

VT20

DIAGNOSTIC TEST
MD-11-DBVTA-D

EP DBVTA-D DL-B
COPYRIGHT 1977
FICHE 1 OF 1

MAR 1977
digital
MADE IN USA

The image displays a grid of 100 small diagnostic test screens, arranged in 10 rows and 10 columns. Each screen shows various data points, including alphanumeric strings, small graphs, and status indicators. The screens are densely packed and cover most of the page area below the header.

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DBVTA-D-D
PRODUCT NAME: VT20 DIAGNOSTIC TEST
DATE RELEASED: JANUARY, 1977
MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT 1973,1977 BY DIGITAL EQUIPMENT CORPORATION

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

VT20 SYSTEM DIAGNOSTIC TEST PROGRAM
DBVTA0.P11

TABLE OF CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS (EQUIPMENT & MEMORY)
3.	LOADING PROCEDURES
	A. STANDARD LOADING PROCEDURE
	B. REMOTE LOADING PROCEDURE
4.	STARTING PROCEDURE
	1. WITH TELETYPE AVAILABLE
	2. WITHOUT TELETYPE
5.	RESTART PROCEDURES
6.	CONSOLE SWITCH SETTINGS
7.	CONTROL SWITCHES (TELETYPE)
8.	PROGRAM TEST TABLE & DIRECTORY
9.	VT LOGIC TEST
10.	VT CHARACTER SET TEST
11.	VT FIELD CONTROL TEST
12.	VT CSR PRESET TEST
13.	VT END OF LINE TEST
14.	VT END OF SCREEN TEST
15.	VT BLANK CONTROL TEST
16.	VT ALIGNMENT TEST
17.	VT FOCUS TEST
18.	VT WORST CASE TEST
19.	VT CURSOR CONTROL TEST
20.	VT ODD ADDRESS TEST
21.	VT RUNALL TEST
22.	ASCII KBD LOGIC TEST
23.	ASCII KBD LIGHT & SWITCH TEST
24.	ASCII KBD CHARACTERS SET TEST
25.	ASCII KBD REPEATABILITY TEST
26.	VT20 SYSTEM TEST
27.	VT/KBD SUBTEST SELECTOR
28.	TEST EXECUTION TIME TABLE
29.	LISTING
1.	A B S T R A C T

THIS PROGRAM TESTS, EXERCISES AND DIAGNOSES THE VT20 SYSTEM (KEYBOARDS & DISPLAYS) IN ITS ENTIRETY. THE PROGRAM CONSISTS OF EIGHTEEN INDIVIDUALLY SELECTABLE TESTS WHICH FACILITATE IN CHECKOUT AND ACCEPTANCE OF THE VT20. THE SYSTEM IS TESTED ON A UNIT (1 KEYBOARD & 1 DISPLAY) BASES. THIS HOLDS TRUE IN ALL TESTS EXCEPT FOR TEST 21. HERE, THE VT20 IS TESTED AS A SLAVE SYSTEM IN CONFIGURATION WITH EITHER A PDP-8 OR PDP-11 HOST COMPUTER. THIS TEST REQUIRES FOR THE VT20 TO BE CABLED VIA SERIAL LINE INTERFACE TO THE HOST COMPUTER. THE HOST DIAGNOSTIC, EITHER MD-11-DZVTE(PDP-11 W/DL11'S), MD-11-DZVTG(PDP-11 W/DH11'S) OR MD-08-DIVTB(PDP-8 W/ KLB1'S) IS TO BE LOADED INTO THE HOST COMPUTER. THIS PROGRAM ACTS AS SERIAL LINE INTERFACE DIAGNOSTIC AND A DATA RECEIVE/TRANSMIT ROUTINE. REFER TO THE SPECIFIC MAINDEC FOR A FURTHER EXPLANATION. THIS DIAGNOSTIC IS SET

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
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

UP TO OPERATE 'WITH' OR 'WITHOUT' A TELETYPE. IF A TELETYPE ISN'T AVAILABLE PROGRAM 'HALTS' ARE USED TO REPORT ERRORS AND COLLECT NEEDED INFORMATION. THE PROGRAM RESPONDS FUNCTIONALLY THE SAME WITH OR WITHOUT THE TELETYPE. IT IS IMPERATIVE THAT THE DISPLAY TESTS BE RUN AND PROVED FULLY OPERATIONAL BEFORE RUNNING THE KEYBOARD TESTS. THIS IS NECESSARY SINCE THE DISPLAY IS USED IN CONJUNCTION WITH THE 'KBD' (FUNCTIONAL & ASCII KEYBOARD) TESTS.

