK CSR
;BRANCH IF OK
;EVEN BYTE CLR FAILED ON CSR
;R1 = CONTENTS OF CSR
;SET TEST NUMBER IN CSR
;INC TO ODD BYTE
;CLR ODD BYTE
;RESTORE TO EVEN
;SAVE CSR
;CHECK CSR
;BRANCH IF OK
;ODD BYTE CLR FAILED ON CSR
;R1 = CONTENTS OF CSR

003472 012777
003500 005237
003504 105077
003510 005337
003514 017701
003520 022701
003524 001401
003526 104001

052506 175510
001210
175500
001210
175470
000106

MOV #052506, @CSR
INC @CSR
CLRB @CSR
DEC @CSR
MOV @CSR, R1
CMP #000106, R1
BEQ .+4
HLT+1

SCOPE

;TEST 13: TEST THAT THE BIS AND BIC INSTRUCTIONS SET AND CLEAR R/W
;BITS OF CSR
;PROBABLE FAULTY LOGIC: M7285 (D2-4) E47

003532 005077
003536 017701
003542 001401
003544 104001

175452
175446

TST13:

CLRB @CSR
MOV @CSR, R1
BEQ .+4
HLT+1

;CLEAR THE CSR
;CHECK AND SAVE CSR
;BRANCH IF CLEARED OK
;RESET FAILED TO CLR CSR

003546 052777
003554 022777
003562 001401
003564 104000

052506 175434
052506 175426

BIS #052506,
CMP #052506,
BEQ .+4
HLT

@CSR ;SET ALL R/W BITS OF CSR
@CSR ;CHECK THAT THEY GOT SET
;BRANCH IF OK
;REG FAILED CMP

003566 042777
003574 017701
003600 001401
003602 104001

052506 175414
175410

BIC #052506,
MOV @CSR, R1
BEQ .+4
HLT+1

@CSR ;CLEAR CSR
;CHECK AND SAVE CSR
;BRANCH IF CLEARED OK
;CLR FAILED TO CLR CSR

003604 104400

SCOPE

;TEST 14: TEST BITS OF TCR FOR READ/WRITE CAPABILITY
;PROBABLE FAULTY LOGIC: M7285 (D2-2) ALL, (D2-3) E8, E20, E21, E43, E41

003606 012777
003614 017701
003620 022701
003624 001401
003626 104001

177777 175400
175374
177777

TST14:

MOV #177777, @TCR
MOV @TCR, R1
CMP #177777, R1
BEQ .+4
HLT+1

;SET ALL BITS OF TCR
;CHECK AND SAVE TCR
;CHECK THAT ALL THE BITS ARE SET
;BRANCH IF OK
;BIT(S) OF TCR FAILED TO SET

003630 005077
003634 017701
003640 001401

175360
175354

CLRB @TCR
MOV @TCR, R1
BEQ .+4

;CLEAR TCR
;CHECK THAT IT CLEARED AND SAVE
;BRANCH IF CLR

SECRET

ALSO CHECK THAT THE RIGHT LINE NO. (D) APPEARS IN TBLFF.
 PROBABLE FAULTY LOGIC: M7295 (D2-5) ALL, (D2-6) E23, E32, E33, E49, (D2-2) E3, E1

[illegible][illegible][illegible]


```
:SET MASTER TRAN SCAN ENABLE
:CHECK AND SAVE XMTB READY

:TRAN READY NEVER CAME UP
:TORBIT WAS SET FIRST
```

00000000	00000000	00000000	174555	55:	BIC	#BIT0.	BITR	: THEN MASTER TRAN SCAN ENABLE
00000000	00000000	00000000	174556		MOV	BCSR,	R1	: CLEAR IT OUT
00000000	00000000	00000000	174557		BFL	4		: CHECK AND SAVE TRAN READY
00000000	00000000	00000000	174558		HLT+1			: BRANCH IF OK
00000000	00000000	00000000	174559		SCOPE			: TRAN READY DIDN'T CLEAR.

```

:*****
:TEST 22:      TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 7
:PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
:*****

```

HEX ADDRESS	HEX DATA	HEX ADDRESS	HEX DATA	ASSEMBLY	COMMENT
00000000	00000000	00000000	174564	TEST2:	MOV #ISR22, @XMTVEC ;SET UP XMTR INTERRUPT VECTOR
00000001	00000000	00000001	174564		MOV #340, @XMTLVL ;AT LEVEL 7
00000002	00000000	00000002	174564		BIC #340, @#PS ;CLEAR PS LEVEL
00000003	00000000	00000003	174564		BIC #340, @#PS ;SET PS TO LEVEL 7
00000004	00000000	00000004	174564		MOV #040400, @DCSR ;SET TRAN MASTER INT. ENABLE
00000005	00000000	00000005	174564		MOV #BIT0, @TCR ;SET TRAN CONTROL BIT, LINE 0
00000006	00000000	00000006	174564		MOV @DCSR, R1 ;WAIT
00000007	00000000	00000007	174564		BPL ;
00000008	00000000	00000008	174564		BR END22 ;OK, BRANCH IF NO INTERRUPT
0004434	104000	0004434	000000	ISR22:	HLT ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 7
0004435	012715	0004435	000000		MOV #END22, (SP) ;MOVE NEW RTI ADR ONTO STACK
0004436	000002	0004436	000000		RTI
0004434	012777	0004434	174552	END22:	MOV XMTLVL, @XMTVEC
0004435	012777	0004435	174555		MOV #IOT, @XMTLVL
0004436	005077	0004436	174540		CLR @TCR
0004437	005077	0004437	174520		CLR @DCSR
0004438	042737	0004438	177776		BIC #340, @#PS
0004436	104400			SCOPE	

```

:*****
:TEST 23:      TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 6
:PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
:*****

```

HEX ADDRESS	HEX INSTRUCTION	HEX COMMENT
004540	012777 004544 174526	TST23: MOV #ISR23, 2XMTVEC ;SET UP XMTR INTERRUPT VECTOR
004544	012777 000340 174526	MOV #340, 2XMTLVL ;AT LEVEL 7
004548	042737 000340 177776	BIC #340, 2#PS ;CLEAR PS LEVEL
004552	052737 000300 177776	BIS #300, 2#PS ;SET PS TO LEVEL 6
004556	012777 040400 174462	MOV #040400, 2CSR ;SET TRAN MASTER INT. ENABLE
004560	012777 000001 174460	MOV #BIT0, 2TOR ;SET TRAN CONTROL BIT, LINE 0
004564	017701 174450	1\$: MOV 2CSR, R1 ;WAIT
004568	100375	BPL 1\$
004572	000404	BR END23 ;OK, BRANCH IF NO INTERRUPT
004576	104000	ISR23: HLT ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 6
004580	012716 004564	MOV #END23, (SP) ;MOVE NEW RTI ADR ONTO STACK
004584	000000	RTI

SCOPE

```

:*****
:TEST 24:      TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 5
:PROBABLE FAULTY LOGIC:  M7921 WIRING, PROPER PRIORITY CHIP
:*****

```

```

TST24:  MOV      #ISR24, @XMTVEC      ;SET UP XMTR INTERRUPT VECTOR
        MOV      #340, @XMTLVL       ;AT LEVEL 7
        BIC      #340, @#PS          ;CLEAR PS LEVEL
        BIS      #240, @#PS          ;SET PS TO LEVEL 5
        MOV      #040400, @CSR       ;SET TRAN MASTER INT. ENABLE
        MOV      @BIT0, @TCR         ;SET TRAN CONTROL BIT, LINE 0
1$:      MOV      @CSR, R1            ;WAIT
        BPL      1$
        BR       END24               ;OK. BRANCH IF NO INTERRUPT

```

```
ISR24:  HLT                ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 5
        MOV                ;MOVE NEW RTI ADR ONTO STACK
        RTI                #END24, (SP)
```

```
END24:  MOV     XMTLVL,3XMTVEC
         MOV     #IOT, 3XMTLVL
         CLR     3TCR
         CLR     3CSR
         BIC     #340,3#PS
```

```

:*****
:TEST 25:      TEST THAT INTERRUPT OCCURS AT LEVEL 4
:PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
:*****

```

```

TST25:  MOV      #ISR25,0XMTVEC      ;SET UP XMTR INTERRUPT VECTOR
        MOV      #340,0XMTLVL      ;AT LEVEL 7
        BIC      #340,0#PS         ;CLEAR PS LEVEL
        BIS      #200,0#PS         ;SET PS TO LEVEL 4
        MOV      #040400,0CSR      ;SET TRAN MASTER INT. ENABLE
        MOV      #BIT0, 0TCR       ;SET TRAN CONTROL BIT. LINE 0
13:      MOV      0CSR, R1           ;WAIT
        BPL      13
        HLT
        BR        END25
        END25
        ;SHOULD HAVE INTERRUPTED AT 13
        ;CONTINUE

```

```
ISR25:  MOV      #END25,(SP)      ;MOVE NEW RTI ADR ONTO STACK
         RTI
```

MAINDEC-11-DZDJA-D
DZDJA.D.P11

TST25:

DJ11 LOGIC TESTS

TEST TRANSMIT INTERRUPT LEVEL

MACY11 27(732) 21-SEP-76 13:43 PAGE 30

```

1330 005014 013777 001226 174202 END25: MOV XMTLVL, XMTVEC
1331 005022 012777 000004 174176 MOV #IOT, XMTLVL
1332 005030 005077 174150 CLR STCR
1333 005034 005077 174150 CLR STCSR
1334 005040 042737 000340 177776 BIC #340, X#PS
1335 005046 104400 SCOPE

```

```

:*****
:TEST 26: TEST THAT INTERRUPT OCCURS AT LEVEL 3
:PROBABLE FAULTY LOGIC: M7821 WIRING, PROPER PRIORITY CHIP
:*****

```

```

1336 005050 012777 005126 174146 TST26: MOV #ISR26, XMTVEC :SET UP XMTR INTERRUPT VECTOR
1337 005056 012777 000340 174142 MOV #340, XMTLVL :AT LEVEL 7
1338 005064 042737 000340 177776 BIC #340, X#PS :CLEAR PS LEVEL
1339 005072 052737 000140 177776 BIS #140, X#PS :SET PS TO LEVEL 3
1340 005100 012777 040400 174102 MOV #040400, STCSR :SET TRAN MASTER INT. ENABLE
1341 005106 012777 000001 174100 MOV #BIT0, STCR :SET TRAN CONTROL BIT, LINE 0
1342 005114 017701 174070 1$: MOV STCSR, R1 :WAIT
1343 005120 100375 BPL 1$ :SHOULD HAVE INTERRUPTED AT LEVEL 3
1344 005122 104000 HLT :CONTINUE
1345 005124 000403 BR END26
1346 005126 012716 005134 ISR26: MOV #END26, (SP) :MOVE NEW RTI ADR ONTO STACK
1347 005132 000002 RTI
1348 005134 013777 001226 174062 END26: MOV XMTLVL, XMTVEC
1349 005142 012777 000004 174056 MOV #IOT, XMTLVL
1350 005150 005077 174040 CLR STCR
1351 005154 005077 174030 CLR STCSR
1352 005160 042737 000340 177776 BIC #340, X#PS
1353 005166 104400 SCOPE

```

```

:*****
:TEST 27: TEST THAT INTERRUPT OCCURS AT LEVEL 2
:PROBABLE FAULTY LOGIC: M7821 WIRING, PROPER PRIORITY CHIP
:*****

```

```

1354 005170 012777 005246 174026 TST27: MOV #ISR27, XMTVEC :SET UP XMTR INTERRUPT VECTOR
1355 005176 012777 000340 174022 MOV #340, XMTLVL :AT LEVEL 7
1356 005204 042737 000340 177776 BIC #340, X#PS :CLEAR PS LEVEL
1357 005212 052737 000100 177776 BIS #100, X#PS :SET PS TO LEVEL 2
1358 005220 012777 040400 173752 MOV #040400, STCSR :SET TRAN MASTER INT. ENABLE
1359 005226 012777 000001 173750 MOV #BIT0, STCR :SET TRAN CONTROL BIT, LINE 0
1360 005234 017701 173750 1$: MOV STCSR, R1 :WAIT
1361 005240 100375 BPL 1$ :SHOULD HAVE INTERRUPTED AT LEVEL 2
1362 005242 104000 HLT :CONTINUE
1363 005244 000403 BR END27
1364 005246 012716 005254 ISR27: MOV #END27, (SP) :MOVE NEW RTI ADR ONTO STACK
1365 005252 000002 RTI

```


1386 005254 013777 001226 173742
1387 005254 012777 000004 173736
1388 005270 005077 173720
1389 005274 005077 173710
1390 005300 042737 000340 177776
1391 005306 104400

END27: MOV XMTLVL, 3XMTVEC
MOV #107, 3XMTLVL
CLR 3TCR
CLR 3CSR
BIC #340, 3#PS

SCOPE

:TEST 30: TEST THAT INTERRUPT OCCURS AT LEVEL 1
:PROBABLE FAULTY LOGIC: M7821 WIRING, PROPER PRIORITY CHIP

1400 005310 012777 005366 173706
1401 005316 012777 000340 173702
1402 005324 042737 000340 177776
1403 005332 052737 000040 177776
1404 005340 012777 040400 173642
1405 005346 012777 000001 173640
1406 005354 017701 173630
1407 005360 100375
1408 005362 104000
1409 005364 000403
1410 005366 012716 005374
1411 005372 000002
1412 005374 013777 001226 173622
1413 005402 012777 000004 173616
1414 005410 005077 173600
1415 005414 005077 173570
1416 005420 042737 000340 177776
1417 005426 104400

TST30: MOV #ISR30, 3XMTVEC ;SET UP XMTR INTERRUPT VECTOR
MOV #340, 3XMTLVL ;AT LEVEL 7
BIC #340, 3#PS ;CLEAR PS LEVEL
BIS #040, 3#PS ;SET PS TO LEVEL 1
MOV #040400, 3CSR ;SET TRAN MASTER INT. ENABLE
MOV #BIT0, 3TCR ;SET TRAN CONTROL BIT, LINE 0
1\$: MOV 3CSR, R1 ;WAIT
BPL 1\$
HLT ;SHOULD HAVE INTERRUPTED AT LEVEL 1
BR END30 ;CONTINUE

ISR30: MOV #END30, (SP) ;MOVE NEW RTI ADR ONTO STACK
RTI

END30: MOV XMTLVL, 3XMTVEC
MOV #107, 3XMTLVL
CLR 3TCR
CLR 3CSR
BIC #340, 3#PS

SCOPE

:TEST 31: TEST THAT INTERRUPT OCCURS AT LEVEL 0
:PROBABLE FAULTY LOGIC: M7821 WIRING, PROPER PRIORITY CHIP

1420 005430 012777 005506 173566
1421 005436 012777 000340 173562
1422 005444 042737 000340 177776
1423 005452 052737 000000 177776
1424 005460 012777 040400 173522
1425 005466 012777 000001 173520
1426 005474 017701 173510
1427 005500 100375
1428 005502 104000
1429 005504 000403
1430 005506 012716 005514
1431 005512 000002

TST31: MOV #ISR31, 3XMTVEC ;SET UP XMTR INTERRUPT VECTOR
MOV #340, 3XMTLVL ;AT LEVEL 7
BIC #340, 3#PS ;CLEAR PS LEVEL
BIS #000, 3#PS ;SET PS TO LEVEL 0
MOV #040400, 3CSR ;SET TRAN MASTER INT. ENABLE
MOV #BIT0, 3TCR ;SET TRAN CONTROL BIT, LINE 0
1\$: MOV 3CSR, R1 ;WAIT
BPL 1\$
HLT ;SHOULD HAVE INTERRUPTED AT LEVEL 0
BR END31 ;CONTINUE

ISR31: MOV #END31, (SP) ;MOVE NEW RTI ADR ONTO STACK
RTI

MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS MACY11 27(732) 21-SEP-76 13:43 PAGE 32
 DZDJA.D.P11 TST31: TEST TRANSMIT INTERRUPT LEVEL

```

1442 005514 013777 001226 173502 END31: MOV XMTLVL, @XMTVEC
1443 005522 012777 000004 173476 MOV #10T, @XMTLVL
1444 005530 005077 173460 CLR @TCR
1445 005534 005077 173450 CLR @CSR
1446 005540 042737 000340 177776 BIC #340, @PS
1447
1448 005546 104400 SCOPE
1449
1450 005550 005037 001302 CLR TIMER ;INITIALIZE TIMER
1451 005554 012737 005562 015770 MOV #.+6, LAD ;RESET LOOP ADDRESS
  
```

```

*****
TEST 32: TEST THAT LINE 0 CAN TRANSMIT AND
          RECEIVE A CHARACTER. (377)
1$: CHECKS THAT DONE SETS IN REASONABLE TIME.
2$: CHECKS THAT CHAR PRESENT IS IN FI/FO
3$: CHECKS THAT NO ERRORS IN FI/FO
4$: CHECKS THAT RIGHT LINE # (0) IN FI/FO
5$: CHECKS FOR RIGHT CHARACTER LENGTH
6$: CHECKS THAT CORRECT DATA WAS RECEIVED
7$: CHECKS THAT CHARACTER PRESENT CLEARS
8$: CHECKS THAT DONE CLEARS
PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
*****
  
```

INITIALIZE
 DEVICE"CSR"REGISTER

```

1452 005562 004737 015566 TST32: JSR PC, @#INITD SET:
                                     :BIT2 = MAINTENANCE
                                     :BIT3 = CLEAR MOS
                                     :BIT8 = MASTER XMTR SCAN ENB
1453                                     :WAIT FOR MOS TO CLEAR
1454                                     SET:
                                     :BITD = RECEIVER ENABLE
1455
1456 005566 052777 000001 173420 LOP32: BIS #BITD, @TCR ;SET XMTR CONTROL BIT, LINE 0
1457 005574 017701 173410 MOV @CSR, R1 ;WAIT FOR XMTR READY
1458 005600 100375 BPL LOP32
1459 005602 012777 000377 173406 MOV #377, @TBUF ;SEND A RUBOUT
1460 005610 005000 CLR R0 ;CLEAR COUNTER
1461 005612 105305 1$: DECB R5 ;SHORT WAIT LOOP
1462 005614 001376 BNE 1$
1463 005616 105777 173366 TSTB @CSR ;WAIT FOR DONE
1464 005622 100405 BMI 2$ ;BRANCH WHEN DONE
1465 005624 005200 INC R0 ;TIME COUNTER
1466 005626 001371 BNE 1$ ;BRANCH IF NOT TIME-OUT
1467 005630 017701 173354 MOV @CSR, R1 ;SAVE CSR
1468 005634 104001 HLT+1 ;DONE NEVER CAME UP
1469                                     :R1 = CONTENTS OF CSR
1470 005636 050037 001302 2$: BIS R0, TIMER ;SAVE TIMER
1471 005642 017701 173344 MOV @RBUF, R1 ;READ THE FI/FO
1472 005646 100401 BMI .+4 ;BRANCH IF CHARACTER READY
1473 005650 104001 HLT+1 ;CHARACTER READY DIDN'T SET
1474                                     :R1 = CONTENTS OF RBUF
1475 005652 032701 070000 3$: BIT #70000, R1 ;CHECK FOR ERROR BITS
1476 005656 001401 BEQ .+4 ;BRANCH IF NONE
  
```


J03

MAINDEC-11-DZDJA-D
DZDJA.D.P11

TST32:

DJ11 LOGIC TESTS

MACY11 27(732)

21-SEP-76

13:43 PAGE 33

TEST ALL OF LINE 0 TRANSMIT AND RECEIVE LOGIC

1498 005660 104001

HLT+1

:ERROR IN RECEIVED CHAR

:R1 = CONTENTS OF RBUF

:BIT14=UART OVERRUN

:BIT13=FRAMING ERROR

:BIT12=PARITY ERROR

:DUPLICATE DATA WORD

:MASK LINE#

:LINE # IN LOW BYTE

:CHECK LINE #

:BRANCH IF OK

:WRONG LINE # RECEIVED

:R1 = CONTENTS OF RBUF

:BITS8-11 = LINE #

:GET MASK OF CHARACTER

:CHECK CHAR LENGTH.

:BRANCH IF OK

:WRONG CHARACTER LENGTH

:R1=DATA FROM FI/FO

:R2=MASK (BITS SET NOT EXPECTED)

:REVERSE THE MASK

:CHECK THE ACTUAL DATA

:BRANCH IF OK

:WRONG CHAR LEN OR DATA ERROR

:R1=DATA FROM FI/FO (COMPLETE WORD)

:R2=DATA (LOW BYTE) EXPECTED

:READ FI/FO

:BRANCH IF CHAR PRESENT NOT SET

:CHARACTER PRESENT STAYED SET

:R1 = CONTENTS OF RBUF

:SAVE THE CSR

:CHECK THE CSR

:BRANCH IF OK

:DONE DIDN'T CLEAR OR OTHER CSR ERROR

:R1 = CONTENTS OF CSR

:CLEAR TCR

:CLEAR CSR

1503 005662 010102

4\$:

MOV

R1,

R2

1504 005664 042702

170377

SIC

#170377,

R2

1505 005670 000302

SWAB

R2

1506 005672 122702

000000

CMPB

#0.,

R2

1507 005676 001401

BEQ

.+4

1508 005700 104001

HLT+1

1509

1510

1511 005702 117702

173400

5\$:

MOVB

3DJLEN,

R2

1512 005706 130201

BITB

R2,

R1

1513 005710 001401

BEQ

.+4

1514 005712 104002

HLT+2

1515

1516

1517 005714 105102

6\$:

COMB

R2

1518 005716 120102

CMPB

R1,

R2

1519 005720 001401

BEQ

.+4

1520 005722 104002

HLT+2

1521

1522

1523 005724 017701

173262

7\$:

MOV

3RBUF,

R1,

1524 005730 100001

BPL

.+4

1525 005732 104001

HLT+1

1526

1527 005734 017701

173250

8\$:

MOV

3CSR,

R1

1528 005740 022701

100405

CMP

#100405,

R1

1529 005744 001401

BEQ

.+4

1530 005746 104001

HLT+1

1531

1532 005750 005077

173240

CLR

3TCR

1533 005754 005077

173230

CLR

3CSR

1534 005760 104400

SCOPE

:*****

:TEST 33: TEST THAT LINE 1 CAN TRANSMIT AND
:RECEIVE A CHARACTER. (377)

:1\$: CHECKS THAT DONE SETS IN REASONABLE TIME.