2. REQUIREMENTS (EQUIPMENT & MEMORY)

- A. VT20 WITH EITHER 8 OR 16K OF MEMORY AND 1 TO 4 TUBES.
- B. HOST COMPUTER W/ CONSOLE DEVICE (TEST 21 ONLY)

- 1. MD-11-DZVTE FOR PDP-11 HOST W/DL11 INTERFACE
- 2. MD-11-DZVTG FOR PDP-11 HOST W/ DH11 INTERFACE
- 3. MD-08-DIVTB FOR PDP-8 HOST W/ KL8J INTERFACE

3. LOADING PROCEDURE

- A. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES IF A READ IN DEVICE (PC11 OR ASR-33) IS AVAILABLE.
- B. THIS PROGRAM MAY ALSO BE BOOTED OVER FROM THE HOST DIAGNOSTIC. THIS IS DONE BY MODIFYING THE BOOTSTRAP LOADED. SIMPLY REPLACE THE PC11 OR TTY CSR ADDRESS IN LOCATION 37776 WITH ONE OF THE DL11 CSR ADDRESSES (175610 OR 175620). REFER TO THE PARTICULAR MAINDEC BEING USED FOR THE BOOT PROCEDURE.

4. STARTING PROCEDURE

- A. THERE ARE TWO STARTING ADDRESSES. (1) WITH TELETYPE AND (2) WITHOUT A TELETYPE.

- 1. WITH TELETYPE (OR OTHER TERMINAL).
LOAD AND START THE PROGRAM AT ADDRESS '200'.
WHEN STARTED THE PROGRAM WILL PRINT THE PROGRAM
HEADER AND THEN A SERIES OF QUESTIONS ARE ASKED.

QUESTION NO.1 "FOREIGN CHARACTER SET (Y OR N)?"

THIS ENABLES THE PROGRAM TO DISPLAY THE FOREIGN CHARACTER SET IN THE CHARACTER DISPLAY TESTS IF THE FOREIGN CHARACTER ROMS ARE PRESENT.

QUESTION NO.2 - "NUMBER OF TUBES?"

TYPE IN THE NUMBER OF TUBES ON SYSTEM (1,2,3 OR 4)

QUESTION NO.3 - "ARE DEFAULT ADDRESSES & VECTORS OK (Y OR N)?"

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

IF THE ANSWER TO THIS QUESTION WAS 'Y', THEN STANDARD 'FK' KEYBOARD AND 'VT' DEVICE ADDRESSES AND VECTOR ADDRESSES WILL BE USED. IT WILL ALSO AUTOMATICALLY ASSOCIATE THE 1ST DL (175610) WITH THE FIRST TUBE, DL-175620 WITH THE 2ND TUBE AND SO ON.

IF QUESTION NO.3 WAS 'N', THEN THE NEXT QUESTIONS WILL BE ASKED:
"FK VECTOR AND DEVICE ADDRESS FOR UNIT X" (WHERE X REPRESENTS THE UNIT ADDRESS TO BE SET UP. RESPOND TO THIS QUESTION BY TYPING THE FK 'VECTOR' AND 'DEVICE' ADDRESS SEPERATED BY A COMMA.

AFTER ALL THE FK ADDRESSES HAVE BEEN SETUP, THE PROGRAM WILL THEN PRINT:
"VT VECTOR AND DEVICE ADDRESS FOR UNIT X" - ANSWER SAME AS ABOVE ONLY FOR VT.

THE PROGRAM WILL THEN PRINT:
"DL11 VECTOR FOR UNIT X" - WHERE X REPRESENTS THE UNIT NUMBER OF THE ASSOCIATED DL11. RESPONSE TO THIS QUESTION BY TYPING THE DL11 VECTOR ADDRESS.

QUESTION NO.3 - "TUBE '0','1','2','3',OR '4'"

TO RUN A TEST ON A SELECTED TUBE, TYPE THE NUMBER OF THE TUBE TO BE TESTED (0,1,2 OR 3) AND (CR). ALSO, AN OPTION TO THIS IS TO TYPE A '4'. THIS WILL CAUSE THE PROGRAM TO TEST ALL AVAILABLE TUBES, ONE AT A TIME ON ANY SELECTED TEST 1-15. THUS, AFTER ONE TUBE HAS COMPLETED A TEST, THE NEXT SEQUENTIAL TUBE WILL AUTOMATICALLY BE TESTED. AFTER THE LAST TUBE HAS BEEN TESTED, THE PROGRAM WILL RE-CYCLE AND AGAIN BE TESTED. THIS OPTION CAN ALSO BE USED TO RUN EVERY TEST (1-15) ON EVERY TUBE (REFER TO SECTION 14.).

QUESTION NO.4 "TYPE TEST NO. TO BE EXECUTED" - RESPOND

BY TYPING THE TEST NO: '0-21' YOU WISH TO RUN.

2. WITHOUT TELETYPE (OR OTHER TERMINAL)

LOAD AND START PROGRAM AT ADDRESS '204'. THE PROGRAM WILL THEN EXECUTE A SERIES OF HALTS, THESE HALTS ENABLE THE USER TO ANSWER QUESTIONS TO THE PROGRAM VIA THE SWITCH REGISTER. PDP 11/05, 11/10
HALT LOCATIONS ARE INDICATED BY THE LOCATIONS ON BRACKETS. THESE HALTS WITH THEIR ASSOCIATED HALT ADDRESSES ARE NOW DESCRIBED:

A. HALT AT ADDRESS 001734 (1736)

HALT USED TO ASK YOU IF YOUR VT20 HAS A FOREIGN CHARACTER SET, IF SO SET THE SWITCH REGISTER EQUAL TO

A 1 (IF NOT CLEAR THE SWITCH REGISTER) THEN PRESS CONTINUE.

B. HALT AT ADDRESS 001746 (1750)

AT THIS HALT, ENTER THE OCTAL NUMBER OF TUBES YOU HAVE (1 THROUGH 4) INTO THE SWITCH REGISTER. ALSO IF YOU WISH TO ENTER THE ADDRESSES AND VECTORS OF THE FK, VT, OR DL11'S, SET SWR (SWITCH REGISTER) BIT 15 TO A 1, OTHERWISE LEAVE SWR BIT 15=0 (CLEARED) AND DEFAULT ADDRESS. AND VECTORS WILL BE USED.

IF YOU INDICATED IN HALT 'B', VIA SETTING SW15=1, THAT DEFAULT ADDRESSES ARE NOT TO BE USED, THEN HALTS 'C' THRU 'G' WILL BE EXECUTED. THIS WILL ENABLE YOU TO SELECT YOUR OWN DEVICE, VECTOR & DL11 ADDRESSES.

NOTE: THIS SERIES OF HALTS WILL OCCUR SEQUENTIALLY FOR EACH TUBE THAT YOU INDICATED SO PAY HEED TO THE HALT ADDRESS - PLEASE.

C. HALT AT ADDRESS 002010 (2012)

ENTER FK VECTOR OF UNIT N INTO THE SWR.

D. HALT AT ADDRESS 002022 (2024)

ENTER FK ADDRESS OF UNIT N INTO THE SWR.

*. AT THIS POINT HALTS C AND D REPEATED FOR ALL TUBES.

E. HALT AT ADDRESS 002050 (2052)

ENTER THE VT VECTOR OF TUBE N - INTO THE SWITCH REGISTER.

F. HALT AT ADDRESS 002062 (2064)

ENTER THE VT ADDRESS OF TUBE N INTO THE SWR

*. HALTS E AND F REPEATED FOR ALL TUBES.

G. HALT AT ADDRESS 002114 (2116)

ENTER DL11 VECTOR ADDRESS FOR TUBE N INTO SWR.

*. HALT G REPEATED FOR ALL TUBES.

H. HALT AT ADDRESS 002136 (2140)

AT THIS HALT ENTER INTO THE SWR THE OCTAL NUMBER OF THE TUBE TO BE TESTED (0,1,2,3 OR 4). IF OCTAL 4 IS ENTERED IN THE SWR, ALL UNITS WILL

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

BE AUTOMATICALLY ALTERNATED AFTER PASS COMPLETION.

I. HALT AT ADDRESS 002634 (2636)

THIS HALT IS FOR THE TEST NUMBER TO BE RUN. ENTER THE NUMBER AND PRESS CONTINUE.

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

5. RESTART PROCEDURE

THE PROGRAM SHOULD BE RESTARTED AT THE SAME LOCATION THAT IT WAS ORIGINALLY STARTED UNLESS A NEW TUBE IS TO BE TESTED. IN THIS CASE TYPE A '↑A' IF A TELETYPE IS BEING USED. OTHERWISE, HALT AND RESTART THE PROGRAM AT LOCATION '210'.

6. CONSOLE SWITCH SETTINGS

A. REFER TO THE INDIVIDUAL TEST DESCRIPTIONS FOR APPLICABLE 'SWR' SETTINGS.

7. CONTROL SWITCHES (TELETYPE)

A. CONTROL <C>
TYPING '↑C' (OBTAINED VIA TYPING THE 'CNTR' & 'C' KEYS SIMULTAEOUSLY) AT ANY TIME ENABLES THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR.

B. CONTROL <A>
TYPING '↑A' (OBTAINED VIA TYPING THE 'CNTRL' AND 'A' KEYS SIMULTANEOUSLY) AT ANY TIME CAUSES A RESTART OF THE CURRENT TEST BEING RUN.