:2\$: CHECKS THAT CHAR PRESENT IS IN FI/FO

:3\$: CHECKS THAT NO ERRORS IN FI/FO

:4\$: CHECKS THAT RIGHT LINE # (1) IN FI/FO

:5\$: CHECKS FOR RIGHT CHARACTER LENGTH

:6\$: CHECKS THAT CORRECT DATA WAS RECEIVED

:7\$: CHECKS THAT CHARACTER PRESENT CLEARS

:8\$: CHECKS THAT DONE CLEARS

:PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD 003 SERIES

:*****

:INITIALIZE
:DEVICE"CSR"REGISTER

SET:

TST33: JSR PC, 3#INITD :BIT2 = MAINTENANCE

1553 005762 004737 015566

MAINDEC-11-DZDJA-D
DZDJD.P11

TST33:

DJ11 LOGIC TESTS

MACY11 27(732) 21-SEP-76 13:43 PAGE 35

TEST ALL OF LINE 1 TRANSMIT AND RECEIVE LOGIC

1610 006134 017701 173050
1611 006140 022701 100405
1612 006144 001401
1613 006146 104001
1614
1615 006150 005077 173040
1616 006154 005077 173030
1617 006160 104400

9\$: MOV 2CSR, R1 ;SAVE THE CSR
CMP #100405, R1 ;CHECK THE CSR
BEQ .+4 ;BRANCH IF OK
HLT+1 ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
CLR 2TCR ;R1 = CONTENTS OF CSR
CLR 2CSR ;CLEAR TCR
SCOPE ;CLEAR CSR

TEST 34: TEST THAT LINE 2 CAN TRANSMIT AND
RECEIVE A CHARACTER. (377)

1\$: CHECKS THAT DONE SETS IN REASONABLE TIME.
2\$: CHECKS THAT CHAR PRESENT IS IN FI/FO
3\$: CHECKS THAT NO ERRORS IN FI/FO
4\$: CHECKS THAT RIGHT LINE # (2) IN FI/FO
5\$: CHECKS FOR RIGHT CHARACTER LENGTH
6\$: CHECKS THAT CORRECT DATA WAS RECEIVED
7\$: CHECKS THAT CHARACTER PRESENT CLEARS
8\$: CHECKS THAT DONE CLEARS

PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES

INITIALIZE
DEVICE"CSR"REGISTER

1626 006162 004737 015566

TST34: JSR PC, 2*INITD SET:
;BIT2 = MAINTENANCE
;BIT3 = CLEAR MOS
;BIT8 = MASTER XMTR SCAN ENB

;WAIT FOR MOS TO CLEAR

SET:
;BIT0 = RECEIVER ENABLE

1643 006166 052777 000004 173020

LOP34: BIS #BIT2, 2TCR ;SET XMTR CONTROL BIT, LINE 2
MOV 2CSR, R1 ;WAIT FOR XMTR READY

1644 006174 017701 173010

1645 006200 100375

1646 006202 012777 000377 173006

1647 006210 005000

1648 006212 105305

1649 006214 001376

1650 006216 105777 172766

1651 006222 100405

1652 006224 005200

1653 006226 001371

1654 006230 017701 172754

1655 006234 104001

1\$: DECB R5 ;SHORT WAIT LOOP

BNE 1\$
TSTB 2CSR ;WAIT FOR DONE
BMI 2\$;BRANCH WHEN DONE
INC R0 ;TIME COUNTER
BNE 1\$;BRANCH IF NOT TIME-OUT
MOV 2CSR, R1 ;SAVE CSR
HLT+1 ;DONE NEVER CAME UP

2\$: BIS R0, TIMER ;SAVE TIMER
MOV 2RBUF, R1 ;READ THE FI/FO
BMI .+4 ;BRANCH IF CHARACTER READY
HLT+1 ;CHARACTER READY DIDN'T SET

3\$: BIT #70000, R1 ;R1 = CONTENTS OF RBUF
BEQ .+4 ;CHECK FOR ERROR BITS
HLT+1 ;BRANCH IF NONE
;ERROR IN RECEIVED CHAR
;R1 = CONTENTS OF RBUF

1656 006236 050037 001302

1657 006242 017701 172744

1658 006246 100401

1659 006250 104001

1660 006252 032701 070000

1661 006256 001401

1662 006260 104001

1663

1664

1665

M03

1666
1667
1668
1669
1670
1671
1672
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

006262 010102
006264 042702 170377
006270 000302
006272 122702 000002
006276 001401
006300 104001

006302 117702 173000
006306 130201
006310 001401
006312 104002

006314 105102
006316 120102
006320 001401
006322 104002

006324 017701 172662
006330 100001
006332 104001

006334 017701 172650
006340 022701 100405
006344 001401
006346 104001

006350 005077 172640
006354 005077 172630
006360 104400

4\$: MOV R1, R2
BIC #170377, R2
SWAB R2
CMPB #2, R2
BEQ .+4
HLT+1

5\$: MOVB 3DJLEN, R2
BITB R2, R1
BEQ .+4
HLT+2

6\$: COMB R2
CMPB R1, R2
BEQ .+4
HLT+2

7\$: MOV 3RBUF, R1
BPL .+4
HLT+1

8\$: MOV 3CSR, R1
CMP #100405, R1
BEQ .+4
HLT+1

CLR 3TCR
CLR 3CSR
SCOPE

:BIT14=UART OVERRUN
:BIT13=FRAMING ERROR
:BIT12=PARITY ERROR
:DUPLICATE DATA WORD
:MASK LINE#
:LINE # IN LOW BYTE
:CHECK LINE #
:BRANCH IF OK
:WRONG LINE # RECEIVED
:R1 = CONTENTS OF RBUF
:BITS8-11 = LINE #
:GET MASK OF CHARACTER
:CHECK CHAR LENGTH.
:BRANCH IF OK
:WRONG CHARACTER LENGTH
:R1=DATA FROM FI/FO
:R2=MASK (BITS SET NOT EXPECTED)
:REVERSE THE MASK
:CHECK THE ACTUAL DATA
:BRANCH IF OK
:WRONG CHAR LEN OR DATA ERROR
:R1=DATA FROM FI/FO (COMPLETE WORD)
:R2=DATA (LOW BYTE) EXPECTED
:READ FI/FO
:BRANCH IF CHAR PRESENT NOT SET
:CHARACTER PRESENT STAYED SET
:R1 = CONTENTS OF RBUF
:SAVE THE CSR
:CHECK THE CSR
:BRANCH IF OK
:DONE DIDN'T CLEAR OR OTHER CSR ERROR
:R1 = CONTENTS OF CSR
:CLEAR TCR
:CLEAR CSR

:TEST 35: TEST THAT LINE 3 CAN TRANSMIT AND
:RECEIVE A CHARACTER. (377)
:1\$: CHECKS THAT DONE SETS IN REASONABLE TIME.
:2\$: CHECKS THAT CHAR PRESENT IS IN FI/FO
:3\$: CHECKS THAT NO ERRORS IN FI/FO
:4\$: CHECKS THAT RIGHT LINE # (3) IN FI/FO
:5\$: CHECKS FOR RIGHT CHARACTER LENGTH
:6\$: CHECKS THAT CORRECT DATA WAS RECEIVED
:7\$: CHECKS THAT CHARACTER PRESENT CLEARS
:8\$: CHECKS THAT DONE CLEARS
:PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES

: INITIALIZE
: DEVICE"CSR"REGISTER

006362 004737 015566

TST35: JSR PC, 3#INITD

SET:
:BIT2 = MAINTENANCE
:BIT3 = CLEAR MOS
:BITS = MASTER XMTR SCAN ENB

Address	Op Code	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------


```

:BIT14=UART OVERRUN
:BIT13=FRAMING ERROR
:BIT12=PARITY ERROR
:DUPLICATE DATA WORD
:MASK LINE#
:LINE # IN LOW BYTE
:CHECK LINE #
:BRANCH IF OK
:WRONG LINE # RECEIVED
:R1 = CONTENTS OF RBUF
:BITS8-11 = LINE #
:GET MASK OF CHARACTER
:CHECK CHAR LENGTH.
:BRANCH IF OK
:WRONG CHARACTER LENGTH
:R1=DATA FROM FI/FO
:R2=MASK (BITS SET NOT EXPECTED)
:REVERSE THE MASK
:CHECK THE ACTUAL DATA
:BRANCH IF OK
:WRONG CHAR LEN OR DATA ERROR
:R1=DATA FROM FI/FO (COMPLETE WORD)
:R2=DATA (LOW BYTE) EXPECTED
:READ FI/FO
:BRANCH IF CHAR PRESENT NOT SET
:CHARACTER PRESENT STAYED SET
:R1 = CONTENTS OF RBUF
:SAVE THE CSR
:CHECK THE CSR
:BRANCH IF OK
:DONE DIDN'T CLEAR OR OTHER CSR ERROR
:R1 = CONTENTS OF CSR
:OUT TO ICR
:CLEAR CSR

```

[illegible]

```

INITIALIZE
DEVICE"CSR"REGISTER

```

006774 00-737 015566

13-37: JSR FC. 20INITD

[illegible]

```

      :WAIT FOR MOS TO CLEAR
      :
      :SET:
      :BIT0 = RECEIVER ENABLE
      :SET XMTR CONTROL BIT, LINE 5
      :WAIT FOR XMTR READY
      :SEND A RUBOUT
      :CLEAR COUNTER
      :SHORT WAIT LOOP
      :WAIT FOR DONE
      :BRANCH WHEN DONE
      :TIME COUNTER
      :BRANCH IF NOT TIME-OUT
      :SAVE CSR
      :DONE NEVER CAME UP
      :R1 = CONTENTS OF CSR
      :SAVE TIMER
      :READ THE FI/FO
      :BRANCH IF CHARACTER READY
      :CHARACTER READY DIDN'T SET
      :R1 = CONTENTS OF RBUF
      :CHECK FOR ERROR BITS
      :BRANCH IF NONE
      :ERROR IN RECEIVED CHAR
      :R1 = CONTENTS OF RBUF
      :BIT14=UART OVERRUN
      :BIT13=FRAMING ERROR
      :BIT12=PARITY ERROR
      :DUPLICATE DATA WORD
      :MASK LINE#
      :LINE # IN LOW BYTE
      :CHECK LINE #
      :BRANCH IF OK
      :WRONG LINE # RECEIVED
      :R1 = CONTENTS OF RBUF
      :BIT58-11 = LINE #
      :GET MASK OF CHARACTER
      :CHECK CHAR LENGTH.
      :BRANCH IF OK
      :WRONG CHARACTER LENGTH
      :R1=DATA FROM FI/FO
      :R2=MASK (BITS SET NOT EXPECTED)
      :REVERSE THE MASK
      :CHECK THE ACTUAL DATA
      :BRANCH IF OK
      :WRONG CHAR LEN OR DATA ERROR
      :R1=DATA FROM FI/FO (COMPLETE WORD)
      :R2=DATA (LOW BYTE) EXPECTED
      :READ FI/FO
      :BRANCH IF CHAR PRESENT NOT SET
      :CHARACTER PRESENT STAYED SET
      :R1 = CONTENTS OF RBUF
      :SAVE THE CSR
      :CHECK THE CSR
  
```

Address	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op465	Op466	Op467	Op468	Op469	Op470	Op471	Op472	Op473	Op474	Op475	Op476	Op477	Op478	Op479	Op480	Op481	Op482	Op483	Op484	Op485	Op486	Op487	Op488	Op489	Op490	Op491	Op492	Op493	Op494	Op495	Op496	Op497	Op498	Op499	Op500	Op501	Op502	Op503	Op504	Op505	Op506	Op507	Op508	Op509	Op510	Op511	Op512	Op513	Op514	Op515	Op516	Op517	Op518	Op519	Op520	Op521	Op522	Op523	Op524	Op525	Op526	Op527	Op528	Op529	Op530	Op531	Op532	Op533	Op534	Op535	Op536	Op537	Op538	Op539	Op540	Op541	Op542	Op543	Op544	Op545	Op546	Op547	Op548	Op549	Op550	Op551	Op552	Op553	Op554	Op555	Op556	Op557	Op558	Op559	Op560	Op561	Op562	Op563	Op564	Op565	Op566	Op567	Op568	Op569	Op570	Op571	Op572	Op573	Op574	Op575	Op576	Op577	Op578	Op579	Op580	Op581	Op582	Op583	Op584	Op585	Op586	Op587	Op588	Op589	Op590	Op591	Op592	Op593	Op594	Op595	Op596	Op597	Op598	Op599	Op600	Op601	Op602	Op603	Op604	Op605	Op606	Op607	Op608	Op609	Op610	Op611	Op612	Op613	Op614	Op615	Op616	Op617	Op618	Op619	Op620	Op621	Op622	Op623	Op624	Op625	Op626	Op627	Op628	Op629	Op630	Op631	Op632	Op633	Op634	Op635	Op636	Op637	Op638	Op639	Op640	Op641	Op642	Op643	Op644	Op645	Op646	Op647	Op648	Op649	Op650	Op651	Op652	Op653	Op654	Op655	Op656	Op657	Op658	Op659	Op660	Op661	Op662	Op663	Op664	Op665	Op666	Op667	Op668	Op669	Op670	Op671	Op672	Op673	Op674	Op675	Op676	Op677	Op678	Op679	Op680	Op681	Op682	Op683	Op684	Op685	Op686	Op687	Op688	Op689	Op690	Op691	Op692	Op693	Op694	Op695	Op696	Op697	Op698	Op699	Op700	Op701	Op702	Op703	Op704	Op705	Op706	Op707	Op708	Op709	Op710	Op711	Op712	Op713	Op714	Op715	Op716	Op717	Op718	Op719	Op720	Op721	Op722	Op723	Op724	Op725	Op726	Op727	Op728	Op729	Op730	Op731	Op732	Op733	Op734	Op735	Op736	Op737	Op738	Op739	Op740	Op741	Op742	Op743	Op744	Op745	Op746	Op747	Op748	Op749	Op750	Op751	Op752	Op753	Op754	Op755	Op756	Op757	Op758	Op759	Op760	Op761	Op762	Op763	Op764	Op765	Op766	Op767	Op768	Op769	Op770	Op771	Op772	Op773	Op774	Op775	Op776	Op777	Op778	Op779	Op780	Op781	Op782	Op783	Op784	Op785	Op786	Op787	Op788	Op789	Op790	Op791	Op792	Op793	Op794	Op795	Op796	Op797	Op798	Op799	Op800	Op801	Op802	Op803	Op804	Op805	Op806	Op807	Op808	Op809	Op810	Op811	Op812	Op813	Op814	Op815	Op816	Op817	Op818	Op819	Op820	Op821	Op822	Op823	Op824	Op825	Op826	Op827	Op828	Op829	Op830	Op831	Op832	Op833	Op834	Op835	Op836	Op837	Op838	Op839	Op840	Op841	Op842	Op843	Op844	Op845	Op846	Op847	Op848	Op849	Op850	Op851	Op852	Op853	Op854	Op855	Op856	Op857	Op858	Op859	Op860	Op861	Op862	Op863	Op864	Op865	Op866	Op867	Op868	Op869	Op870	Op871	Op872	Op873	Op874	Op875	Op876	Op877	Op878	Op879	Op880	Op881	Op882	Op883	Op884	Op885	Op886	Op887	Op888	Op889	Op890	Op891	Op892	Op893	Op894	Op895	Op896	Op897	Op898	Op899	Op900	Op901	Op902	Op903	Op904	Op905	Op906	Op907	Op908	Op909	Op910	Op911	Op912	Op913	Op914	Op915	Op916	Op917	Op918	Op919	Op920	Op921	Op922	Op923	Op924	Op925	Op926	Op927	Op928	Op929	Op930	Op931	Op932	Op933	Op934	Op935	Op936	Op937	Op938	Op939	Op940	Op941	Op942	Op943	Op944	Op945	Op946	Op947	Op948	Op949	Op950	Op951	Op952	Op953	Op954	Op955	Op956	Op957	Op958	Op959	Op960	Op961	Op962	Op963	Op964	Op965	Op966	Op967	Op968	Op969	Op970	Op971	Op972	Op973	Op974	Op975	Op976	Op977	Op978	Op979	Op980	Op981	Op982	Op983	Op984	Op985	Op986	Op987	Op988	Op989	Op990	Op991	Op992	Op993	Op994	Op995	Op996	Op997	Op998	Op999
---------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

E04

MAINDEC-11-DZDJA-D
DZDJA.D.P11DJ11 LOGIC TESTS
TEST ALL OF LINE 5 TRANSMIT AND RECEIVE LOGIC

MACY11 27(732) 21-SEP-76 13:43 PAGE 41

007156 001401
007156 104001
007156 005077 172026
007156 005077 172016
007156 104400BEQ .+4
HLT+1
CLR BCR
CLR BCSR
SCOPE:BRANCH IF OK
:DONE DIDN'T CLEAR OR OTHER CSR ERROR
:R1 = CONTENTS OF CSR
:CLEAR TCR
:CLEAR CSR

 :TEST 40: TEST THAT LINE 6 CAN TRANSMIT AND
 :RECEIVE A CHARACTER. (377)
 :1\$: CHECKS THAT DONE SETS IN REASONABLE TIME.
 :2\$: CHECKS THAT CHAR PRESENT IS IN FI/FO
 :3\$: CHECKS THAT NO ERRORS IN FI/FO
 :4\$: CHECKS THAT RIGHT LINE # (6) IN FI/FO
 :5\$: CHECKS FOR RIGHT CHARACTER LENGTH
 :6\$: CHECKS THAT CORRECT DATA WAS RECEIVED
 :7\$: CHECKS THAT CHARACTER PRESENT CLEARS
 :8\$: CHECKS THAT DONE CLEARS
 :PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES

INITIALIZE
 DEVICE"CSR"REGISTER

007174 004737 015566

:TEST 40: JSR PC, 3*INITD

SET:
 :BIT2 = MAINTENANCE
 :BIT3 = CLEAR MOS
 :BIT8 = MASTER XMTR SCAN ENB

:WAIT FOR MOS TO CLEAR

SET:
 :BIT0 = RECEIVER ENABLE

007200 052777 000100 172006
 007200 017701 171776
 007200 100375
 007200 012777 000377 171774
 007200 005000
 007200 105305
 007200 001376
 007200 105777 171754
 007200 100405
 007200 005200
 007200 001371
 007200 017701 171742
 007200 104001

LOP40: BIS #BIT6, BCR
 MOV BCSR, R1
 BPL LOP40
 MOV #377, BBUF
 CLR RO
 1\$: DECB RS
 BNE 1\$
 TSTB BCSR
 BMI 2\$
 INC RO
 BNE 1\$
 MOV BCSR, R1
 HLT+1

:SET XMTR CONTROL BIT, LINE 6
 :WAIT FOR XMTR READY
 :SEND A RUBOUT
 :CLEAR COUNTER
 :SHORT WAIT LOOP

:WAIT FOR DONE
 :BRANCH WHEN DONE
 :TIME COUNTER
 :BRANCH IF NOT TIME-OUT
 :SAVE CSR
 :DONE NEVER CAME UP
 :R1 = CONTENTS OF CSR

007250 050037 001302
 007250 017701 171732
 007250 100401
 007250 104001

2\$: BIS RO, TIMER
 MOV BBUF, R1
 BMI .+4
 HLT+1

:SAVE TIMER
 :READ THE FI/FO
 :BRANCH IF CHARACTER READY
 :CHARACTER READY DIDN'T SET
 :R1 = CONTENTS OF RBUF
 :CHECK FOR ERROR BITS
 :BRANCH IF NONE
 :ERROR IN RECEIVED CHAR
 :R1 = CONTENTS OF RBUF
 :BIT14=UART OVERRUN
 :BIT13=FRAMING ERROR

007264 032701 070000
 007264 001401
 007264 104001

3\$: BIT #70000, R1
 BEQ .+4
 HLT+1

```

:BIT12=PARITY ERROR
:DUPLICATE DATA WORD
:MASK LINE#
:LINE # IN LOW BYTE
:CHECK LINE #
:BRANCH IF OK
:WRONG LINE # RECEIVED
:R1 = CONTENTS OF RBUF
:BITS8-11 = LINE #
:GET MASK OF CHARACTER
:CHECK CHAR LENGTH.
:BRANCH IF OK
:WRONG CHARACTER LENGTH
:R1=DATA FROM FI/FO
:R2=MASK (BITS SET NOT EXPECTED)
:REVERSE THE MASK
:CHECK THE ACTURAL DATA
:BRANCH IF OK
:WRONG CHAR LEN OR DATA ERROR
:R1=DATA FROM FI/FO (COMPLETE WORD)
:R2=DATA (LOW BYTE) EXPECTED
:READ FI/FO
:BRANCH IF CHAR PRESENT NOT SET
:CHARACTER PRESENT STAYED SET
:R1 = CONTENTS OF RBUF
:SAVE THE CSR
:CHECK THE CSR
:BRANCH IF OK
:DONE DIDN'T CLEAR OR OTHER CSR ERROR
:R1 = CONTENTS OF CSR
:CLEAR TCR
:CLEAR CSR

```

```

:      INITIALIZE
:      DEVICE"CSR"REGISTER
:
:      JSR      PC,      @INITD
:
:      SET:
:      ;BIT2 = MAINTENANCE
:      ;BIT3 = CLEAR MOS
:      ;BIT8 = MASTER XMTR SCAN ENB
:
:      WAIT FOR MOS TO CLEAR
:
:      SET:

```


Address	Hex	Dec	Label	Op	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416</
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	---------

H04

MAINDEC-11-0203A-0
21-SEP-76DJ11 LOGIC TESTS
TEST 41:MACY11 27(732) 21-SEP-76 13:43 PAGE 44
TEST ALL OF LINE 7 TRANSMIT AND RECEIVE LOGIC

```

0007000 0050077 171426
0007000 0050077 171416
0007000 1044000
0007000 0050077 001306
0076000 012737 007606 015770

```

```

CLR      3TCR
CLR      3CSR
SCOPE
INC      DJLEN
MOV      #.+6, LAD

```

```

;R1 = CONTENTS OF CSR
;CLEAR TCR
;CLEAR CSR

```

```

*****
TEST 42:      TEST THAT LINE 8 CAN TRANSMIT AND
                RECEIVE A CHARACTER. (377)
1$: CHECKS THAT DONE SETS IN REASONABLE TIME.
2$: CHECKS THAT CHAR PRESENT IS IN FI/FO
3$: CHECKS THAT NO ERRORS IN FI/FO
4$: CHECKS THAT RIGHT LINE # (8) IN FI/FO
5$: CHECKS FOR RIGHT CHARACTER LENGTH
6$: CHECKS THAT CORRECT DATA WAS RECEIVED
7$: CHECKS THAT CHARACTER PRESENT CLEARS
8$: CHECKS THAT DONE CLEARS
PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
*****

```

```

INITIALIZE
DEVICE"CSR"REGISTER

```

```

007606 004737 015566

```

```

TEST42: JSR      PC,      3#INITD
SET:
;BIT2 = MAINTENANCE
;BIT3 = CLEAR MOS
;BIT8 = MASTER XMTR SCAN ENB

```

```

WAIT FOR MOS TO CLEAR

```

```

SET:
;BIT0 = RECEIVER ENABLE

```

```

007612 052777 000400 171374
007620 017701 171364
007624 100375
007626 012777 000377 171362
007634 005000
007636 105305
007640 001376
007642 105777 171342
007646 100405
007650 005200
007652 001371
007654 017701 171330
007660 104001

```

```

LOP42: BIS      #BIT8, 3TCR ;SET XMTR CONTROL BIT, LINE 8
MOV      3CSR, R1 ;WAIT FOR XMTR READY
BFL      LOP42
MOV      #377, 3TBUF ;SEND A RUBOUT
CLR      R0 ;CLEAR COUNTER
1$: DECB      R5 ;SHORT WAIT LOOP
BNE      1$
TSTB     3CSR ;WAIT FOR DONE
BMI      2$ ;BRANCH WHEN DONE
INC      R0 ;TIME COUNTER
BNE      1$ ;BRANCH IF NOT TIME-OUT
MOV      3CSR, R1 ;SAVE CSR
HLT+1 ;DONE NEVER CAME UP
2$: BIS      R0, TIMER ;R1 = CONTENTS OF CSR
MOV      3RBUF, R1 ;SAVE TIMER
BMI      .+4 ;READ THE FI/FO
HLT+1 ;BRANCH IF CHARACTER READY
3$: BIT      #70000, R1 ;CHARACTER READY DIDN'T SET
BEQ      .+4 ;R1 = CONTENTS OF RBUF
HLT+1 ;CHECK FOR ERROR BITS
;BRANCH IF NONE
;ERROR IN RECEIVED CHAR
;R1 = CONTENTS OF RBUF
;BIT14=UART OVERRUN
;BIT13=FRAMING ERROR

```

```

007676 032701 070000
007702 001401
007704 104001

```


2170	007706	010102		4\$:	MOV	R1	R2	:BIT12=PARITY ERROR
2171	007710	042702	170377		BIC	#170377	R2	:DUPLICATE DATA WORD
2172	007714	000302			SWAB	R2		:MASK LINE #
2173	007716	122702	000010		CMPB	#8	R2	:LINE # IN LOW BYTE
2174	007722	001401			BEG	+.4		:CHECK LINE #
2175	007724	104001			HLT+1			:BRANCH IF OK
2176								:WRONG LINE # RECEIVED
2177								:R1 = CONTENTS OF RBUF
2178	007726	117702	171354	5\$:	MOVB	3DJLEN	R2	:BITS8-11 = LINE #
2179	007732	130201			BITB	R2	R1	:GET MASK OF CHARACTER
2180	007734	001401			BEG	+.4		:CHECK CHAR LENGTH
2181	007736	104002			HLT+2			:BRANCH IF OK
2182								:WRONG CHARACTER LENGTH
2183								:R1=DATA FROM FI/FO
2184	007740	105102		6\$:	COMB	R2		:R2=MASK (BITS SET NOT EXPECTED)
2185	007742	120102			CMPB	R1	R2	:REVERSE THE MASK
2186	007744	001401			BEG	+.4		:CHECK THE ACTUAL DATA
2187	007746	104002			HLT+2			:BRANCH IF OK
2188								:WRONG CHAR LEN OR DATA ERROR
2189								:R1=DATA FROM FI/FO (COMPLETE WORD)
2190								:R2=DATA (LOW BYTE) EXPECTED
2191	007750	017701	171236	7\$:	MOV	3RBUF	R1	:READ FI/FO
2192	007754	100001			BPL	+.4		:BRANCH IF CHAR PRESENT NOT SET
2193	007756	104001			HLT+1			:CHARACTER PRESENT STAYED SET
2194								:R1 = CONTENTS OF RBUF
2195	007760	017701	171224	8\$:	MOV	3CSR	R1	:SAVE THE CSR
2196	007764	022701	100405		CMP	#100405	R1	:CHECK THE CSR
2197	007770	001401			BEG	+.4		:BRANCH IF OK
2198	007772	104001			HLT+1			:DONE DIDN'T CLEAR OR OTHER CSR ERROR
2199								:R1 = CONTENTS OF CSR
2200	007774	005077	171214		CLR	3TCR		:CLEAR TCR
2201	010000	005077	171204		CLR	3CSR		:CLEAR CSR
2202	010004	104400			SCOPE			

```

*****
:TEST 43:      TEST THAT LINE 9 CAN TRANSMIT AND
:              RECEIVE A CHARACTER. (377)
:
: 1$: CHECKS THAT DONE SETS IN REASONABLE TIME.
: 2$: CHECKS THAT CHAR PRESENT IS IN FI/FO
: 3$: CHECKS THAT NO ERRORS IN FI/FO
: 4$: CHECKS THAT RIGHT LINE # (9) IN FI/FO
: 5$: CHECKS FOR RIGHT CHARACTER LENGTH
: 6$: CHECKS THAT CORRECT DATA WAS RECEIVED
: 7$: CHECKS THAT CHARACTER PRESENT CLEARS
: 8$: CHECKS THAT DONE CLEARS
:PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
*****

```

```

:
:      INITIALIZE
:      DEVICE"CSR"REGISTER
:
: 010006 004737 015566
:
: TST43: JSR    PC,      3#INITD
:
:
: WAIT FOR MOS TO CLEAR
:
:      SET:
:      :BIT2 = MAINTENANCE
:      :BIT3 = CLEAR MOS
:      :BIT8 = MASTER XMTR SCAN ENB
:
:      SET:

```


K04

MAINDEC-11-DZDJA-D
DZDJAD.F11DJ11 LOGIC TESTS
TST43: TEST ALL OF LINE 9 TRANSMIT AND RECEIVE LOGIC

MACY11 27(732) 21-SEP-76 13:43 PAGE 47

;R1 = CONTENTS OF CSR
;CLEAR TCR
;CLEAR CSR010174 005077 171014
010200 005077 171004
010204 104400CLR JTCR
CLR JCSR
SCOPE

```

*****
TEST 44: TEST THAT LINE 10 CAN TRANSMIT AND
          RECEIVE A CHARACTER. (377)
1$: CHECKS THAT DONE SETS IN REASONABLE TIME.
2$: CHECKS THAT CHAR PRESENT IS IN FI/FO
3$: CHECKS THAT NO ERRORS IN FI/FO
4$: CHECKS THAT RIGHT LINE # (10) IN FI/FO
5$: CHECKS FOR RIGHT CHARACTER LENGTH
6$: CHECKS THAT CORRECT DATA WAS RECEIVED
7$: CHECKS THAT CHARACTER PRESENT CLEARS
8$: CHECKS THAT DONE CLEARS
PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD DC3 SERIES
*****

```

INITIALIZE
DEVICE"CSR"REGISTER

010206 004737 015566

TST44: JSR PC, J#INITD

SET:
;BIT2 = MAINTENANCE
;BIT3 = CLEAR MOS
;BITS = MASTER XMTR SCAN ENB

;WAIT FOR MOS TO CLEAR

SET:
;BIT0 = RECEIVER ENABLE010212 052777 002000 170774
010220 017701 170764LOP44: BIS #BIT10, JTCR
MOV JCSR, R1;SET XMTR CONTROL BIT, LINE 10
;WAIT FOR XMTR READY010224 100375
010226 012777 000377 170762BPL LOP44
MOV #377, JTBUF;SEND A RUBOUT
;CLEAR COUNTER
;SHORT WAIT LOOP010234 005000
010236 105305

1\$: DECB R5

;WAIT FOR DONE
;BRANCH WHEN DONE010240 001376
010242 105777 170742BNE 1\$
TSTB JCSR;TIME COUNTER
;BRANCH IF NOT TIME-OUT010246 100405
010250 005200BMI 2\$
INC R0;SAVE CSR
;DONE NEVER CAME UP010252 001371
010254 017701 170730BNE 1\$
MOV JCSR, R1;R1 = CONTENTS OF CSR
;SAVE TIMER010260 104001
010262 050037 001302HLT+1
2\$: BIS R0, TIMER;READ THE FI/FO
;BRANCH IF CHARACTER READY010266 017701 170720
010272 100401MOV JTBUF, R1
BMI .+4;CHARACTER READY DIDN'T SET
;R1 = CONTENTS OF RBUF010274 104001
010276 032701 070000HLT+1
3\$: BIT #70000, R1;CHECK FOR ERROR BITS
;BRANCH IF NONE010302 001401
010304 104001BEQ .+4
HLT+1;ERROR IN RECEIVED CHAR
;R1 = CONTENTS OF RBUF

010306 010102

4\$: MOV R1, R2

;BIT14=UART OVERRUN
;BIT13=FRAMING ERROR
;BIT12=PARITY ERROR
;DUPLICATE DATA WORD

L04

MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS MACY11 27(732) 21-SEP-76 13:43 PAGE 48
 DZDJA.D.F11 TST44: TEST ALL OF LINE 10 TRANSMIT AND RECEIVE LOGIC

2338	010310	042702	170377		BIC	#170377,R2	:MASK LINE#
2339	010314	000302			SWAB	R2	:LINE # IN LOW BYTE
2340	010316	122702	000012		CMPB	#10., R2	:CHECK LINE #
2341	010322	001401			BEQ	.+4	:BRANCH IF OK
2342	010324	104001			HLT+1		:WRONG LINE # RECEIVED
2343							:R1 = CONTENTS OF RBUF
2344							:BITS8-11 = LINE #
2345	010326	117702	170754	5\$:	MOVB	3DJLEN, R2	:GET MASK OF CHARACTER
2346	010332	130201			BITB	R2, R1	:CHECK CHAR LENGTH.
2347	010334	001401			BEQ	.+4	:BRANCH IF OK
2348	010336	104002			HLT+2		:WRONG CHARACTER LENGTH
2349							:R1=DATA FROM FI/FO
2350							:R2=MASK (BITS SET NOT EXPECTED)
2351	010340	105102		6\$:	COMB	R2	:REVERSE THE MASK
2352	010342	120102			CMPB	R1, R2	:CHECK THE ACTUAL DATA
2353	010344	001401			BEQ	.+4	:BRANCH IF OK
2354	010346	104002			HLT+2		:WRONG CHAR LEN OR DATA ERROR
2355							:R1=DATA FROM FI/FO (COMPLETE WORD)
2356							:R2=DATA (LOW BYTE) EXPECTED
2357	010350	017701	170636	7\$:	MOV	3RBUF, R1	:READ FI/FO
2358	010354	100001			BPL	.+4	:BRANCH IF CHAR PRESENT NOT SET
2359	010356	104001			HLT+1		:CHARACTER PRESENT STAYED SET
2360							:R1 = CONTENTS OF RBUF
2361	010360	017701	170624	8\$:	MOV	3CSR, R1	:SAVE THE CSR
2362	010364	022701	100405		CMP	#100405,R1	:CHECK THE CSR
2363	010370	001401			BEQ	.+4	:BRANCH IF OK
2364	010372	104001			HLT+1		:DONE DIDN'T CLEAR OR OTHER CSR ERROR
2365							:R1 = CONTENTS OF CSR
2366	010374	005077	170614		CLR	3TCR	:CLEAR TCR
2367	010400	005077	170604		CLR	3CSR	:CLEAR CSR
2368	010404	104400			SCOPE		

 :TEST 45: TEST THAT LINE 11 CAN TRANSMIT AND
 :RECEIVE A CHARACTER. (377)
 :1\$: CHECKS THAT DONE SETS IN REASONABLE TIME.
 :2\$: CHECKS THAT CHAR PRESENT IS IN FI/FO
 :3\$: CHECKS THAT NO ERRORS IN FI/FO
 :4\$: CHECKS THAT RIGHT LINE # (11) IN FI/FO
 :5\$: CHECKS FOR RIGHT CHARACTER LENGTH
 :6\$: CHECKS THAT CORRECT DATA WAS RECEIVED
 :7\$: CHECKS THAT CHARACTER PRESENT CLEARS
 :8\$: CHECKS THAT DONE CLEARS
 :PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES

INITIALIZE
 DEVICE"CSR"REGISTER

TST45: JSR PC, 3#INITD SET:
 :BIT2 = MAINTENANCE
 :BIT3 = CLEAR MOS
 :BIT8 = MASTER XMTR SCAN ENB
 :WAIT FOR MOS TO CLEAR
 SET:
 :BIT0 = RECEIVER ENABLE

2394	010412	052777	004000	170574	BIS	#BIT11, STCR	:SET XMTR CONTROL BIT, LINE 11
2395	010420	017701	170564		MOV	STCR, R1	:WAIT FOR XMTR READY
2396	010424	100375			BPL	LOP45	
2397	010426	012777	000377	170562	MOV	#377, STBUF	:SEND A RUBOUT
2398	010434	005000			CLR	R0	:CLEAR COUNTER
2399	010436	105305			1\$: DECB	R5	:SHORT WAIT LOOP
2400	010440	001376			SNE	1\$	
2401	010442	105777	170542		TSTB	STCR	:WAIT FOR DONE
2402	010446	100405			BMI	2\$	:BRANCH WHEN DONE
2403	010450	005200			INC	R0	:TIME COUNTER
2404	010452	001371			BNE	1\$	:BRANCH IF NOT TIME-OUT
2405	010454	017701	170530		MOV	STCR, R1	:SAVE CSR
2406	010460	104001			HLT+1		:DONE NEVER CAME UP
2407							:R1 = CONTENTS OF CSR
2408	010462	050037	001302		2\$: BIS	R0, TIMER	:SAVE TIMER
2409	010466	017701	170520		MOV	STBUF, R1	:READ THE FI/FO
2410	010472	100401			BMI	.+4	:BRANCH IF CHARACTER READY
2411	010474	104001			HLT+1		:CHARACTER READY DIDN'T SET
2412							:R1 = CONTENTS OF RBUF
2413	010476	032701	070000		3\$: BIT	#70000, R1	:CHECK FOR ERROR BITS
2414	010502	001401			BEQ	.+4	:BRANCH IF NONE
2415	010504	104001			HLT+1		:ERROR IN RECEIVED CHAR
2416							:R1 = CONTENTS OF RBUF
2417							:BIT14=UART OVERRUN
2418							:BIT13=FRAMING ERROR
2419							:BIT12=PARITY ERROR
2420	010506	010102			4\$: MOV	R1, R2	:DUPLICATE DATA WORD
2421	010510	042702	170377		BIC	#170377, R2	:MASK LINE#
2422	010514	000302			SWAB	R2	:LINE # IN LOW BYTE
2423	010516	122702	000013		CMPB	#11., R2	:CHECK LINE #
2424	010522	001401			BEQ	.+4	:BRANCH IF OK
2425	010524	104001			HLT+1		:WRONG LINE # RECEIVED
2426							:R1 = CONTENTS OF RBUF
2427							:BITS8-11 = LINE #
2428	010526	117702	170554		5\$: MOVB	STJLEN, R2	:GET MASK OF CHARACTER
2429	010532	130201			BITB	R2, R1	:CHECK CHAR LENGTH.
2430	010534	001401			BEQ	.+4	:BRANCH IF OK
2431	010536	104002			HLT+2		:WRONG CHARACTER LENGTH
2432							:R1=DATA FROM FI/FO
2433							:R2=MASK (BITS SET NOT EXPECTED)
2434	010540	105102			6\$: COMB	R2	:REVERSE THE MASK
2435	010542	120102			CMPB	R1, R2	:CHECK THE ACTUAL DATA
2436	010544	001401			BEQ	.+4	:BRANCH IF OK
2437	010546	104002			HLT+2		:WRONG CHAR LEN OR DATA ERROR
2438							:R1=DATA FROM FI/FO (COMPLETE WORD)
2439							:R2=DATA (LOW BYTE) EXPECTED
2440	010550	017701	170436		7\$: MOV	STBUF, R1	:READ FI/FO
2441	010554	100001			BPL	.+4	:BRANCH IF CHAR PRESENT NOT SET
2442	010556	104001			HLT+1		:CHARACTER PRESENT STAYED SET
2443							:R1 = CONTENTS OF RBUF
2444	010560	017701	170424		8\$: MOV	STCR, R1	:SAVE THE CSR
2445	010564	022701	100405		CMP	#100405, R1	:CHECK THE CSR
2446	010570	001401			BEQ	.+4	:BRANCH IF OK
2447	010572	104001			HLT+1		:DONE DIDN'T CLEAR OR OTHER CSR ERROR
2448							:R1 = CONTENTS OF CSR
2449	010574	005177	170414		CLR	STCR	:CLEAR TCR

N04

MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS MACY11 27(732) 21-SEP-76 13:43 PAGE 50
DZDJA.D.P11 TST45: TEST ALL OF LINE 11 TRANSMIT AND RECEIVE LOGIC

010600 005077 170404
010604 104400
010606 005237 001306
010612 012737 010620 015770

CLR QCSR ;CLEAR CSR
SCOPE
INC DJLEN
MOV #.+6, LAD

TEST 46: TEST THAT LINE 12 CAN TRANSMIT AND
RECEIVE A CHARACTER. (377)
1\$: CHECKS THAT DONE SETS IN REASONABLE TIME.
2\$: CHECKS THAT CHAR PRESENT IS IN FI/FO
3\$: CHECKS THAT NO ERRORS IN FI/FO
4\$: CHECKS THAT RIGHT LINE # (12) IN FI/FO
5\$: CHECKS FOR RIGHT CHARACTER LENGTH
6\$: CHECKS THAT CORRECT DATA WAS RECEIVED
7\$: CHECKS THAT CHARACTER PRESENT CLEARS
8\$: CHECKS THAT DONE CLEARS
PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD DC3 SERIES

INITIALIZE
DEVICE"CSR"REGISTER

010620 004737 015566

TST46: JSR PC, Q#INITD

SET:
;BIT2 = MAINTENANCE
;BIT3 = CLEAR MOS
;BITS = MASTER XMTR SCAN ENB

;WAIT FOR MOS TO CLEAR

SET:
;BIT0 = RECEIVER ENABLE

010624 052777 010000 170362

LOP46: BIS #BIT12, QCSR ;SET XMTR CONTROL BIT, LINE 12

;WAIT FOR XMTR READY

010632 017701 170352

MOV QCSR, R1

010636 100375

BPL LOP46

010640 012777 000377 170350

MOV #377, QTBUF ;SEND A RUBOUT

010646 005000

CLR R0 ;CLEAR COUNTER

010650 105305

1\$: DECB R5 ;SHORT WAIT LOOP

010652 001376

BNE 1\$

010654 105777 170320

TSTB QCSR ;WAIT FOR DONE

010660 100405

BMI 2\$;BRANCH WHEN DONE

010662 005200

INC R0 ;TIME COUNTER

010664 001371

BNE 1\$;BRANCH IF NOT TIME-OUT

010666 017701 170316

MOV QCSR, R1 ;SAVE CSR

010672 104001

HLT+1 ;DONE NEVER CAME UP

010674 050037 001302

2\$: BIS R0, TIMER ;SAVE TIMER

010700 017701 170306

MOV QRBUFF, R1 ;READ THE FI/FO

010704 100401

BMI .+4 ;BRANCH IF CHARACTER READY

010706 104001

HLT+1 ;CHARACTER READY DIDN'T SET

010710 032701 070000

3\$: BIT #70000, R1 ;R1 = CONTENTS OF RBUF

010714 001401

BEQ .+4 ;CHECK FOR ERROR BITS

010716 104001

HLT+1 ;BRANCH IF NONE

010720 010102

4\$: MOV R1, R2 ;ERROR IN RECEIVED CHAR

;R1 = CONTENTS OF RBUF

;BIT14=UART OVERRUN

;BIT13=FRAMING ERROR

;BIT12=PARITY ERROR

;DUPLICATE DATA WORD

12 TRANSMIT AND RECEIVE LOGIC

[illegible]

```

: MASK LINE *
: LINE * IN LOW BYTE
: CHECK LINE *
: BRANCH IF OK
: WRONG LINE * RECEIVED
: R1 = CONTENTS OF RBUF
: BITS8-11 = LINE *
: GET MASK OF CHARACTER
: CHECK CHAR LENGTH.
: BRANCH IF OK
: WRONG CHARACTER LENGTH
: R1=DATA FROM FI/FO
: R2=MASK (BITS SET NOT EXPECTED)
: REVERSE THE MASK
: CHECK THE ACTUAL DATA
: BRANCH IF OK
: WRONG CHAR LEN OR DATA ERROR
: R1=DATA FROM FI/FO (COMPLETE WORD)
: R2=DATA (LOW BYTE) EXPECTED
: READ FI/FO
: BRANCH IF CHAR PRESENT NOT SET
: CHARACTER PRESENT STAYED SET
: R1 = CONTENTS OF RBUF
: SOLVE THE CSR
: CHECK THE CSR
: BRANCH IF OK
: DONE DIDN'T CLEAR OR OTHER CSR ERROR
: R1 = CONTENTS OF CSR
: CLEAR TOR
: CLEAR CSR

```

```

*****
TEST 17:  LINE 13 CAN TRANSMIT AND
          CHECKED. (OK)
          THAT ONE UNIT IN REASONABLE TIME.
          CHECKED. (OK)
          THAT NO ERRORS IN F1/F0
          CHECKED. (OK)
          THAT RIGHT LINE # (13) IN F1/F0
          FOR A RIGHT CHARACTER LENGTH
          CHECKED. (OK)
          THAT CORRECT DATA WAS RECEIVED
          CHECKED. (OK)
          THAT CHARACTER UNIT CLEARS
          CHECKED. (OK)
*****
PROBABLY FAULTY LOGIC:  M7205 (C2-7) ALL; M7276 ALL; UNIT CARD BUS SERIES
*****

```

```

:                               INITIALIZE
:                               DEVICE"CSR"REGISTER
19147:  ISR      PC.      @INITD

```

000000 = MAINTENANCE
000000 = CLEVER MOOSE
000000 = MASTER XMTB SCAN ENG

000000 = BUCKLE UP PLEASE
000000 = BUCKLE UP PLEASE

D05

MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53

MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53

MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53

MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53

MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53
 MACY11 271732) 21-SEP-76 13:43 PAGE 53

011010 005077 167772
 011010 104400

CLR 0CSR :CLEAR CSR
 SCOPE

 :TEST 50: TEST THAT LINE 14 CAN TRANSMIT AND
 :RECEIVE A CHARACTER. (377)
 :1S: CHECKS THAT DONE SETS IN REASONABLE TIME.
 :2S: CHECKS THAT CHAR PRESENT IS IN FI/FO
 :3S: CHECKS THAT NO ERRORS IN FI/FO
 :4S: CHECKS THAT RIGHT LINE # (14) IN FI/FO
 :5S: CHECKS FOR RIGHT CHARACTER LENGTH
 :6S: CHECKS THAT CORRECT DATA WAS RECEIVED
 :7S: CHECKS THAT CHARACTER PRESENT CLEARS
 :8S: CHECKS THAT DONE CLEARS
 :PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES

INITIALIZE
 DEVICE"CSR"REGISTER

011000 004737 015566

:TEST0: JSR PC, 0*INIT0 :SET:
 :BIT2 = MAINTENANCE
 :BIT3 = CLEAR MOS
 :BIT8 = MASTER XMTR SCAN ENB

:WAIT FOR MOS TO CLEAR

:SET:
 :BIT0 = RECEIVER_ENABLE

000000 000000 040000 167762
 000000 000000 167752
 000000 000000 000377 167750
 000000 000000 167730
 000000 000000 167716

:LOP50: BIS #BIT14, 0CSR :SET XMTR CONTROL BIT, LINE 14
 :MOV 0CSR, R1 :WAIT FOR XMTR READY
 :BPL 0CSR :SEND A RUBOUT
 :MOV #377, 0RBUF :CLEAR COUNTER
 :CLR 0R0 :SHORT WAIT LOOP
 :1S: DEC0 R5 :WAIT FOR DONE
 :JNE 1S :BRANCH WHEN DONE
 :TSTB 0CSR :TIME COUNTER
 :BNC 1S :BRANCH IF NOT TIME-OUT
 :BNC 1S :SAVE CSR
 :MOV 0CSR, R1 :DONE NEVER CAME UP
 :HLT+1 :R1 = CONTENTS OF CSR
 :2S: BIS 0R0, TIMER :SAVE TIMER
 :MOV 0RBUF, R1 :READ THE FI/FO
 :BNC 1S :BRANCH IF CHARACTER READY
 :BNC 1S :CHARACTER READY DIDN'T SET
 :BNC 1S :R1 = CONTENTS OF RBUF
 :BNC 1S :CHECK FOR ERROR BITS
 :BNC 1S :BRANCH IF NONE
 :BNC 1S :ERROR IN RECEIVED CHAR
 :BNC 1S :R1 = CONTENTS OF RBUF
 :BNC 1S :BIT14=UART OVERRUN
 :BNC 1S :BIT13=FRAMING ERROR
 :BNC 1S :BIT12=PARITY ERROR
 :BNC 1S :DUPLICATE DATA WORD
 :BNC 1S :MASK LINE#
 :BNC 1S :LINE # IN LOW BYTE

000000 000000 001300
 000000 000000 167700
 000000 000000 104001
 000000 000000 104001

:3S: BIS 0R0, TIMER :SAVE TIMER
 :MOV 0RBUF, R1 :READ THE FI/FO
 :BNC 1S :BRANCH IF CHARACTER READY
 :BNC 1S :CHARACTER READY DIDN'T SET
 :BNC 1S :R1 = CONTENTS OF RBUF
 :BNC 1S :CHECK FOR ERROR BITS
 :BNC 1S :BRANCH IF NONE
 :BNC 1S :ERROR IN RECEIVED CHAR
 :BNC 1S :R1 = CONTENTS OF RBUF
 :BNC 1S :BIT14=UART OVERRUN
 :BNC 1S :BIT13=FRAMING ERROR
 :BNC 1S :BIT12=PARITY ERROR
 :BNC 1S :DUPLICATE DATA WORD
 :BNC 1S :MASK LINE#
 :BNC 1S :LINE # IN LOW BYTE

000000 000000 070000
 000000 000000 001501
 000000 000000 104001

:3S: BIS 0R0, TIMER :SAVE TIMER
 :MOV 0RBUF, R1 :READ THE FI/FO
 :BNC 1S :BRANCH IF CHARACTER READY
 :BNC 1S :CHARACTER READY DIDN'T SET
 :BNC 1S :R1 = CONTENTS OF RBUF
 :BNC 1S :CHECK FOR ERROR BITS
 :BNC 1S :BRANCH IF NONE
 :BNC 1S :ERROR IN RECEIVED CHAR
 :BNC 1S :R1 = CONTENTS OF RBUF
 :BNC 1S :BIT14=UART OVERRUN
 :BNC 1S :BIT13=FRAMING ERROR
 :BNC 1S :BIT12=PARITY ERROR
 :BNC 1S :DUPLICATE DATA WORD
 :BNC 1S :MASK LINE#
 :BNC 1S :LINE # IN LOW BYTE

000000 000000 010102
 000000 000000 042700
 000000 000000 170377

:4S: MOV R1, R2
 :BIC #170377, R2
 :SWAB R2
 :LINE # IN LOW BYTE

E05

MAINDEC-11-02DJA-D
D2DJA.D.F11

TSTSD:

DJ11 LOGIC TESTS

MACY11 27(732) 21-SEP-76 13:43 PAGE 54

TEST ALL OF LINE 14 TRANSMIT AND RECEIVE LOGIC

011330 122702 000016
011334 001401
011336 104001CMPB #14., R2
BEQ .+4
HLT+1:CHECK LINE #
:BRANCH IF OK
:WRONG LINE # RECEIVED
:R1 = CONTENTS OF RBUF
:BITS8-11 = LINE #
:GET MASK OF CHARACTER
:CHECK CHAR LENGTH.
:BRANCH IF OK
:WRONG CHARACTER LENGTH011340 117702 167742
011344 130301
011346 001401
011350 10400253: MOVB DDJLEN, R2
BITB R2, R1
BEQ .+4
HLT+2:R1=DATA FROM FI/FO
:R2=MASK (BITS SET NOT EXPECTED)
:REVERSE THE MASK
:CHECK THE ACTUAL DATA
:BRANCH IF OK
:WRONG CHAR LEN OR DATA ERROR
:R1=DATA FROM FI/FO (COMPLETE WORD)
:R2=DATA (LOW BYTE) EXPECTED011352 106100
011354 100001
011356 001401
011360 10400263: COMB R2
CMPB R1, R2
BEQ .+4
HLT+2:READ FI/FO
:BRANCH IF CHAR PRESENT NOT SET
:CHARACTER PRESENT STAYED SET
:R1 = CONTENTS OF RBUF011362 017701 167824
011364 100001
011366 10400173: MOV RBUF, R1
BPL .+4
HLT+1:SAVE THE CSR
:CHECK THE CSR
:BRANCH IF OK
:DONE DIDN'T CLEAR OR OTHER CSR ERROR
:R1 = CONTENTS OF CSR
:CLEAR TCR
:CLEAR CSR011368 017701 167812
011370 000001
011372 100405
011374 10400183: MOV RCSR, R1
CMP #100405, R1
BEQ .+4
HLT+1011376 000001
011378 000001
011380 104400CLR RCSR
CLR RCSR
SCOPE

:TEST S1: TEST THAT LINE 15 CAN TRANSMIT AND
RECEIVE A CHARACTER. (377)
:13: CHECKS THAT DONE SETS IN REASONABLE TIME.
:23: CHECKS THAT CHAR PRESENT IS IN FI/FO
:33: CHECKS THAT NO ERRORS IN FI/FO
:43: CHECKS THAT RIGHT LINE # (15) IN FI/FO
:53: CHECKS FOR RIGHT CHARACTER LENGTH
:63: CHECKS THAT CORRECT DATA WAS RECEIVED
:73: CHECKS THAT CHARACTER PRESENT CLEARS
:83: CHECKS THAT DONE CLEARS
:PROBABLE FAULTY LOGIC: M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES

: INITIALIZE
: DEVICE"CSR"REGISTER

011420 004737 015566

TST51: JSR PC, @INITD

SET:
:BIT2 = MAINTENANCE
:BIT3 = CLEAR MOS
:BIT8 = MASTER XMTR SCAN ENB

:WAIT FOR MOS TO CLEAR

SET:
:BIT0 = RECEIVER ENABLE

011424 052777 100000 167562
011432 017701 167552LOP51: BIS #BIT15, RCSR
MOV RCSR, R1

:SET XMTR CONTROL BIT, LINE 15
:WAIT FOR XMTR READY

222JAP. P. 11

TST51:

TEST ALL OF LINE IS TRANSMIT AND RECEIVE LOGIC

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

G05

MAINDEC-11-DZDJA-D
DZDJA.D.P11

TST51:

DJ11 LOGIC TESTS

MACY11 27(732) 21-SEP-76 13:43 PAGE 56

TEST ALL OF LINE 15 TRANSMIT AND RECEIVE LOGIC

011620	162737	000003	001306	SUB	#3,	DJLEN	
011626	005237	001302		INC	TIMER		:WORSE CASE TIME, ONE CHARACTER
011632	013700	001302		MOV	TIMER, R0		:DUP TIMER
011638	006200			ASR	R0		: /2
011644	006200			ASR	R0		: /4
011644	005200			INC	R0		: +1
011644	006137	001302		ROL	TIMER		: TIMER * 2
011650	060037	001302		ADD	R0,	TIMER	: 2.25 TIMES ONE CHARACTER TIME
011654	012737	011652	015770	MOV	#.46,	LAD	: RESET LOOP ADDRESS

 :TEST 52: TEST THAT CHARACTER PRESENT (BIT15) OF RBUF
 : IS CLEARED (BY CLEAR MOS).
 : PROBABLE FAULTY LOGIC: M7285 (D2-8) E17,E14,E15; M7279; M7280

INITIALIZE

011662	004737	015566		:TEST 52: JSR	PC,2*INITD		
				:SET:			
					BIT2 = MAINTENANCE		
					BIT3 = CLEAR MOS		
					BIT8 = TRANS SCAN ENABLE		
					WAIT FOR BIT4 = MOS CLEAR		
				:SET:			
011666	012777	000001	167320		BIT0 = RECEIVER ENABLE		
011674	012704	000004		MOV	#BIT0, 2TOR		:TRANS CONTROL, LINED
011680	017701	167304		MOV	#4,R4		:SET UP COUNTER
011686	100275		18:	MOV	2CSR,R1		:WAIT FOR TRANS READY
011692	010477	167304		BPL	18		
011698	009204			MOV	R4,2TBUF		:TRANSMIT COUNT
011704	001371			DEC	R4		:COUNT DOWN
				BNE	18		
011716	105777	167266	28:	TSTB	2CSR		:MAKE SURE DONE IS SET
011722	100275			BPL	28		
011728	052777	000010	167266	BIT	#BIT3,2CSR		:CLEAR MOS
011734	032777	000020	167260	BIT	#BIT4,2CSR		:WAIT FOR CLAMOS TO FINISH
011740	001374			BNE	38		
011746	017701	167244		MOV	2RBUF,R1		:CHECK RBUF AND SAVE
011752	100001			BPL	.+4		
011758	104001			HLT	+1		
011764	017701	167232		MOV	2CSR,R1		:SAVE CSR
011770	022701	100405		CMP	#100405,R1		:CHECK CSR
011776	001401			BEG	.+4		:BRANCH IF OK
011782	104001			HLT	+1		:CSR ERROR, POSSIBILITIES:
							:(1) DONE DIDN'T CLEAR
							:(2) TRANSMITTER UART DIDN'T CLR.
							:R1=CONTENTS OF CSR
011766	013700	001302		MOV	TIMER, R0		:SET UP TIMER
011772	106205		48:	DECB	R5		:SHORT WAIT LOOP
011778	001375			BNE	48		

H05

MA: NOEC-11-0200A-0
020000.F11

TST53: TEST CLEAR MOS

MAY11 27(732) 21-SEP-76 13:43 PAGE 57

011773	017701	167206	MOV	BCSR, R1	:SAVE CSR
011800	105701		TSTB	R1	:CHECK FOR DONE
011800	100001		BPL	.+4	:BRANCH IF OK
011800	104001		HLT+1		:DONE CAME UP!
					:MOS MUST NOT HAVE CLEARED
011801	005300		DEC	RD	:TIMER COUNT
011801	001367		BNE	4\$	
011801	002701	100405	CMP	#100405, R1	:CHECK CSR
011801	001401		BEG	.+4	:BRANCH IF OK
011801	104001		HLT+1		:CSR ERROR
011802	017701	167162	MOV	RBUFF, R1	:CHECK RBUFF
011802	100001		BPL	.+4	:BRANCH IF OK
011802	104001		HLT+1		:RBUFF NOT EMPTY!
011803	005077	167154	CLR	BTOR	
011803	005077	167144	CLR	BCSR	
012044	104400		SCOPE		

:TEST 53: TEST THAT TRANSMITTER READY CLEARS WHEN TBUF IS LOADED
:NOTE: DUE TO THE DOUBLE BUFFERING BY THE UART, TWO CHARACTERS
:MUST BE LOADED TO INSURE SEEING TRANSMITTER READY CLEAR
:PROBABLE FAULTY LOGIC: M7285 (D2-6) E39, E23, E49

INITIALIZE
DEVICE"CSR"REGISTER

012046 004737 015556

TST53: JSR PC, @INITO
SET:
:BIT2 = MAINTENANCE
:BIT3 = CLEAR MOS
:BIT8 = MASTER XMTR SCAN ENB

:WAIT FOR MOS TO CLEAR

SET:
:BIT0 = RECEIVER ENABLE

012052	052777	000001	167134
012060	017701	167124	
012064	100375		
012066	012777	000001	167122
012074	017701	167110	
012100	100375		
012102	012777	000002	167106
012110	017701	167074	
012114	100001		
012116	104001		

1\$:	BIS	#BIT0,	BTOR	:TRANS CONTROL, LINE 0
	MOV	BCSR,	R1	:WAIT FOR XMTR READY
	BPL	1\$		
2\$:	MOV	#1,	BTBUF	:TRANSMIT A 1
	MOV	BCSR,	R1	:WAIT FOR XMTR READY
	BPL	2\$		
	MOV	#2,	BTBUF	:TRANSMIT A 2
	MOV	BCSR,	R1	:CHECK FOR XMTR READY
	BPL	.+4		:BRANCH IF XMTR READY CLEARED
	HLT+1			:TRANSMITTER READY FAILED TO CLEAR
				:R1 = CONTENTS OF CSR
3\$:	MOV	TIMER,	RD	:SET UP TIMER
	DECB	R5		:SHORT WAIT LOOP
	BNE	3\$		
	MOV	BCSR,	R1	:SAVE CSR FOR THE RECORD
	BR	.+2		:NOP FOR TIMING

012120	013700	001302	
012124	105305		
012126	001376		
012130	017701	167054	
012134	000400		

MAINDEC-11-DZDJA-D
DZDJA.D.P11TST53: DJ11 LOGIC TESTS
TEST TRANSMIT READY

MACY11 27(732) 21-SEP-76 13:43 PAGE 58

```

2898 012136 005300 DEC R0 ;TIMER COUNT
2899 012140 001371 BNE 3$ ;BRANCH IF MORE TIME
2900
2901 012142 022701 100605 CMP #100605,R1 ;CHECK CSR
2902 012146 001401 BEQ .+4 ;BRANCH IF OK
2903 012150 104001 HLT+1 ;DONE DIDN'T SET OR OTHER CSR ERROR
2904
2905 012152 017701 167034 MOV 3RBUF, R1 ;R1 = CONTENTS OF CSR
2906 012156 100401 SMI .+4 ;CHECK RBUF FOR CHAR PRESENT
2907 012160 104001 HLT+1 ;BRANCH IF CHAR PRESENT
2908 ;CHAR PRESENT MISSING
2909 ;R1 = CONTENTS OF RBUF
2910 012162 022701 100001 CMP #100001,R1 ;CHECK THE DATA
2911 012166 001401 BEQ .+4 ;BRANCH IF OK
2912 012170 104001 HLT+1 ;RECEIVER ERROR
2913 ;R1 = CNTENTS OF RBUF
2914 012172 017701 167014 MOV 3RBUF, R1 ;CHECK RBUF FOR SECOND CHAR
2915 012176 100401 SMI .+4 ;BRANCH IF CHAR PRESENT
2916 012200 104001 HLT+1 ;CHAR PRESENT MISSING
2917 ;R1 = CONTENTS OF RBUF
2918 012202 022701 100002 CMP #100002,R1 ;CHECK THE DATA
2919 012206 001401 BEQ .+4 ;BRANCH IF OK
2920 012210 104001 HLT+1 ;RECEIVER ERROR
2921 ;R1 = CNTENTS OF RBUF
2922 012212 017701 166774 MOV 3RBUF, R1 ;CHECK FOR NO MORE CHARACTERS
2923 012216 100001 BPL .+4 ;BRANCH IF CHAR PRESENT CLEARED
2924 012220 104001 HLT+1 ;CHAR PRESENT NOT CLEAR!
2925 ;R1 = CONTENTS OF RBUF
2926 012222 017701 166762 MOV 3CSR, R1 ;SAVE CSR
2927 012226 022701 100405 CMP #100405,R1 ;CHECK CSR
2928 012232 001401 BEQ .+4 ;BRANCH IF OK
2929 012234 104001 HLT+1 ;CSR ERROR
2930 ;R1 = CONTENTS OF CSR
2931 012236 005077 166752 CLR 3TCR ;CLEAR TCR
2932 012242 005077 166742 CLR 3CSR ;CLEAR CSR
2933 012246 104400 SCOPE

```

```

*****
;TEST 54: TEST THAT RECEIVER ENABLE ON A 0 INHIBITS DONE
; AND CHARACTER PRESENT.
;PROBABLE FAULTY LOGIC: M7285 (D2-7) E32, E15
*****

```

```

012250 012777 000414 166732 TST54: MOV #414, 3CSR ;BIT2 = MAINTENANCE
012256 032777 000020 166724 10$: BIT #BIT4, 3CSR ;BIT3 = CLEAR MOS
012264 001374 000000 166720 1$: BIS #BIT0, 3TCR ;BIT8 = MASTER TRAN SCAN ENB
012266 052777 000000 166710 1$: MOV 3CSR, R1 ;WAIT FOR MOS TO CLEAR
012274 017701 000252 166706 2$: MOV #252, 3TBUF ;SET XMTR CONTROL BIT, LINED
012300 100375 001302 2$: MOV TIMER, R0 ;WAIT FOR XMTR READY
012302 012777 000252 166706 2$: MOV #252, 3TBUF ;SEND AN "*"
012310 013700 001302 2$: MOV TIMER, R0 ;SET UP TIMER
012314 105305 2$: DECB R5 ;SHORT WAIT LOOP
012316 001376 2$: BNE 2$
012320 017701 166664 MOV 3CSR, R1 ;SAVE CSR FOR TYPING

```


MAINDEC-11-DZDJA-D
DZDJA.D.P11

TST54:

DJ11 LOGIC TESTS
TEST RECEIVER ENABLE

MACY11 27(732) 21-SEP-76 13:43 PAGE 59

2954	012324	105701			TSTB	R1			;CHECK FOR DONE
2955	012326	100002			BPL	3\$			;BRANCH IF NOT SET
2956	012330	104001			HLT+1				;DONE SET WHEN RCV ENB CLR
2957									;R1=CONTENTS OF CSR
2958	012332	000402			BR	4\$			
2959	012334	005300		3\$:	DEC	RO			;TIMER COUNT
2960	012336	001366			BNE	2\$			;BRANCH IF MORE TIMER
2961									
2962	012340	017701	166646		MOV	3RBUF,	R1		;CHECK AND SAVE FI/FO
2963	012344	100001			BPL	.+4			;BRANCH IF OK
2964	012346	104001			HLT+1				;CHARACTER PRESENT IN FI/FO
2965									;R1=DATA FROM FI/FO
2966	012350	005277	166634		INC	3CSR			;SET RECEIVER ENABLE
2967	012354	017701	166630		MOV	3CSR,	R1		;SAVE CSR
2968	012360	105701		5\$:	TSTB	R1			;CHECK FOR DONE
2969	012362	100403			BMI	6\$			;BRANCH IF OK
2970	012364	105300			DECB	RO			;SHORT TIMER
2971	012366	001372			BNE	5\$			
2972	012370	104001			HLT+1				;DONE DIDN'T COME UP WHEN RECEIVER ENABLED
2973									;UART SHOULD HAVE HELD A CHARACTER
2974									;R1 = CONTENTS OF CSR
2975	012372	017701	166614		MOV	3RBUF,	R1		;CHECK FOR CHARACTER PRESENT
2976	012376	100401		6\$:	BMI	.+4			;BRANCH IF OK
2977	012400	104001			HLT+1				;CHARACTER PRESENT MISSING
2978									;R1 = CONTENTS OF RBUF
2979	012402	005077	166606		CLR	3TCR			;CLR TRANS CONTROL REG
2980									
2981	012406	104400			SCOPE				
2982									
2983									
2984									
2985									
2986									
2987									
2988	012410	004737	015556		TST55:	JSR	PC,	3#INITC	;INITIALIZE
2989									;BIT1 = HALF DUPLEX
2990									;BIT2 = MAINTENANCE
2991									;BIT3 = CLEAR MOS
2992									;BIT8 = MASTER TRAN SCAN ENB
2993									;WAIT FOR MOS TO CLEAR
2994									;BIT0 = RECEIVER ENABLE
2995	012414	012777	000001	166572	MOV	#BIT0,	3TCR		;SET XMTR CONTROL BIT, LINED
2996	012422	017701	166562		MOV	3CSR,	R1		;WAIT FOR XMTR READY
2997	012426	100375		1\$:	BPL	1\$			
2998	012430	012777	000252	166560	MOV	#252,	3TBUF		;SEND AN "*"
2999	012436	013700	001302		MOV	TIMER,	RO		;SET UP TIMER
3000	012442	105305		2\$:	DECB	R5			;SHORT WAIT LOOP
3001	012444	001376			BNE	2\$			
3002	012446	017701	166536		MOV	3CSR,	R1		;SAVE CSR
3003	012452	105701			TSTB	R1			;CHECK FOR DONE
3004	012454	100002			BPL	3\$			;BRANCH IF NOT SET
3005	012456	104001			HLT+1				;DONE SET WHEN HALF DUPLEX (BIT1) SET
3006									;R1=CONTENTS OF CSR
3007	012460	000402			BR	4\$			
3008	012462	005300		3\$:	DEC	RO			;TIMER COUNT
3009	012464	001366			BNE	2\$			;BRANCH IF MORE TIMER

 :TEST 55: TEST THAT HALF DUPLEX (BIT1) DISABLES THE RECEIVER UARTS.
 :PROBABLE FAULTY LOGIC: M7285 (D2-4) E32, E17, E22, (D2-2) E5, E1
 :*****

K05

 MAINDEC-11-DZDJA-D
 DZDJAD.F11 TST55: DJ11 LOGIC TESTS
 TEST HALF DUPLEX

MACY11 27(732) 21-SEP-76 13:43 PAGE 60

```

3010
3011 012466 017701 166520 4$: MOV 3RBUF, R1 ;CHECK AND SAVE FI/FO
3012 012472 100001 BPL .+4 ;BRANCH IF OK
3013 012474 104001 HLT+1 ;CHARACTER PRESENT IN FI/FO
3014 ;R1=DATA FROM FI/FO
3015 012476 042777 000002 166504 BIC #BIT1, 3CSR ;CLEAR HALF DUPLEX BIT
3016 012504 000240 NOP
3017 012506 000240 NOP
3018 012510 000240 NOP
3019 012512 000240 NOP
3020 012514 017701 166472 5$: MOV 3RBUF, R1 ;CHECK FOR CHAR PRESENT
3021 012520 100001 BPL .+4 ;BRANCH IF CHAR NOT PRESENT
3022 012522 104001 HLT+1 ;CHAR PRESENT AFTER H/D CLEARED
3023 ;R1 = CONTENTS OF RBUF
3024 012524 005077 166464 CLR 3TCR ;CLR TRANS CONTROL REG
3025
3026 012530 104400 SCOPE
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065

```

```

*****
:TEST 56: TEST THAT RECEIVER INTERRUPT DOES NOT OCCUR AT LEVEL 5
:PROBABLE FAULTY LOGIC: M7821 WIRING, PROPER PRIORITY CHIP
*****

```

```

3034 012532 012777 012620 166460 TST56: MOV #ISR56, 3RCVVEC ;SET UP XMTR INTERRUPT VECTOR
3035 012540 012777 000340 166454 MOV #340, 3RCVLVL ;AT LEVEL 7
3036 012546 042737 000340 177776 BIC #340, 3#PS ;CLEAR PS LEVEL
3037 012554 052737 000240 177776 BIS #240, 3#PS ;SET PS TO LEVEL 5
3038 012552 004737 015546 JSR PC, 3#INITB ;SET:
3039 ;BIT2 = MAINTENANCE
3040 ;BIT3 = CLEAR MOS
3041 ;BIT6 = RECEIVER INTERRUPT ENABLE
3042 ;BIT8 = MASTER TRANS SCAN ENABLE
3043 ;WAIT FOR MOS TO CLEAR
3044 ;BIT0 = RECEIVER ENABLE
3045 ;SET TRAN CONTROL BIT, LINE 0
3046 012566 012777 000001 166420 1$: MOV #BIT0, 3TCR ;WAIT
3047 012574 017701 166410 MOV 3CSR, R1
3048 012600 100375 BPL 1$
3049 012602 012777 000025 166406 2$: MOV #25, 3TBUF ;SEND #25
3050 012610 105777 166374 TSTB 3CSR ;WAIT FOR DONE
3051 012614 100375 BPL 2$
3052 012616 000404 BR END56 ;OK, BRANCH IF NO INTERRUPT
3053 012620 104000 ISR56: HLT ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 5
3054 012622 012716 012630 MOV #END56, (SP) ;MOVE NEW RTI ADR ONTO STACK
3055 012626 000002 RTI
3056
3057 012630 017701 166356 END56: MOV 3RBUF, R1 ;READ THE CHARACTER
3058 012634 100401 BMI .+4 ;BRANCH IF CHAR PRESENT
3059 012636 104001 HLT+1 ;CHAR PRESENT MISSING
3060 ;R1 = CONTENTS OF RBUF
3061 012640 022701 100025 CMP #100025, R1 ;CHECK THE DATA
3062 012644 001401 BEQ .+4 ;BRANCH IF OK
3063 012646 104001 HLT+1 ;RECEIVED DATA ERROR
3064 ;R1 = CONTENTS OF RBUF
3065 012650 013777 001222 166342 MOV RCVLVL, 3RCVVEC

```



```

3066 012656 012777 000004 166336 MOV #IOT, @RCVLVL
3067 012664 005077 166324 CLR @TCR
3068 012670 005077 166314 CLR @CSR
3069 012674 042737 000340 177776 BIC #340, @#PS
3070
3071 012702 104400 SCOPE
3072
3073
3074
3075
3076
3077
3078
3079
3080 012704 012777 012774 166306 TST57: MOV #ISR57, @RCVVEC ;SET UP XMTR INTERRUPT VECTOR
3081 012712 012777 000340 166302 MOV #340, @RCVLVL ;AT LEVEL 7
3082 012720 042737 000340 177776 BIC #340, @#PS ;CLEAR PS LEVEL
3083 012726 052737 000200 177776 BIS #200, @#PS ;SET PS TO LEVEL 4
3084 012734 004737 015546 JSR PC, @#INITB ;SET:
3085
3086 ;BIT2 = MAINTENANCE
3087 ;BIT3 = CLEAR MOS
3088 ;BIT6 = RECEIVER INTERRUPT ENABLE
3089 ;BIT8 = MASTER TRANS SCAN ENABLE
3090 ;WAIT FOR MOS TO CLEAR
3091 ;BIT0 = RECEIVER ENABLE
3092 ;SET TRAN CONTROL BIT, LINE 0
3093 ;WAIT
3094 012740 012777 000001 166246 1$: MOV #BIT0, @TCR
3095 012746 017701 166236 1$: MOV @CSR, R1
3096 012752 100375 BPL 1$
3097 012754 012777 000025 166234 2$: MOV #25, @TBUF ;SEND #25
3098 012762 105777 166222 2$: TSTB @CSR ;WAIT FOR DONE
3099 012766 100375 BPL 2$
3100 012770 104000 HLT ;SHOULD HAVE INTERRUPTED AT LEVEL 4
3101 012772 000403 BR END57 ;CONTINUE
3102
3103
3104
3105
3106 012774 012716 013002 ISR57: MOV #END57, (SP) ;MOVE NEW RTI ADR ONTO STACK
3107 013000 000002 RTI
3108
3109
3110
3111 013002 017701 166204 END57: MOV @RBUF, R1 ;READ THE CHARACTER
3112 013006 100401 BMI .+4 ;BRANCH IF CHAR PRESENT
3113 013010 104001 HLT+1 ;CHAR PRESENT MISSING
3114
3115 ;R1 = CONTENTS OF RBUF
3116 013012 022701 100025 CMP #100025, R1 ;CHECK THE DATA
3117 013016 001401 BEQ .+4 ;BRANCH IF OK
3118 013020 104001 HLT+1 ;RECEIVED DATA ERROR
3119 ;R1 = CONTENTS OF RBUF
3120
3121 013022 013777 001222 166170 MOV RCVLVL, @RCVVEC
3122 013030 012777 000004 166164 MOV #IOT, @RCVLVL
3123 013036 005077 166152 CLR @TCR
3124 013042 005077 166142 CLR @CSR
3125 013046 042737 000340 177776 BIC #340, @#PS
3126
3127 013054 104400 SCOPE
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200

```

```

:*****
:TEST 57: TEST THAT RECEIVER INTERRUPT OCCURES AT LEVEL 4
:PROBABLE FAULTY LOGIC: M7821 WIRING, PROPER PRIORITY CHIP
:*****

```

```

:SHOULD HAVE INTERRUPTED AT LEVEL 4
:CONTINUE

```

```

;MOVE NEW RTI ADR ONTO STACK

```

```

;READ THE CHARACTER
;BRANCH IF CHAR PRESENT
;CHAR PRESENT MISSING
;R1 = CONTENTS OF RBUF
;CHECK THE DATA
;BRANCH IF OK
;RECEIVED DATA ERROR
;R1 = CONTENTS OF RBUF

```

```

:*****
:TEST 60: TEST FI/FO OVERRUN
:THE FI/FO BUFFER SHOULD HOLD 64 CHARACTERS.
:*****

```

M05

MAINDEC-11-DZDJA-D
DZDJA0.F11

TST60: DJ11 LOGIC TESTS
TEST FI/FO OVERRUN

MACY11 27(732) 21-SEP-76 13:43 PAGE 62

PROBABLE FAULTY LOGIC: M7285 (D1-7) E32, E17, E22 (D2-2) E5, E1

INITIALIZE
DEVICE"CSR"REGISTER

013056 004737 015566

TST60: JSR PC, @#INITD

SET:
;BIT2 = MAINTENANCE
;BIT3 = CLEAR MOS
;BIT8 = MASTER XMTR SCAN ENB

WAIT FOR MOS TO CLEAR

SET:
;BIT0 = RECEIVER ENABLE

013062 012777 177777 166124

MOV #177777, @TCR

;TRANS CONTROL BIT, ALL LINES
;SET UP COUNTER - 64. CHAR FI/FO BUFF
;SAVE AND WAIT FOR TRANS READY

013070 012700 000100

MOV #100, R0

013074 017701 166110

1\$: MOV @CSR, R1

013100 100375

BPL 1\$

013102 000377 166110

SWAB @TBUF

;TRANSMIT LINE # ON LINE

013106 032701 020000

BIT #BIT13, R1

;CHECK FI/FO OVERRUN

013112 001401

BEQ .+4

;BRANCH IF OK

013114 104001

HLT+1

;FI/FO OVERRUN TOO SOON

013116 005300

DEC R0

;R1=CONTENTS OF CSR

013120 001365

BNE 1\$

;COUNT DOWN

013122 013700 001302

MOV TIMER, R0

;SET UP TIMER

013126 017701 166056

2\$: MOV @CSR, R1

;WAIT FOR XMTR READY

013132 100375

BPL 2\$

013134 105305

3\$: DECB R5

;SHORT WAIT LOOP

013136 001376

BNE 3\$

013140 017701 166044

MOV @CSR, R1

;SAVE CSR FOR THE RECORD

013144 100401

BMI .+4

;BRANCH IF TRANS READY

013146 104001

HLT+1

;TRANS READY MYSTERIOUSLY DISAPPEARED

013150 005300

DEC R0

;R1=CONTENTS OF CSR

013152 001370

BNE 3\$

;TIME 3 CHARACTER LENGTHS

;FI/FO SHOULD NOW BE FULL

013154 105701

TSTB R1

;CHECK THAT DONE IS SET

013156 100401

BMI .+4

;BRANCH IF OK

013160 104001

HLT+1

;DONE DIDN'T COME UP!

013162 022701 100605

CMP #100605, R1

;R1 = CONTENTS OF CSR

013166 001401

BEQ .+4

;CHECK THAT FI/FO NOT OVERRUN

013170 104001

HLT+1

;BRANCH IF OK

;FI/FO OVERRUN SET

;OR SOME OTHER CSR ERROR

;R1=CONTENTS OF CSR

TEST 60A: TEST THAT FI/FO OVERRUN COMES UP WHEN 65TH CHARACTER
IS RECEIVED WITHOUT READING FI/FO

013172 000377 166020

T60A: SWAB

@TBUF

;SEND 65TH CHARACTER

013176 013700 001302

MOV

TIMER, R0

;SET UP TIMER

ADDRESS	HEX	ASCII	DATA	INSTR	COMMENT
3178	013202	105305		11\$: DECB R5	; SHORT WAIT LOOP
3179	013204	001376		BNE 11\$	
3180	013206	017701	165776	MOV JCSR,R1	; SAVE CSR
3181	013212	032701	020000	BIT #BIT13,R1	; CHECK FI/FO OVERRUN
3182	013216	001003		BNE 12\$	; BRANCH WHEN SET
3183	013220	005300		DEC R0	; TIMER
3184	013222	001367		SNE 11\$	; BRANCH IF MORE TIME
3185	013224	104001		HLT+1	; FI/FO OVERRUN DIDN'T COMEUP
3186					; R1 = CONTENTS OF CSR
3187	013226	022701	120605	12\$: CMP #120605,R1	; CHECK TOTAL CSR
3188	013232	001401		BEQ .+4	; BRANCH IF OK
3189	013234	104001		HLT+1	; SOMETHING IN CSR FOULED UP
3190					; R1=CONTENTS OF CSR

```

:*****
:TEST 60B:      TEST THAT READING THE RECEIVER BUFFER CAUSES FI/FO
:                OVERRUN TO CLEAR.
:
:    NOTE:      BECAUSE OF TIMING OF THE FI/FO, FI/FO OVERRUN CAN COME
:                BACK UP AFTER READING ONE CHARACTER, SO A SECOND MUST
:                BE READ TO INSURE THAT FI/FO OVERRUN IS CLEAR.
:*****

```

Address	Hex	Hex	Hex	Label	Assembly	Comments
3200	013236	012700	000002	T60B:	MOV #2, R0	;SET UP COUNTER - 2 CHARACTERS
3201	013242	017701	165744	21\$:	MOV RBUF, R1	;CHECK AND SAVE FIRST CHAR IN FI/FO
3202	013246	100401			BMI .+4	;BRANCH IF CHAR PRESENT
3203	013250	104001			HLT+1	;CHARACTER PRESENT GONE!
3204						
3205	013252	032701	070000		BIT #070000,R1	;CHECK RECEIVER ERRORS
3206	013256	001401			BEQ .+4	;BRANCH IF OK
3207	013260	104001			HLT+1	;RECEIVER ERROR
3208						;R1=CONTENTS OF RBUF
3209						;BIT14=UART OVERRUN
3210						;BIT13=FRAMMING ERROR
3211						;BIT12=PARITY ERROR
3212						
3213	013262	010102			MOV R1, R2	;PUT LINE # IN R2
3214	013264	000302			SWAB R2	
3215	013266	042702	177760		BIC #177760,R2	
3216	013272	120102			CMFB R1, R2	;CHECK DATA (=LINE#)
3217	013274	001401			BEQ .+4	;BRANCH IF OK
3218	013276	104001			HLT+1	;WRONG DATA RECEIVED
3219						;R1=FI/FO DATA
3220						; (DATA SHOULD=LINE#)
3221	013300	005300		22\$:	DEC R0	;COUNT CHARACTERS
3222	013302	001403			BEQ 24\$	;BRANCH WHEN DONE
3223	013304	105305		23\$:	DECB R5	;SHORT WAIT LOOP - GIVE FI/FO TIME
3224	013306	001376			BNE 23\$	
3225	013310	000754			BR 21\$	;GO READ ANOTHER
3226						
3227	013312	017701	165672	24\$:	MOV RCSR, R1	;SAVE CSR
3228	013316	022701	100605		CMP #100605,R1	;CHECK THAT FI/FO OVERRUN CLEARED
3229	013322	001401			BEQ .+4	;BRANCH IF OK
3230	013324	104001			HLT+1	;FI/FO OVERRUN DIDN'T CLR
3231						;OR SOMEOTHER CSR PROBLEM
3232						;R1=CONTENTS OF CSR

TEST 600: TEST THAT FI/FO OVERRUN INTERRUPT
DOESN'T OCCUR WHEN THE PROCESSOR IS AT LEVEL 5

TEST 600: TEST THAT FI/FO OVERRUN INTERRUPT
DOESN'T OCCUR WHEN THE PROCESSOR IS AT LEVEL 5

MAY11 27(732) 21-SEP-76 13:43 PAGE 64

TEST 600: TEST THAT FI/FO OVERRUN INTERRUPT
DOESN'T OCCUR WHEN THE PROCESSOR IS AT LEVEL 5

```

1600:      MOV     R0, #0          ; CLEAR PSW
          MOV     R1, #0          ; SET PROCESSOR TO LEVEL 5
          MOV     R2, #0          ; SET UP RECEIVER INTERRUPT, VEC
          MOV     R3, #0          ; SET STATUS ENABLE
          MOV     R4, #0          ; SAVE CSR
          MOV     R5, #0          ; CHECK CSR
          MOV     R6, #0          ; CSR ERROR
          MOV     R7, #0          ; R1=CONTENTS OF CSR
          MOV     R8, #0          ; SEND LINE #
          MOV     R9, #0          ; WAIT FOR TRANSMITTER READY
          MOV     R10, #0         ; SEND LINE #
          MOV     R11, #0         ; SEND CSR
          MOV     R12, #0         ; WAIT FOR FI/FO OVERRUN
          MOV     R13, #0         ; SKIP ISR
          MOV     R14, #0         ; SAVE CSR
          MOV     R15, #0         ; INTERRUPT OCCURRED AT LEVEL 5
          MOV     R16, #0         ; RECOVER ONE CHARACTER
          MOV     R17, #0         ; RESET RETURN ADDRESS
          MOV     R18, #0         ; RETURN
          MOV     R19, #0         ; SET UP COUNTER - 8 CHARACTERS
          MOV     R20, #0         ; READ ONE CHARACTER
          MOV     R21, #0         ; BRANCH IF CHARACTER PRESENT
          MOV     R22, #0         ; CHECK ERRORS
          MOV     R23, #0         ; DUP DATA
          MOV     R24, #0         ; CLR ALL BUT LINE #
          MOV     R25, #0         ; CHECK DATA
          MOV     R26, #0         ; COUNT CHARACTERS
          MOV     R27, #0         ; BRANCH IF DONE
          MOV     R28, #0         ; SHORT WAIT LOOP
          MOV     R29, #0         ; GO READ ANOTHER CHARACTER
          MOV     R30, #0         ; SAVE CSR
          MOV     R31, #0         ; CHECK THAT FI/FO OVERRUN CLEARED
          MOV     R32, #0         ; BRANCH IF OK

```


E06

MAINDEC-11-DZDJA-0
DZDJA0.P11

TST61:

DJ11 LOGIC TESTS

MACY11 27(732)

21-SEP-76 13:43 PAGE 67

TEST RECEIVER UART OVERRUN ON ALL LINES

014152	105777	165032	48:	TSTB	BCSR	:R2 = CONTENTS OF CSR
014158	100375			BPL	48	:WAIT FOR DONE
014160	017702	165026		MOV	RBUFF, R2	:READ FI/FO
014164	100401			BMI	.+4	:BRANCH IF CHARACTER PRESENT
014166	104002			HLT+2		:NO CHARACTER PRESENT
						:R1 = LINE #
						:R2 = CONTENTS OF RBUFF
014170	032702	040000		BIT	#BIT14, R2	:CHECK FOR UART OVERRUN
014174	001001			SNE	.+4	:BRANCH IF OK
014176	104002			HLT+2		:UART OVERRUN MISSING
						:R1 = LINE #
						:R2 = CONTENTS OF RBUFF
014200	122702	000002		CMPE	R2, R2	:CHECK THE DATA
014204	001401			BEQ	.+4	:BRANCH IF OK
014206	104002			HLT+2		:DATA ERROR - 3RD CHAR OVERRUNS 2ND
						:R1 = LINE #
						:R2 = CONTENTS OF RBUFF
014208	010203			MOV	R2, R2	:DUP DATA
014210	000003			SWAB	R2	
014212	004003	177700		BITC	#177700, R2	:MASK ALL BIT LINE #, ERROR BITS
014214	000003			CMPE	R1, R2	:CHECK LINE #, ERRORS
014216	000003			BEQ	.+4	:BRANCH IF OK
				HLT+2		:LINE # OR OTHER RBUFF ERROR
						:R1 = LINE #
						:R2 = CONTENTS OF RBUFF
014220	017703	164760	78:	MOV	RBUFF, R2	:CHECK FOR MORE DATA
014224	100003			BEQ	8	:BRANCH IF OK
014226	104003			HLT+2		:EXTRA DATA IN FI/FO!
						:R1 = LINE #
						:R2 = CONTENTS OF RBUFF
014228	000773			BR	78	
014230	000377	164744	88:	DEC	BCSR	:CLEAR RECEIVER ENABLE
014234	000377	164740		MOV	BCSR	:SAVE CSR
014236	000377	100404		CMPE	#100404, R2	:CHECK CSR
014238	000401			BEQ	.+4	:BRANCH IF OK
014240	104003			HLT+2		:CSR ERROR
						:R1 = LINE NUMBER
						:R2 = CONTENTS OF CSR
014242	005201			INC	R1	:COUNT LINES
014244	006277	164726		ASL	BTOR	:GO TO NEXT LINE
014246	103874			BCC	LOF61	:BRANCH BACK IF MORE LINES
014270	005077	164714		CLR	BCSR	
014274	104400			SCOPE		

 :TEST 62: TEST BITS OF BCSR FOR READ/WRITE CAPABILITY
 :PROBABLE FAULTY LOGIC: M7285 (D2-2) E5, E1, (D2-3) E16, E2, E19, E35
 :*****

014276	012777	002010	164704	TST62:	MOV	#002010, BCSR	:SET CSR
014304	012777	177777	164702		MOV	#177777, BCSR	:SET ALL BITS OF BCSR

014312	017701	164676	MOV	BCSR, R1	:CHECK AND SAVE BCSR
014316	022701	177777	CMP	#177777, R1	:CHECK THAT ALL THE BITS ARE SET
014320	001401		BEG	.+4	:BRANCH IF OK
014324	104001		HLT+1		:BIT(S) OF BCSR FAILED TO SET
014326	005077	164662	CLR	BCSR	:CLEAR BCSR
014332	017701	164656	MOV	BCSR, R1	:CHECK THAT IT CLEARED AND SAVE
014336	001401		BEG	.+4	:BRANCH IF CLR
014340	104001		HLT+1		:BIT(S) OF BCSR FAILED TO CLEAR
014342	104400		SCOPE		

 :TEST 63: TEST THAT LINED CAN TRANSMIT AND RECEIVE A BREAK
 :ALSO CHECKS FRAMING ERROR(RBUF BIT13)
 :ALSO CHECKS PARITY ERROR(RBUF BIT12)
 :IF ODD PARITY IS SELECTED
 :PROBABLE FAULTY LOGIC: M7295 (D2-2) E5, E1, (D2-3) E16, E2, E19, E35

014344	004737	015536	TST63: JSR	PC,	3#INITA	:INITIALIZE
						:BIT2 = MAINTENANCE
						:BIT3 = CLEAR MOS
						:BIT10 = R/W BCSR
						:WAIT FOR MOS TO CLEAR
						:BIT0 = RECEIVER ENABLE
						:SEND BREAKS, LINE 0
						:SET UP TIMER
						:SHORT WAIT LOOP
014350	012777	000001	MOV	#1,	BCSR	
014356	013700	001302	MOV	TIMER,	RD	
014362	105305		DEC8	R5		
014364	001376		BNE	15		
014366	017701	164616	MOV	BCSR,	R1	:SAVE CSR
014372	105701		TSTB	R1		:CHECK FOR DONE
014374	100404		BMI	25		:BRANCH WHEN DONE
014376	005200		INC	RD		:WAIT A WHILE
014400	001370		BNE	15		
014402	104001		HLT+1			:DONE NEVER CAME UP
						:R1=CONTENTS OF CSR
014404	000430		BR	45		
014406	022701	002205	CMP	#2205,	R1	:CHECK REST OF CSR
014412	001401		BEG	.+4		:BRANCH IF OK
014414	104001		HLT+1			:CSR ERROR
						:R1=CONTENTS OF CSR
014416	017701	164570	MOV	RBUF,	R1	:GET DATA FROM FI/FO
014422	100401		BMI	.+4		:BRANCH IF OK
014424	104001		HLT+1			:CHARACTER PRESENT NOT UP
						:R1=CONTENTS OF RBUF
014426	032701	020000	BIT	#020000,	R1	:CHECK FOR FRAMING ERROR
014432	001001		BNE	.+4		:BRANCH IF OK
014434	104001		HLT+1			:FRAMING ERROR NOT UP
						:R1=CONTENTS OF RBUF
014436	133737	001310	BITB	DJPAR,	PARITY	:CHECK ODD PARITY FLAG
014444	001404		BEG	35		:BRANCH IF NOT
014446	032701	010000	BIT	#010000,	R1	:CHECK PARITY ERROR
014452	001005		BNE	45		


```

014454 104001 HLT+1 ;ODD PARITY SHOULD CAUSE PARITY ERROR
;R1=CONTENTS OF RBUF

014456 032701 010000 3$: BIT #010000,R1 ;CHECK PARITY ERROR
014458 001401 BEQ .+4 ;BRANCH IF OK
014464 104001 HLT+1 ;EVEN PARITY OR NO PARITY
;SHOULDN'T CAUSE PARITY ERROR
;R1=CONTENTS OF RBUF

014466 032701 040000 4$: BIT #040000,R1 ;CHECK UART OVERRUN
014468 001401 BEQ .+4 ;BRANCH IF OK
014474 104001 HLT+1 ;UART OVERRUN SET!!
;R1=CONTENTS OF RBUF

014476 032701 007400 BIT #007400,R1 ;CHECK LINE #
014502 001401 BEQ .+4 ;BRANCH IF OK
014504 104001 HLT+1 ;WRONG LINE# IN FI/FO
;R1=CONTENTS OF RBUF

014506 105701 TSTB R1 ;CHECK DATA
014508 001401 BEQ .+4 ;BRANCH IF OK
014510 104001 HLT+1 ;WRONG DATA RECEIVED
;R1=CONTENTS OF RBUF

014514 017701 164472 MOV RBUF, R1 ;READ FI/FO AGAIN
014520 100001 BR .+4 ;BRANCH IF OK
014522 104001 HLT+1 ;EXTRA CHAR IN FI/FO
;R1 = CONTENTS OF RBUF

014524 005077 164464 5$: CLR BCSR ;CLEAR BREAK CONTROL REG
014530 105305 DEC B ;SHORT WAIT LOOP-REGISTER A MARK
014532 001376 BNE B5
014534 104400 SCOPE

```

```

*****
:TEST 64: TEST THAT EACH LINE CAN TRANSMIT AND RECEIVE A BREAK
: ALSO CHECKS FRAMING ERROR(RBUF BIT12)
: ALSO CHECKS PARITY ERROR(RBUF BIT13)
: IF ODD PARITY IS SELECTED
:PROBABLE FAULTY LOGIC: M7285 (DR-A) R5, R1, (DR-3) R16, R2, R19, R35
*****

```

```

014536 004737 015536 TST64: JSR PC, @INITA ;INITIALIZE
;BIT2 = MAINTENANCE
;BIT3 = CLEAR MOS
;BIT10 = R/W BCSR
;WAIT FOR MOS TO CLEAR
;BIT0 = RECEIVER ENABLE
;SET UP LINE COUNTER
;SET UP LINE MARKER
;SET UP TIMER
;SET BREAK CONTROL BIT, LINE # IN R1
;SHORT WAIT LOOP

014542 005001 CLR R1
014544 012704 MOV #1, R4
014550 013700 MOV TIMER, R0
014554 050477 BIS R4, BCSR
014560 105305 DEC B
014562 001376 BNE B5
014564 017702 MOV BCSR, R2
014570 105702 TSTB R2
014572 100404 BMI B5
014574 005200 INC R0
;SAVE CSR
;CHECK FOR DONE
;BRANCH WHEN DONE
;WAIT A WHILE

```

014576	001370			BNE	13				
014600	104002			HLT+2					:DONE NEVER CAME UP
									:R1 = LINE #
									:R2 = CONTENTS OF CSR
014602	000433			BR	43				
014604	002702	002205	23:	CMP	#2205, R2				:CHECK REST OF CSR
014610	001401			BEG	.+4				:BRANCH IF OK
014612	104002			HLT+2					:CSR ERROR
									:R1 = LINE #
									:R2 = CONTENTS OF CSR
014614	017702	164372		MOV	RBUF, R2				:GET DATA FROM FI/FO
014616	100401			BNE	.+4				:BRANCH IF OK
014618	104002			HLT+2					:CHARACTER PRESENT NOT UP
									:R1 = LINE #
									:R2 = CONTENTS OF RBUF
014624	032702	020000		BIT	#020000, R2				:CHECK FOR FRAMING ERROR
014626	001001			BNE	.+4				:BRANCH IF OK
014628	104002			HLT+2					:FRAMING ERROR NOT UP
									:R1 = LINE #
									:R2 = CONTENTS OF RBUF
014634	010103			MOV	R1, R3				:GET LINE #
014636	006303			BIC	R3, R3				:/4
014640	006303			BIC	R3, R3				
014642	133763	001310	001276	BITB	DUPAR, PARITY(3)				:CHECK ODD PARITY FLAG
014644	001404			BNE	38				:BRANCH IF NOT
014646	032702	010000		BIT	#010000, R2				:CHECK PARITY ERROR
014648	001005			BNE	43				
014650	104002			HLT+2					:ODD PARITY SHOULD CAUSE PARITY ERROR
									:R1 = LINE #
									:R2 = CONTENTS OF RBUF
014662	032702	010000	33:	BIT	#010000, R2				:CHECK PARITY ERROR
014664	001401			BEG	43				:BRANCH IF OK
014670	104002			HLT+2					:EVEN PARITY OR NO PARITY
									:SHOULDN'T CAUSE PARITY ERROR
									:R1 = LINE #
									:R2 = CONTENTS OF RBUF
014672	032702	040000	43:	BIT	#040000, R2				:CHECK UART OVERRUN
014674	001401			BEG	.+4				:BRANCH IF OK
014700	104002			HLT+2					:UART OVERRUN SET!!
									:R1 = LINE #
									:R2 = CONTENTS OF RBUF
014702	010203			MOV	R2, R3				:DUP DATA
014704	000303			SWAB	R3				:LINE # IN LOW BYTE
014706	042703	177760		BIC	#177760, R3				:MASK ALL BUT LINE #
014712	020103			CMP	R1, R3				:CHECK FOR RIGHT LINE #
014714	001401			BEG	.+4				:BRANCH IF OK
014716	104002			HLT+2					:WRONG LINE# IN FI/FO
									:R1 = LINE #
									:R2 = CONTENTS OF RBUF
014720	105702			TSTB	R2				:CHECK DATA

MAINDEC-11-DZDJA-D
DZDJA.D.F11

TST64:

DJ11 LOGIC TESTS
TEST BREAKS ON ALL LINES

MACY11 27(732) 21-SEP-76 13:43 PAGE 71

3626	014722	001401		BEQ	+.4		: BRANCH IF OK
3627	014724	104002		HLT+2			: WRONG DATA RECEIVED
3628							: R1 = LINE #
3629							: R2 = CONTENTS OF RBUF
3630	014726	017702	164260	MOV	3RBUF, R2		: READ FI/FO AGAIN
3631	014732	100001		BPL	+.4		: BRANCH IF OK
3632	014734	104002		HLT+2			: EXTRA CHAR IN FI/FO
3633							: R1 = LINE #
3634							: R2 = CONTENTS OF RBUF
3635	014736	005077	164252	CLR	3BCSR		: CLEAR BREAK CONTROL REG
3636	014742	005201		INC	R1		: COUNT LINES
3637	014744	006304		ASL	R4		: UPDATE LINE MARKER
3638	014746	103300		BCC	LOP64		: BRANCH IF MORE LINES
3639							
3640	014750	005077	164234	CLR	3CSR		: CLEAR CSR
3641	014754	104400		SCOPE			
3642							
3643	014756	012737	000002 015772	MOV	#2, TIMES		

 : TEST 65: TEST THAT RESET CLEARS ALL BUFFERS
 : NOTE: THE FI/FO BUFFER IS NOT COMPLETELY CLEARED
 : BY RESET; ONLY CHARACTER PRESENT IS CLEARED.
 : PROBABLE FAULTY LOGIC: M7285 (D2-8) E13
 : *****

3652	014764	052737	000340 177776	TST65:	BIS	#340, 3#PS	: SET PROCESSOR TO LEVEL 7
3653	014772	012777	177777 164214	MOV	#177777, 3TCR		: SET ALL TCR BITS
3654	015000	012777	052507 164202	MOV	#052507, 3CSR		: SET ALL R/W BITS OF CSR
3655	015006	012777	177777 164200	MOV	#177777, 3BCSR		: SET ALL BREAK CONTROL BITS (SEND BREAKS)
3656				: NOTE:	ALL LINES SHOULD BE SENDING BREAKS, BUT NONE SHOULD BE RECEIVING		
3657				:	BECAUSE THE HALF DUPLEX BIT IS SET		
3658	015014	012777	177777 164174	MOV	#177777, 3TBUF		: LOADING TRANS BUFF WHEN BREAK BIT SET
3659							: SHOULD DO NOTHING
3660	015022	000005		RESET			: CLEAR THE WORLD
3661	015024	013737	001210 015034	MOV	CSR, 1\$		: CHECK CSR AND SAVE
3662	015032	013701		MOV	3(PC)+, R1		
3663	015034	000000		1\$:	000000		
3664	015036	001401		BEQ	+.4		
3665	015040	104001		HLT+1			
3666							
3667	015042	017701	164146	MOV	3TCR, R1		: CHECK TCR AND SAVE
3668	015046	001401		BEQ	+.4		
3669	015050	104001		HLT+1			
3670							
3671	015052	017701	164136	MOV	3BCSR, R1		: CHECK BCSR AND SAVE
3672	015056	001401		BEQ	+.4		
3673	015060	104001		HLT+1			
3674							
3675	015062	017701	164124	MOV	3RBUF, R1		: CHECK RBUF AND SAVE
3676	015066	100001		BPL	+.4		
3677	015070	104001		HLT+1			
3678							
3679	015072	017701	164120	MOV	3TBUF, R1		: CHECK TBUF AND SAVE
3680	015076	001401		BEQ	+.4		
3681	015100	104001		HLT+1			

3682
3683 015102 104400
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737

SCOPE

:TEST 66: SEND A BINARY COUNT PATTERN ON EACH LINE
:PROBABLE FAULTY LOGIC: COULD BE ALMOST ANYWHERE!

TST66: CLR R1 ;SET UP LINE COUNTER
MOV #100000,R3 ;SET UP RCV DATA
CLR R4 ;SET UP TRANS DATA
BIC #340, @#PS ;CLEAR PROCESSOR PRIORITY
BIS #200, @#PS ;SET PRIORITY TO 4
MOV #1, R0 ;SET UP LINE MARKER
MOV #10, @CSR ;CLEAR MOS
MOV #010505, @CSR ;BIT0 = RECEIVE ENABLE
;BIT2 = MAINTENANCE
;BIT6 = RECEIVER INTERRUPT ENB
;BIT8 = TRANS SCAN ENABLE
;BIT12= STATUS ENABLE
;WAIT FOR MOS TO CLEAR
10\$: BIT #BIT4, @CSR
BNE 10\$
MOV #ISR66, @RCVVEC ;SET UP RECEIVER INTERRUPT VECTOR
MOV #240, @RCVLVL
1\$: MOV R0, @TCR ;TRANS CONTROL, ONE LINE AT A TIME
2\$: TST @CSR ;WAIT FOR TRANS READY
BPL 2\$
MOV R4, @TBUF ;SEND DATA
INCB R4 ;BINARY COUNT
BNE 2\$
3\$: TSTB R3 ;WAIT FOR RECEIVER DONE
BNE 3\$
MOV @CSR, R2 ;SAVE CSR
CMP #110505, R2 ;CHECK CSR
BEQ .+4 ;BRANCH IF OK
HLT+2 ;CSR ERROR
;R1=LINE #
;R2=CONTENTS OF CSR
MOV @RBUF, R2 ;CHECK CHARACTER PRESENT
BPL .+4 ;BRANCH IF OK
HLT+2 ;CHARACTER PRESENT SET!!
;R1=LINE #
;R2=CONTENTS OF CSR
ADD #400, R3 ;UPDATE LINE # IN EXPECTED DATA
INC R1 ;UPDATE LINE #
BIT #3, R1 ;CHECK FOR FOURTH LINE
BNE 4\$;BRANCH IF NOT
INC DJLEN ;MOVE CHARACTER LENGTH POINTER
ASL R0 ;UPDATE LINE MARKER
BCC 1\$;BRANCH IF MORE
BR END66 ;SKIP ISR
4\$: ISR66: MOV @RBUF, R2 ;READ FIRST DATA
BMI 11\$;BRANCH IF CHARACTER PRESENT
HLT+3 ;INTERRUPT BUT NO CHAR PRESENT

3738							R1=LINE #
3739							R2=CONTENTS OF RBUF
3740							R3=EXPECTED DATA
3741	015304	020203		11\$:	CMP	R2,	R3
3742	015306	001401			BEG	.+4	
3743	015310	104003			HLT+3		
3744							CHECK THE DATA
3745							BRANCH IF OK
3746							DATA ERROR
3747	015312	105203			INCB	R3	R1=LINE #
3748	015314	147703	163766		BICB	DJLEN, R3	R2=CONTENTS OF RBUF
3749	015320	001403			BEG	12\$	R3=EXPECTED DATA
3750	015322	017702	163664		MOV	RBUF, R2	UPDATE EXPECTED DATA
3751	015326	100766			BMI	11\$	MASK CHARACTER LENGTH
3752	015330	000002		12\$:	RTI		BRANCH OUT IF DATA=0
3753							READ MORE DATA
3754	015332	162737	000004	001306	END66:	SUB	#4, DJLEN
3755	015340	013777	001222	163652	MOV	RCVLVL, RCVVEC	; RESET CHAR LENGTH POINTER
3756	015346	012777	000004	163646	MOV	#IOT, RCVLVL	; RESTORE RECEIVER INT. VEC
3757	015354	005077	163634		CLR	TCCR	
3758	015360	005077	163624		CLR	CCSR	; CLEAR TCR
3759	015364	104400			SCOPE		; CLEAR CSR
3760							
3761	015366	012737	000020	015772	MOV	#20, TIMES	
3762	015374	023737	001304	001234	CMP	DJUUT, UNITS	; CHECK FOR LAST UNIT
3763	015402	002002			BGE	DONE	; BRANCH IF LAST UNIT
3764	015404	000137	002302		JMP	RESTAR	; JUMP IF NOT
3765							
3766	015410				DONE:		
3767	015410	004737	017640		JSR	PC, KBDINT	
3768	015414	062737	000001	001206	ADD	#1, PCNT+2	; ADD 1 TO THE PASS COUNT
3769	015422	005537	001204		ADC	PCNT	; MAKE IT DOUBLE PREC.
3770	015426	032777	002000	163662	BIT	#SW10, JSWR	; RING THE BELL?
3771	015434	001004			BNE	1\$	; NO!
3772	015436	000004	000007		TYPE	, BELL	; RING THE BELL
3773	015442	000004	000177		TYPE	, 177	; TYPE A FILLER FOR 11/05
3774	015446	005046			CLR	-(6)	; CLEAR TRACE TRAP
3775	015450	032777	010000	163640	BIT	#SW12, JSWR	; RUN WITH TRT?
3776	015456	001010			BNE	2\$	; SKIP T BIT
3777	015460	005137	015532		COM	, TBIT	; COMPLIMENT FLAG
3778	015464	100005			BPL	2\$	; SKIP IF PLUS
3779	015466	052716	000020		BIS	#20, (6)	; SET TRACE TRAP
3780	015472	012746	015526		MOV	#3\$, -(6)	; JUMP TO START OF TEST
3781	015476	000002			RTI		; RETURN
3782	015500	012746	015506		MOV	#4\$, -(6)	; JUMP TO START OF TEST
3783	015504	000002			RTI		; RETURN
3784	015506	013700	000042		MOV	#42, RO	; GET MONITOR ADDRESS
3785	015512	001405			BEG	3\$	; IF NONE
3786	015514	000005			RESET		; RESET AND
3787		015516			\$ENDAD =	.	
3788	015516	004710			JSR	7, (0)	; GO TO MONITOR
3789	015520	000240			NOP		; SAVE ROOM
3790	015522	000240			NOP		; FOR
3791	015524	000240			NOP		; ACT11
3792	015526	000137	002302		JMP	RESTAR	; RETURN
3793							

```

3794 015532 000000 .TBIT: 0 ;T BIT FLAG
3795
3796 015534 000002 YESRT: RTI ;RETURN FROM TRACE TRAP
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850

```

INITIALIZATION ROUTINE
DEVICE CSR REGISTER
ON FAILURE: REGISTER 0 CONTAINS ERROR ADDRESS
SET:
BIT01 = HALF DUPLEX
BIT02 = MAINTENANCE
BIT03 = CLEAR MOS
BIT05 = RECEIVER INTERRUPT ENABLE
BIT08 = MASTER XMTR SCAN ENS
BIT10 = R/W BCSR
WAIT FOR MOS TO CLEAR
SET:
BIT00 = RECEIVER ENABLE
SET
BIT(S)2,3,10 THEN 0
INITA: MOV #2014,BCSR
BR INTR
INITB: MOV #514,BCSR
BR INTR
INITC: MOV #416,BCSR
BR INTR
INITD: MOV #414,BCSR
BR INTR
INITR: CLR R0
1\$: INC R0
BNE 2\$
MOV (SP),R0
SUB #2,R0
HLT
2\$: BIT #BIT4,BCSR
BNE 1\$
BIS #1,BCSR
RTS PC
CONTINUE
\$SCOPE SCOPE LOOP HANDLER
THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.

M06

```

3850          : "SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
3851          : RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
3852
3853 015632 004737 017640          TRAPS: JSR      PC,      KBDINT
3854 015636 032777 000400 163452      BIT      #SW8,ASWR      : LOOP ON SPEC. TEST?
3855 015644 001404          BEQ      1$      : NO LOOP ON SPEC. TEST
3856 015646 127737 163444 001200      CMPB     ASWR,ICNT      : ON RIGHT TEST? *SW7-0*
3857 015654 001434          BEQ      OVER$      : NOT RIGHT TEST
3858 015656 032777 040000 163432 1$:      BIT      #SW14,ASWR      : LOOP ON TEST?
3859 015664 001026          BNE      KITS$      : LOOP ON TEST IS SET
3860 015666 032777 004000 163422      BIT      #SW11,ASWR      : KILL ITERATIONS
3861 015674 001012          BNE      SVLAD$      : YES - KILL ITERATIONS
3862 015676 105737 001201          TSTB     ICNT+1      : FIRST ONE?
3863 015702 001404          BEQ      2$      : BRANCH IF FIRST
3864 015704 123737 015772 001201      CMPB     TIMES,ICNT+1      : DONE?
3865 015712 001013          BNE      KITS$      : BRANCH IF NOT
3866 015714 112737 000001 001201 2$:      MOVB     #1,ICNT+1      : FIRST ITERATION
3867 015722 105237 001200          SVLAD$: INCB     ICNT      : COUNT TEST NUMBERS
3868 015726 011637 015770          MOV      (6),LAD      : SAVE LOOP ADDRESS
3869 015732 013737 001200 001320      MOV      ICNT,3#DISPLAY      : DISPLAY TEST NO. AND ITERATION COUNT
3870 015740 000002          RTI              : RETURN
3871
3872 015742 105237 001201          KITS$: INCB     ICNT+1      : INC THE ITERATION COUNT
3873 015746 013737 001200 001320      OVER$: MOV      ICNT,3#DISPLAY      : SET UP DISPLAY
3874 015754 005737 015770          TST      LAD      : FIRST ONE?
3875 015760 001760          BEQ      SVLAD$      : YES
3876 015762 013716 015770          MOV      LAD,(6)      : FUDGE RETURN ADDRESS
3877 015766 000002          RTI              : FIXES PS
3878
3879 015770 000000          LAD:      0      : LOOP ADDRESS
3880 015772 000020          TIMES: 20      : RUN 20 TIMES

```

3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929

015774 004737 017640
016000 032777 002000 163310
016006 001402
016010 000004 000007
016014 005237 001202
016020 032777 020000 163270
016026 001026
016030 000004 016034
016040 011637 016144
016044 162737 000002 016144
016052 117737 000066 016142
016060 013705 016144
016064 004737 016562
016070 000004 016074
016100 004737 016146
016104 005777 163206
016110 100001
016112 000000
016114 004737 017640
016120 032777 001000 163170
016126 001001
016130 000002
016132 105037 001201
016136 000137 015742

016142 000000
016144 000000

016146
016146 013705 001210
016152 004737 016562
016156 042737 007700 016204
016164 105337 016142
016170 100411
016172 062737 000100 016204
016200 000004 017241
016204 010005
016206 004737 016562
016212 000764
016214 000207

```

; $HLT ERROR TIMEOUT HANDLER
; THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
; ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
; AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
; "HALT" ON ERROR, AND INHIBIT TIMEOUTS. AN OPTIONAL ARGUMENT
; (HLT+3) WILL BE PLACED IN "HLTCT$:" FOR ADDITIONAL TIMEOUTS.

EMTS: JSR PC, KBDINT
      BIT #SW10, QSWR ; BELL ON ERROR?
      BEQ 1$ ; NO - SKIP
      TYPE ,BELL ; RING BELL
      INC ERRORS ; COUNT THE NUMBER OF ERRORS
      BIT #SW13, QSWR ; SKIP TIMEOUT IF SET
      BNE 2$ ; SKIP TIMEOUTS
      TYPE ,+2 ; ASCIIZ <15><12>
      MOV (6), HLTADR ; PUT ADDRESS OF INSTRUCTION ON STACK
      SUB #2, HLTADR ; FUDGE ADDRESS
      MOV B, HLTADR, HLTCT$ ; GET HLT ARGUMENT
      MOV HLTADR, TTY ; TYPE HLTADR IN OCTAL
      JSR PC, PRINTR ; TYPE LEADING ZERO'S
      TYPE ,+2 ; ASCIIZ " "
      JSR PC, ERRORS ; GO TO USER ERROR ROUTINE
      TST QSWR ; HALT ON ERROR
      BPL ,+4 ; SKIP IF CONTINUE
      HALT ; HALT ON ERROR!

      JSR PC, KBDINT
      BIT #SW9, QSWR ; CHECK FOR INHIBIT LOOP ON ERROR
      BNE ,+4 ; SKIP IF LOOP ON ERROR
      RTI ; RETURN
      CLRB ICNT+1 ; CLEAR ITERATION COUNT
      JMP KITS ; LOOP ON TEST UNTIL NO ERRORS

HLTCT$: 0 ; HLT ARGUMENT
HLTADR: 0 ; LAST HLT INSTRUCTION EXECUTED

ERRORS: MOV CSR, TTY ; TYPE CSR IN OCTAL
      JSR PC, PRINTR ; TYPE LEADING ZERO'S
      BIC #7700, 2$
      DEC B HLTCT$
      BMI 3$
      ADD #100, 2$
      TYPE ,SPACE
      MOV %0, TTY ; TYPE REGISTER X IN OCTAL
      JSR %7, PRINTR
      BR 1$
      RTS PC

```


[illegible]

C07

MAINDEC-11-020JA-D
020JAD.F11

DUAL LOGIC TESTS

TTY INPUT ROUTINE

MARCY 11 27(732) 21-SEP-76 13:43 PAGE 78

[illegible]

017334
000015
016510
153373
000013

664669
111-111

325

Unit	Day
0.6-1.0	0.5
1.1-1.5	1.0
1.6-2.0	1.5
2.1-2.5	2.0
2.6-3.0	2.5
3.1-3.5	3.0
3.6-4.0	3.5
4.1-4.5	4.0
4.6-5.0	4.5
5.1-5.5	5.0
5.6-6.0	5.5
6.1-6.5	6.0
6.6-7.0	6.5
7.1-7.5	7.0
7.6-8.0	7.5
8.1-8.5	8.0
8.6-9.0	8.5
9.1-9.5	9.0
9.6-10.0	9.5
10.1-10.5	10.0
10.6-11.0	10.5
11.1-11.5	11.0
11.6-12.0	11.5
12.1-12.5	12.0
12.6-13.0	12.5
13.1-13.5	13.0
13.6-14.0	13.5
14.1-14.5	14.0
14.6-15.0	14.5
15.1-15.5	15.0
15.6-16.0	15.5
16.1-16.5	16.0
16.6-17.0	16.5
17.1-17.5	17.0
17.6-18.0	17.5
18.1-18.5	18.0
18.6-19.0	18.5
19.1-19.5	19.0
19.6-20.0	19.5
20.1-20.5	20.0
20.6-21.0	20.5
21.1-21.5	21.0
21.6-22.0	21.5
22.1-22.5	22.0
22.6-23.0	22.5
23.1-23.5	23.0
23.6-24.0	23.5
24.1-24.5	24.0
24.6-25.0	24.5
25.1-25.5	25.0
25.6-26.0	25.5
26.1-26.5	26.0
26.6-27.0	26.5
27.1-27.5	27.0
27.6-28.0	27.5
28.1-28.5	28.0
28.6-29.0	28.5
29.1-29.5	29.0
29.6-30.0	29.5
30.1-30.5	30.0
30.6-31.0	30.5
31.1-31.5	31.0
31.6-32.0	31.5
32.1-32.5	32.0
32.6-33.0	32.5
33.1-33.5	33.0
33.6-34.0	33.5
34.1-34.5	34.0
34.6-35.0	34.5
35.1-35.5	35.0
35.6-36.0	35.5
36.1-36.5	36.0
36.6-37.0	36.5
37.1-37.5	37.0
37.6-38.0	37.5
38.1-38.5	38.0
38.6-39.0	38.5
39.1-39.5	39.0
39.6-40.0	39.5
40.1-40.5	40.0
40.6-41.0	40.5
41.1-41.5	41.0
41.6-42.0	41.5
42.1-42.5	42.0
42.6-43.0	42.5
43.1-43.5	43.0
43.6-44.0	43.5
44.1-44.5	44.0
44.6-45.0	44.5
45.1-45.5	45.0
45.6-46.0	45.5
46.1-46.5	46.0
46.6-47.0	46.5
47.1-47.5	47.0
47.6-48.0	47.5
48.1-48.5	48.0
48.6-49.0	48.5
49.1-49.5	49.0
49.6-50.0	49.5
50.1-50.5	50.0
50.6-51.0	50.5
51.1-51.5	51.0
51.6-52.0	51.5
52.1-52.5	52.0
52.6-53.0	52.5
53.1-53.5	53.0
53.6-54.0	53.5
54.1-54.5	54.0
54.6-55.0	54.5
55.1-55.5	55.0
55.6-56.0	55.5
56.1-56.5	56.0
56.6-57.0	56.5
57.1-57.5	57.0
57.6-58.0	57.5
58.1-58.5	58.0
58.6-59.0	58.5
59.1-59.5	59.0
59.6-60.0	59.5
60.1-60.5	60.0
60.6-61.0	60.5
61.1-61.5	61.0
61.6-62.0	61.5
62.1-62.5	62.0
62.6-63.0	62.5
63.1-63.5	63.0
63.6-64.0	63.5
64.1-64.5	64.0
64.6-65.0	

0
BLKW 20

$\text{C}_2\text{H}_5\text{OH} + \text{H}_2\text{O} \rightleftharpoons \text{C}_2\text{H}_5\text{OH}_2^+$
 $\text{C}_2\text{H}_5\text{OH}_2^+ + \text{H}_2\text{O} \rightleftharpoons \text{C}_2\text{H}_5\text{OH}_3^+$
 $\text{C}_2\text{H}_5\text{OH}_3^+ + \text{H}_2\text{O} \rightleftharpoons \text{C}_2\text{H}_5\text{OH}_4^+$

```
:ECHO IT
:CHECK FOR RETURN
:LOOP IF NOT RETURN
```

```

:ZAP RETURN (THE IS)
:TYPE A LINE FEED
:RESTORE R3
:RETURN

```

:TTY INPUT AREA

MAINPFC-11-02DJA-D
 000000.7:11

 D11 LOGIC TESTS
 OCTAL DUMP OF A WORD

MACY11 27(732) 21-SEP-76 13:43 PAGE 79

: SUCTAL

OCTAL TYPEOUT ROUTINE

: THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
 : ALL 6 CHARACTERS. SUPPRESS LEADING ZEROES. TYPE AN 18 BIT ADDRESS. OR TYPE
 : THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP15, OR BITYPE MACRO'S.

012737	170101	016720	BITYPE: MOV	#170101..PR	: SET BIT FLAG AND 16 CHARACTER COUNT
000001	016720	PRINTR: MOV	.PTIT	: NOW TYPE IT IN BIT FORM	
016720	016721	PRINTS: CLR	#1..PR	: SET ZERO FILL SWITCH	
177772	016721	.PTIT: MOV	.+6	: SKIP	
016722			.PR	: SUPPRESS LEADING ZERO'S	
			#-6..PR+1	: SET COUNT	
			R4 -(6)	: SAVE R4	
			#.PR+2,R4	: SET POINTER TO FIRST ASCII CHAR.	
			(4)	: CLEAR FIRST BYTE	
			.PRF	: ROTATE FIRST BIT	
			(4)	: CLEAR BYTE OF CHARACTER	
000100	016720	.PRF: CLR	#100..PR	: BIT TYPING MODE?	
			.PRF	: YES - SKIP 2 ROTATES	
			TTY	: ROTATE BIT INTO C	
			(4)	: BACK IT	
			TTY	: ROTATE BIT INTO C	
			(4)	: BACK IT	
			TTY	: ROTATE BIT INTO C	
			(4)	: BACK IT	
			(4)	: IS IT ZERO?	
			.+6	: SKIP INC	
016720			.PRF	: SET FILL SWITCH	
016720			.PRF	: CHECK FILL SWITCH	
000060			#.PR+1	: SKIP BIT SET	
016721			#.PR+2,R4	: MAKE INTO ASCII CHAR	
016722			.+6	: INC COUNT	
000060			#.PR+2,R4	: EMPTY BUFFER?	
016722			.+6	: SKIP IF NOT	
			#.PR+2,R4	: LOAD 1 ZERO	
			(4)	: NULL TERMINATOR	
			(6)+,R4	: RESTORE R4	
			.PR	: RETURN	
			.PRF	: COUNT, SWITCH, AND OUTPUT BUFFER	

E07

MAINDEC-11-02DJA-D
02DJD.P11DJ11 LOGIC TESTS
POWER DOWN AND UP ROUTINES

MACY11 27(732) 21-SEP-76 13:43 PAGE 80

017072	000126	PDOWNS:	MOV	#ILLUP,SPUVECS	:SET FOR FAST UP
000340	000122		MOV	#340,SPUVECS+2	:PRIO:7
			MOV	RO, -(6)	:PUSH RO ON STACK
			MOV	R1, -(6)	:PUSH R1 ON STACK
			MOV	R2, -(6)	:PUSH R2 ON STACK
			MOV	R3, -(6)	:PUSH R3 ON STACK
			MOV	R4, -(6)	:PUSH R4 ON STACK
			MOV	R5, -(6)	:PUSH R5 ON STACK
017076			MOV	SP, SAVR6	:SAVE SP
017010	000072		MOV	#PUPE,SPUVECS	:SET UP VECTOR
			HALT		:WAIT FOR PF
017010	012706	PUPS:	MOV	.SAVR6,SP	:GET SP
017014	005001		CLR	R1	:WAIT LOOP FOR THE TTY
017016	005001	16:	INC	R1	:WAIT FOR THE INC
017020	001376		BNE	16	:OF WORD
017022	012603		MOV	(6)+,R5	:POP STACK INTO R5
017024	012603		MOV	(6)+,R4	:POP STACK INTO R4
017026	012603		MOV	(6)+,R3	:POP STACK INTO R3
017030	012603		MOV	(6)+,R2	:POP STACK INTO R2
017032	012603		MOV	(6)+,R1	:POP STACK INTO R1
017034	012603		MOV	(6)+,R0	:POP STACK INTO R0
017036	012737	016744	MOV	#PDOWNS,SP	:SET UP THE POWER DOWN VECTOR
017040	000024	000026	MOV	#340,SPUVECS	:PRIO:7
017042	000004	017046	TYPE		:ASCIZ <15><12>"POWER"
017044	000137	002300	JMP	RESTART	:JMP TO USER ADDRESS
017072	000000		ILLUP:	HALT	:THE POWER UP SEQUENCE WAS STARTED
017074	000776		BR	.-2	:BEFORE THE POWER DOWN WAS COMPLETE
017076	000000		.SAVR6:	0	:PUT THE SP HERE
017100	000024	000026	PUVECS:	R4,R6	:POWER UP VECTOR

[illegible][illegible]

017126 000002

[illegible][illegible]

000004

000004

051522
030461
042522
000040
052103
042104

```

:*****
: IOT HANDLER.  REENTERENT ROUTINE TO EITHER TYPE MESSAGES OR
:   INDICATE A FAULSE INTERRUPT OR TRAP.
:*****

```

```

IOTRAP:  CMP      #1000, (SP)      :CHECK RETURN ADDRESS FOR FALSE TRAP
          BLT      IOTS           :BRANCH IF "TYPE" COMMAND INTENDED
          SUB      #4, (SP)       :GET VECTOR ADDRESS FROM RETURN ADDRESS
          MOV      (SP)+, R1      :PUT IN R1 FOR TYPING
          TST      (SP)+         :POP STACK
          MOV      (SP), R2      :SAVE RETURN ADDRESS FOR TYPING
          HLT+2                :UNEXPECTED INTERRUPT OR TRAP
                                     :R1 = VECTOR ADDRESS
                                     :R2 = RETURN PC
          BTT                    :CONTINUE THE PROGRAM

```

```

:MOALL  STYPE
:        STYPE      MESSAGE TYPEOUT ROUTINE

```

```
: THIS ROUTINE IS USED TO TYPE ASCII MESSAGES ON THE TTY. THE  
: CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ADR" - TYPES THE  
: MESSAGE STARTING IN LOCATION "ADR:" 2) "TYPE CHAR" - TYPES  
: THE ASCII "CHAR", AND 3) "PRINT (<15><12>"MESSAGE") - TYPES  
: THE MESSAGE WHICH IS IN LINE ASCII.
```

```

1073:  MOV      TTY, -(6)          ;SAVE TTY
      MOV      2(6), TTY      ;GET ADDRESS TO BE TYPED
      BIT      #177400, TTY   ;IS IT A TYPEM?
      BNE      15             ;NO
      MOV      TTY, .TYPE     ;GET THE CHARACTER
      MOV      #.TYPE, TTY    ;FUDGE THE ADDRESS
15:    TSTB     (TTY)           ;TERMINATOR?
      BEQ      25             ;GET OUT IF SO
      MOVB     (TTY)+, 3#177566 ;LOAD AND TYPE THE CHARACTER
      TSTB     3#177564       ;IS THE PRINTER READY
      BPL      14             ;WAIT UNTIL IT IS
      BR       15            ;GET THE NEXT CHARACTER
25:    MOV      2(6), -(6)     ;GET ADDRESS TO BE TYPED
      ADD      #2, 4(6)        ;ADD 2 TO THE ADDRESS
      CMP      (6)+, 2(6)      ;IS IT .+2?
      BNE      35             ;NO
      ADD      #2, TTY         ;ADD 2 TO THE ADDRESS
      BIC      #1, TTY        ;BACK UP TO AN EVEN BYTE
      MOV      TTY, 2(6)      ;RESTORE ADDRESS
35:    MOV      (6)+, TTY      ;RESTORE TTY
      RTI                    ;RETURN
      .TYPE:  D               ;CHARACTER TYPE LOCATION

```

RETURN: .ASCIZ (15) (12)
SPACE: .ASCIZ (15) (12)
MSGCODE: .ASCIZ (15) (12) "FIRST BILL ADDRESS: "

MSGVEC: .ASCIZ <15><12>"VECTOR ADDRESS: "

020072	9360	5011	1600
020073	9360	5011	1600
020074	9360	5011	1600
020075	9360	5011	1600
020076	9360	5011	1600
020077	9360	5011	1600
020078	9360	5011	1600
020079	9360	5011	1600
020080	9360	5011	1600
020081	9360	5011	1600
020082	9360	5011	1600
020083	9360	5011	1600
020084	9360	5011	1600
020085	9360	5011	1600
020086	9360	5011	1600
020087	9360	5011	1600
020088	9360	5011	1600
020089	9360	5011	1600
020090	9360	5011	1600
020091	9360	5011	1600
020092	9360	5011	1600
020093	9360	5011	1600
020094	9360	5011	1600
020095	9360	5011	1600
020096	9360	5011	1600
020097	9360	5011	1600
020098	9360	5011	1600
020099	9360	5011	1600
020100	9360	5011	1600

MSGNUM: .ASCIZ (15) (12) "NO. OF DJ11'S: "

MSGCON: .ASCII <15><12>"STANDARD CONFIGURATION" "

MSGLIN: .ASCIZ (15) 12 "LINES "

MODEM: :BOC17 " - "
MODEM: :BOC17 (18) (12)"CHAR LENGTH: "

```
MSGPAR: .ASCIZ  (5) (5) "PARITY(NO, ODD, EVEN): "
```

.EVEN

...SUBROUTINE TO PUT ONLY ONE LINE OF DATA ON THE SYSTEM
...THIS ROUTINE ONLY UPDATES THE DATA WHEN THE PROGRAM IS
...NOT IN THE RUNNING STATE. THE ROUTINE WILL COME!

[illegible]

```

;ROUTINE TO DETERMINE VECTOR ADDRESSES

```

```
38:  MOV     $0, 0(%SP) ;SAVE 0% VECTOR ON THE STACK
```


QUALITATIVE TESTS
SYSTEMATIC SYSTEM SIZE

Address	Hex	Label	Instruction	Comment
000000	000000		MOV SWREG, SWR	
000001	000001		MOV SWREG, SWR	
000002	000002		MOV SWREG, SWR	
000003	000003		MOV SWREG, SWR	
000004	000004		MOV SWREG, SWR	
000005	000005		MOV SWREG, SWR	
000006	000006		MOV SWREG, SWR	
000007	000007		MOV SWREG, SWR	
000008	000008		MOV SWREG, SWR	
000009	000009		MOV SWREG, SWR	
00000A	00000A		MOV SWREG, SWR	
00000B	00000B		MOV SWREG, SWR	
00000C	00000C		MOV SWREG, SWR	
00000D	00000D		MOV SWREG, SWR	
00000E	00000E		MOV SWREG, SWR	
00000F	00000F		MOV SWREG, SWR	
000010	000010		MOV SWREG, SWR	
000011	000011		MOV SWREG, SWR	
000012	000012		MOV SWREG, SWR	
000013	000013		MOV SWREG, SWR	
000014	000014		MOV SWREG, SWR	
000015	000015		MOV SWREG, SWR	
000016	000016		MOV SWREG, SWR	
000017	000017		MOV SWREG, SWR	
000018	000018		MOV SWREG, SWR	
000019	000019		MOV SWREG, SWR	
00001A	00001A		MOV SWREG, SWR	
00001B	00001B		MOV SWREG, SWR	
00001C	00001C		MOV SWREG, SWR	
00001D	00001D		MOV SWREG, SWR	
00001E	00001E		MOV SWREG, SWR	
00001F	00001F		MOV SWREG, SWR	
000020	000020		MOV SWREG, SWR	
000021	000021		MOV SWREG, SWR	
000022	000022		MOV SWREG, SWR	
000023	000023		MOV SWREG, SWR	
000024	000024		MOV SWREG, SWR	
000025	000025		MOV SWREG, SWR	
000026	000026		MOV SWREG, SWR	
000027	000027		MOV SWREG, SWR	
000028	000028		MOV SWREG, SWR	
000029	000029		MOV SWREG, SWR	
00002A	00002A		MOV SWREG, SWR	
00002B	00002B		MOV SWREG, SWR	
00002C	00002C		MOV SWREG, SWR	
00002D	00002D		MOV SWREG, SWR	
00002E	00002E		MOV SWREG, SWR	
00002F	00002F		MOV SWREG, SWR	
000030	000030		MOV SWREG, SWR	
000031	000031		MOV SWREG, SWR	
000032	000032		MOV SWREG, SWR	
000033	000033		MOV SWREG, SWR	
000034	000034		MOV SWREG, SWR	
000035	000035		MOV SWREG, SWR	
000036	000036		MOV SWREG, SWR	
000037	000037		MOV SWREG, SWR	
000038	000038		MOV SWREG, SWR	
000039	000039		MOV SWREG, SWR	
00003A	00003A		MOV SWREG, SWR	
00003B	00003B		MOV SWREG, SWR	
00003C	00003C		MOV SWREG, SWR	
00003D	00003D		MOV SWREG, SWR	
00003E	00003E		MOV SWREG, SWR	
00003F	00003F		MOV SWREG, SWR	
000040	000040		MOV SWREG, SWR	
000041	000041		MOV SWREG, SWR	
000042	000042		MOV SWREG, SWR	
000043	000043		MOV SWREG, SWR	
000044	000044		MOV SWREG, SWR	
000045	000045		MOV SWREG, SWR	
000046	000046		MOV SWREG, SWR	
000047	000047		MOV SWREG, SWR	
000048	000048		MOV SWREG, SWR	
000049	000049		MOV SWREG, SWR	
00004A	00004A		MOV SWREG, SWR	
00004B	00004B		MOV SWREG, SWR	
00004				

[illegible]

1	00	000000	0
2	00	000000	0
3	00	000000	0
4	00	000000	0
5	00	000000	0
6	00	000000	0
7	00	000000	0
8	00	000000	0
9	00	000000	0
10	00	000000	0
11	00	000000	0
12	00	000000	0
13	00	000000	0
14	00	000000	0
15	00	000000	0
16	00	000000	0
17	00	000000	0
18	00	000000	0
19	00	000000	0
20	00	000000	0
21	00	000000	0
22	00	000000	0
23	00	000000	0
24	00	000000	0
25	00	000000	0
26	00	000000	0
27	00	000000	0
28	00	000000	0
29	00	000000	0
30	00	000000	0
31	00	000000	0
32	00	000000	0
33	00	000000	0
34	00	000000	0
35	00	000000	0
36	00	000000	0
37	00	000000	0
38	00	000000	0
39	00	000000	0
40	00	000000	0
41	00	000000	0
42	00	000000	0
43	00	000000	0
44	00	000000	0
45	00	000000	0
46	00	000000	0
47	00	000000	0
48	00	000000	0
49	00	000000	0
50	00	000000	0
51	00	000000	0
52	00	000000	0
53	00	000000	0
54	00	000000	0
55	00	000000	0
56	00	000000	0
57	00	000000	0
58	00	000000	0
59	00	000000	0
60	00	000000	0
61	00	000000	0
62	00	000000	0
63	00	000000	0
64	00	000000	0
65	00	000000	0
66	00	000000	0
67	00	000000	0
68	00	000000	0
69	00	000000	0
70	00	000000	0
71	00	000000	0
72	00	000000	0
73	00	000000	0
74	00	000000	0
75	00	000000	0
76	00	000000	0
77	00	000000	0
78	00	000000	0
79	00	000000	0
80	00	000000	0
81	00	000000	0
82	00	000000	0
83	00	000000	0
84	00	000000	0
85	00	000000	0
86	00	000000	0
87	00	000000	0
88	00	000000	0
89	00	000000	0
90	00	000000	0
91	00	000000	0
92	00	000000	0
93	00	000000	0
94	00	000000	0
95	00	000000	0
96	00	000000	0
97	00	000000	0
98	00	0000	

```

18:      R18      PC
CNT:      0
NEW18:    .A0012 " NEW= "
BAREP0:    .A0012 <18><18>"BARE= "
.END

```


J07

MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS MACY11 27(732) 21-SEP-76 13:43 PAGE 86
 DZDJA.D.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

AUTO	017470	4158*	3458	3463*	3464	3484*	3539*	3563*	3635*	3655*	3671		
BCSR	001314	3457*											
BEGIN	001324	571*											
BELL	000007	3772	3893										
BITIPS	016552	4005*											
BITO	000001	497*	969	979	1111	1133	1201	1219	1237	1255	1293	1321	1349
		1377	1405	1433	1477	2812	2846	2935	3045	3090	3298		
BIT1	000002	498*	800	801	810	1560							
BIT10	002000	007*	884	885	894	2311							
BIT11	004000	008*	2394										
BIT12	010000	009*	906	915	2479	3244							
BIT13	020000	010*	2562	3140	3181	3255	3312						
BIT14	040000	011*	926	927	936	2645	3410						
BIT15	100000	012*	2728										
BIT16	000004	013*	821	822	831	948	962	981	1643				
BIT17	000010	014*	950	952	1726	2823	3316						
BIT18	000020	015*	956	1811	2924	2944	3379	3703	3938				
BIT19	000040	016*	1894										
BIT20	000100	017*	842	843	852	1977							
BIT21	000200	018*	2060										
BIT22	000400	019*	863	864	873	2145							
BIT23	001000	020*	2228										
CLAVEC	002336	727	732*										
ONT	020050	3940*	3958*	4237	4242*								
ONTLU	017770	730	4208	4228*	4236								
CONFIG	001766	629	660*	675									
CSR	001210	539*	738*	745*	750	779	780*	781*	800*	801	805	809*	810
		822	826	830*	831	842*	843	847	851*	852	863*	864*	865
		873	884*	885	889	893*	894	905*	906	910	914*	915	916
		931	935*	936	948*	950*	951	967*	968	969	973	976*	977
		994*	995*	996	1001*	1002*	1003*	1004*	1005	1019*	1020	1024*	1025
		1030	1110*	1113	1129	1134	1149*	1154	1174	1179	1200*	1203*	1204
		1220	1236*	1238	1249*	1264*	1266	1277*	1292*	1294	1305*	1320*	1321
		1348*	1350	1361*	1376*	1379	1389*	1404*	1406	1417*	1432*	1434*	1435
		1484	1488	1527	1533*	1561	1567	1571	1610	1616*	1644	1650	1651
		1699*	1727	1733	1737	1776	1782*	1812	1818	1822*	1861	1867*	1868
		1905	1944	1950*	1978	1984	1988	2027	2033*	2061	2067	2071*	2072
		2145	2152	2156	2195	2201*	2229	2235	2239	2278	2284*	2293	2294
		2361	2367*	2395	2401	2405	2444	2450*	2480	2486	2490	2529	2530
		2569	2573	2612	2618*	2646	2652	2656	2695	2701*	2729	2735	2736
		2784*	2814	2820	2823*	2824	2830	2843	2860*	2883	2886	2899	2900
		2931*	2941*	2944	2947	2953	2966*	2967	2996	3002	3015*	3046	3047
		3091	3094	3113*	3137	3148	3152	3180	3227	3244*	3245	3251	3252
		3287	3302	3308	3311	3316*	3319	3340	3365*	3376*	3379	3383	3384
		3396*	3397	3403	3405*	3436	3446*	3456*	3488	3566	3640*	3654*	3655
		3698*	3703	3708	3715	3758*	3817*	3820*	3823*	3826*	3838	3841*	3842
		560*	605										
DEFADR	001312	561*											
DEFVEC	001314	550*	601	603	605*	606*	607	745	4172*	4185			
DEVADR	001230	564*	3869*	3873*	4222*								
DISPLA	001320	526*	4222										
DISPRE	000174	558*	740*	747*	1511	1594	1677	1760	1784*	1845	1928	2011	2094
DJLEN	001306	2179	2262	2345	2428	2452*	2513	2596	2679	2762	2786*	3730*	3748
DJMXNO	000020	515*	655										
DJPAR	001310	559*	741*	748*	3510	3596							
DJUUT	001304	557*	658*	742	744*	749*	3762						

[illegible]

L07

 MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS MACY11 27(732) 21-SEP-76 13:43 PAGE 88
 DZDJD.F11 CROSS REFERENCE TABLE -- USER SYMBOLS

IOTRAP	017104	3941	3971	3972	3998#															
IOTS	017130	576	4077#																	
TSR33	004424	4078	4097#																	
TSR33	004424	1232	1242#																	
TSR33	004424	1260	1270#																	
TSR34	004664	1288	1298#																	
TSR35	005006	1316	1327#																	
TSR35	005126	1344	1355#																	
TSR35	005246	1372	1383#																	
TSR35	005366	1400	1411#																	
TSR35	005506	1428	1439#																	
TSR36	012620	3034	3053#																	
TSR37	012774	3079	3099#																	
TSR36	015274	3705	3735#																	
KBDINT	017640	3767	3853	3890	3908	4201#														
KITS	015742	3859	3865	3872#	3913															
LAD	015770	766*	1101*	1451*	1785*	2119*	2453*	2794*	3868*	3874	3876	3879#								
LENGTH	001236	553#	665	678	747															
LEVEL7=	000340	512#	778																	
LOP32	005574	1478#	1479																	
LOP33	005774	1561#	1562																	
LOP34	006174	1644#	1645																	
LOP35	006374	1727#	1728																	
LOP36	006606	1812#	1813																	
LOP37	007006	1895#	1896																	
LOP40	007206	1978#	1979																	
LOP41	007406	2061#	2062																	
LOP43	007620	2146#	2147																	
LOP43	010020	2229#	2230																	
LOP44	010220	2312#	2313																	
LOP45	010420	2395#	2396																	
LOP46	010632	2480#	2481																	
LOP47	011032	2563#	2564																	
LOP50	011232	2646#	2647																	
LOP51	011432	2729#	2730																	
LOP61	014060	3383#	3384	3444																
LOP64	014550	3562#	3638																	
MSGADR	017244	599	4122#																	
MSGCON	017343	660	4134#																	
MSGDAS	017410	684	4141#																	
MSGLEN	017414	689	4142#																	
MSGLIN	017377	681	4139#																	
MSGNUM	017320	622	4130#																	
MSGPAR	017435	704	4145#																	
MSGVEC	017274	610	4126#																	
N =	000067	421#	770	794#	815#	836#	857#	878#	899#	920#	941#	988#	1012#	1035#						
		1053#	1077#	1100#	1102	1141#	1192#	1226#	1254#	1282#	1310#	1338#	1366#	1394#						
		1422#	1450#	1452	1535#	1618#	1701#	1784#	1786	1869#	1952#	2035#	2118#	2190#						
		2203#	2286#	2369#	2452#	2454	2537#	2620#	2703#	2786#	2795	2864#	2934#	2992#						
		3028#	3073#	3118#	3370#	3450#	3469#	3545#	3643#	3644	3685#	3761#								
		4233	4244#																	
NEWIS	020052	514#																		
OPEN =	000000	3857	3873#																	
OVERS	015746	554#	679	718	3510	3596														
PARITY	001276	495#	572*	596*	623*	661*	683*	697*	690*	705*	730*	1470*	1553*	1536*						
PC =	%000007	1719*	1804*	1887*	1970*	2053*	2138*	2221*	2304*	2387*	2472*	2555*	2638*	2721*						

MO7

 MAINDEC-11-DZDJA-D DJ11 LOGIC TESTS MACY11 27(732) 21-SEP-76 13:43 PAGE 89
 DZDJA0.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

		2804*	2875*	2388*	3038*	3083*	3128*	3478*	3554*	3662	3767*	3842*	3853*	3890*
		3902*	3904*	3908*	3920*	3929*	3934*	3995*	4038*	4193*	4208*	4209*	4225*	4232*
		4240*												
PONT	001204	537*	592*	593*	3768*	3769*	3794							
PDOWN\$	016744	578	4040*	4062										
PRINTR	016562	3902	3920	3927	4007*	4232								
PRINTS	016572	693	687	4009*										
PS	= 177776	486*	1234*	1235*	1250*	1262*	1263*	1278*	1290*	1291*	1306*	1318*	1319*	1334*
		1346*	1347*	1362*	1374*	1375*	1390*	1402*	1403*	1418*	1430*	1431*	1446*	3036*
		3037*	3069*	3081*	3082*	3114*	3240*	3241*	3299*	3300*	3366*	3652*	3694*	3695*
		3961	3965*											
FSTEMP	016362	3961*	3965	3968*										
FUPS	017010	4049	4052*											
FUVECS	017100	4040*	4041*	4049*	4071*									
RBUF	001212	540*	752*	1492	1523	1575	1606	1658	1689	1741	1772	1826	1857	1909
		1940	1992	2023	2075	2106	2160	2191	2243	2274	2326	2357	2409	2440
		2494	2525	2577	2608	2660	2691	2743	2774	2827	2855	2905	2913	2921
		2962	2975	3011	3020	3057	3102	3201	3261	3266	3325	3356	3405	3428
		3501	3535	3581	3630	3675	3721	3735	3750					
RCVLVL	001222	546*	760*	3035*	3065	3066*	3080*	3110	3111*	3243*	3360	3361*	3706*	3755
		3756*												
RCVVEC	001220	545*	739*	746*	758	3034*	3065*	3079*	3110*	3242*	3301*	3360*	3705*	3755*
READIN	016216	600	611	3933*	4234									
READS	016364	623	661	690	705	3934	3970*							
RESTAR	002302	532	597	663	671	673	724*	3764	3792	4065				
RETURN	017236	4120*												
RO	=%000000	487*	573*	574*	575*	576*	577*	578*	579*	580*	581*	592*	593*	595*
		664*	668*	676*	720*	721	732*	733	734*	735*	736	1112*	1115*	1121
		1153*	1156*	1164	1202*	1207*	1211*	1214*	1481*	1486*	1491	1564*	1569*	1574
		1647*	1652*	1657	1730*	1735*	1740	1815*	1820*	1825	1898*	1903*	1908	1981*
		1986*	1991	2064*	2069*	2074	2149*	2154*	2159	2232*	2237*	2242	2315*	2320*
		2325	2398*	2403*	2408	2483*	2488*	2493	2566*	2571*	2576	2649*	2654*	2659
		2732*	2737*	2742	2788*	2789*	2790*	2791*	2793	2839*	2848*	2893*	2898*	2950*
		2959*	2970*	2999*	3008*	3136*	3144*	3147*	3156*	3177*	3183*	3200*	3221*	3265*
		3281*	3324*	3341*	3343	3389*	3394*	3485*	3491*	3562*	3569*	3696*	3707	3731*
		3784*	3828*	3830*	3833*	3834*	4042	4061*	4159*	4163	4165*	4171*	4172	
R1	=%0000001	488*	626*	628	630	632*	645	646*	647*	648*	650*	651*	653*	655
		657	665*	666*	677*	682	685*	686	688*	716	750*	751*	752	753*
		754	756*	757	758*	759	760	761	762	763	764	787*	951*	952
		956	962	968*	980*	996*	997	1005*	1006	1020*	1030*	1042*	1043	1048*
		1061*	1062	1070*	1071	1085*	1095*	1113*	1122*	1123*	1124*	1125	1134*	1150*
		1167	1186*	1203*	1212*	1220*	1238*	1266*	1294*	1322*	1350*	1378*	1406*	1434*
		1478*	1488*	1492*	1496	1503	1512	1518	1523*	1527*	1528	1561*	1571*	1575*
		1579	1586	1595	1601	1606*	1610*	1611	1644*	1654*	1658*	1662	1669	1678
		1684	1689*	1693*	1694	1727*	1737*	1741*	1745	1752	1761	1767	1772*	1776*
		1777	1812*	1822*	1826*	1830	1837	1846	1852	1857*	1861*	1862	1895*	1905*
		1909*	1913	1920	1929	1935	1940*	1944*	1945	1978*	1998*	1992*	1996	2003
		2012	2018	2023*	2027*	2028	2061*	2071*	2075*	2079	2086	2095	2101	2106*
		2110*	2111	2146*	2156*	2160*	2164	2171	2180	2186	2191*	2195*	2196	2229*
		2239*	2243*	2247	2254	2263	2269	2274*	2278*	2279	2312*	2322*	2326*	2330
		2337	2346	2352	2357*	2361*	2362	2395*	2405*	2409*	2413	2420	2429	2435
		2440*	2444*	2445	2480*	2490*	2494*	2498	2505	2514	2520	2525*	2529*	2530
		2563*	2573*	2577*	2581	2588	2597	2603	2608*	2612*	2613	2646*	2656*	2660*
		2664	2671	2680	2686	2691*	2695*	2696	2729*	2739*	2743*	2747	2754	2763
		2769	2774*	2778*	2779	2814*	2827*	2830*	2831	2843*	2844	2851	2855*	2883*
		2886*	2889*	2896*	2901	2905*	2909	2913*	2917	2921*	2925*	2926	2947*	2953*

		2954	2962*	2967*	2968	2975*	2996*	3002*	3003	3011*	3020*	3046*	3057*	3061
		3091*	3102*	3106	3137*	3140	3148*	3152*	3161	3165	3180*	3181	3187	3201*
		3205	3213	3216	3227*	3228	3245*	3246	3254*	3255	3259*	3256*	3270	3274
		3277	3287*	3288	3302*	3303	3311*	3312	3319*	3320	3325*	3329	3333	3336
		3340*	3345	3351	3356*	3381*	3423	3442*	3458*	3459	3464*	3488*	3489	3496
		3501*	3506	3512	3517	3522	3526	3530	3535*	3560*	3593	3618	3636*	3652*
		3667*	3671*	3675*	3679*	3691*	3727*	3728	3936	3939*	3944*	3951*	3953*	3955*
		3957*	3964*	4043	4053*	4054*	4060*	4080*	4161*	4166*	4176	4185*	4186*	4189
		4190*												
R2	=%0000002	489#	625*	626	635	637	639	641	678*	696*	697*	701	703*	1129
		1154*	1165*	1166*	1167	1179*	1503*	1504*	1505*	1506	1511*	1512	1517*	1518
		1586*	1587*	1588*	1589	1594*	1595	1600*	1601	1669*	1670*	1671*	1672	1677*
		1678	1683*	1684	1752*	1753*	1754*	1755	1760*	1761	1766*	1767	1837*	1838*
		1839*	1840	1845*	1846	1851*	1852	1920*	1921*	1922*	1923	1928*	1929	1934*
		1935	2003*	2004*	2005*	2006	2011*	2012	2017*	2018	2086*	2087*	2088*	2089
		2094*	2095	2100*	2101	2171*	2172*	2173*	2174	2179*	2180	2185*	2186	2254*
		2255*	2256*	2257	2262*	2263	2268*	2269	2337*	2338*	2339*	2340	2345*	2346
		2351*	2352	2420*	2421*	2422*	2423	2428*	2429	2434*	2435	2505*	2506*	2507*
		2508	2513*	2514	2519*	2520	2588*	2589*	2590*	2591	2596*	2597	2602*	2603
		2671*	2672*	2673*	2674	2679*	2680	2685*	2686	2754*	2755*	2756*	2757	2762*
		2763	2768*	2769	3213*	3214*	3215*	3216	3274*	3275*	3276*	3277	3333*	3334*
		3335*	3336	3383*	3386*	3392*	3397*	3398	3405*	3410	3415	3420	3428*	3436*
		3437	3566*	3567	3575	3581*	3587	3598	3604	3610	3615	3625	3630*	3715*
		3716	3721*	3735*	3741	3750*	3937	3941*	3942	3945	3963*	4044	4059*	4082*
		4159*	4169*											
R3	=%0000003	490#	641*	642*	651	679*	714*	715*	718*	1174*	3420*	3421*	3422*	3423
		3593*	3594*	3595*	3615*	3616*	3617*	3618	3692*	3713	3726*	3741	3747*	3748*
		3938	3945*	3946	3948*	3949	3957	3962*	3970	3971*	3972	3994*	4045	4058*
R4	=%0000004	491#	680*	714	719*	1151*	1152	1178	1187*	2813*	2816	2817*	3561*	3563
		3637*	3693*	3710	3711*	4011	4012*	4032	4037*	4046	4057*			
R5	=%0000005	492#	600*	611*	949*	958*	1482*	1565*	1648*	1731*	1816*	1899*	1992*	2065*
		2150*	2233*	2316*	2399*	2484*	2567*	2650*	2733*	2840*	2894*	2951*	3000*	3150*
	</													

[illegible]

MACY 11 27(732)
USER SYMBOLS

00000000-11-00000-0 FULL LOGIC TESTS
 00000000-11-00000-0 CROSS REFERENCE TABLES -

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13</																																																																																							

21-SEP-76 13:43 PAGE 93

[illegible]

1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	4221	

REFERENCE TABLE -- MACRO NAMES

[illegible]

[illegible]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87													

[illegible][illegible]

K08

Spooler runtime 16 Seconds, 71 KCS, 416 disk reads, 3 disk writes, 97 pages.

Shoulder Punting 16 Seconds, 171 KCS, 116 Disk Reads, 9 Disk Writes, 37 pages
 12-01-76 15:13:59 Monitor IPC-0 0070 (103) success

[illegible]