8. PROGRAM TEST TABLE & DIRECTORY

TEST NAME	TEST NO.	DESCRIPTION
-----	-----	-----
VT LOGIC	00	9
VT CHARACTER	01	10
VT FIELD MODE	02	11
VT CSR PRESET	03	12
VT END OF LINE	04	13
VT END OF SCREEN	05	14
VT BLANK CONTROL	06	15
VT ALIGNMENT	07	16
VT FOCUS	10	17
VT WORST CASE	11	18
VT CURSOR CONTROL	12	19
VT ODD ADDRESS	13	20
VT RUN-ALL	14	21
ASCII KBD LOGIC	15	22
ASCII KBD SWITCH & LIGHT	16	23
ASCII KBD CHARACTER SET	17	24

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

ASCII KBD REPEATIBILITY	20	25
VT/KBD SYSTEM TEST	21	26
VT/KBD SUBTEST SELECTOR	22	27
EXECUTION TIMES		28

9. VT LOGIC TEST (0)

A. TEST DESCRIPTION

THE 'VT' LOGIC TEST CONSISTS OF '28' SUBTESTS WHICH EXERCISE AND DIAGNOSE THE 'VT' LOGIC. EACH SUBTEST IS LOOPED '100' TIMES TO TEST LOGIC RELIABILITY. THESE TESTS ARE RUN AS A UNIT WHEN THE 'VT' LOGIC TEST IS SELECTED OR MAY BE SELECTED INDIVIDUALLY BY THE FK/VT SUBTEST SELECTOR (REFER TO SECTION 27). AT THE END OF EACH SUBTEST THE 'AUDIO BEEP' IS SOUNDED TO INDICATE A NEW SUBTEST IS BEING EXECUTED.

B. LOGIC ERRORS

THERE ARE TWO ERROR REPORTING SCHEMES USED FOR REPORTING 'VT' LOGIC ERRORS, (1) WITH TELETYPE AND (2) WITH PROGRAM HALTS.

- ON ENCOUNTERING A LOGIC ERROR (ALL DATA SWITCHES DOWN) THE FAILING SUBTEST NUMBER, ERROR ADDRESS, AND CONTENTS OF THE 'VT' REGISTERS ARE TYPED OUT. THERE ARE TWO ERROR FORMATS USED, (A) NORMAL 'VT' LOGIC ERROR OR (B) 'VT' SHIFT LOGIC ERROR.

(A). NORMAL ERROR FORMAT

TST NO.	PASS	MA	CSR	MAR
A	B	C	D	E

WHERE:

- A= FAILING SUBTEST NO.
- B= PASS ON WHICH ERROR OCCURRED.
- C= ERROR ADDRESS (MEMORY ADDRESS)
- D= CONTENTS OF CONTROL & STATUS ADDRESS REGISTER
- E= CONTENTS OF MAINTENANCE ADDRESS REGISTER*

*'SWR' BITS '2-0' MAY BE SET TO SELECT ANY ONE OF THE 'B' ADDRESSES THAT CAN BE READ BACK VIA THE MAINTENANCE REGISTER.

'SWR' 2-0	ADDRESS DATA SELECTED
-----	-----

000	COMMAND AND STATUS REGISTER
-----	-----------------------------

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

001	DATA ADDRESS REGISTER
010	SHIFT REG. INPUT DATA(EVEN ROW)
011	SHIFT REG. OUTPUT DATA(EVEN ROW)
100	CURSOR ADDRESS REGISTER
101	STARTING ADDRESS REGISTER
110	SHIFT REG. INPUT DATA(ODD ROW)
111	SHIFT REG. OUTPUT DATA(ODD ROW)

(B). SHIFT LOGIC ERROR FORMAT

TST NO.	MA	SHIFT	IN/OUT	EXPT'D	RECV'D
A	B	C	D	E	F

WHERE:

A= FAILING SUBTEST NO.
B= ERROR ADDRESS (MEMORY ADDRESS).
C= THE SHIFT IN OCTAL ON WHICH ERROR OCCURRED (1-32).
D= DIRECTION OF DATA SHIFT
E= EXPECTED DATA (SHIFTED IN FROM SHIFT TABLE).
F= RECEIVED DATA (CONTENTS OF 'MAR').

2. ON ENCOUNTERING A LOGIC ERROR (WITH DATA SW15 UP) THE PROGRAM WILL SOUND THE AUDIO BEEP AND LOAD THE INTERNAL REGISTERS WITH ERROR DATA AND HALT. THESE REGISTERS ARE SET UP IN ONE OF TWO WAYS: (A) NORMAL LOGIC ERROR AND (B) SHIFT LOGIC ERROR.

(A). NORMAL LOGIC ERROR

R0= FAILING SUBTEST NO.
R1= PASS ON WHICH ERROR OCCURRED
R2= ERROR ADDRESS
R3= CONTENTS OF CONTROL & STATUS ADDRESS REG.
R4= CONTENTS OF MAINTENANCE ADDRESS REG *(REFER ABOVE).

(B). SHIFT LOGIC ERROR

R0= FAILING SUBTEST NO.
R1= PASS ON WHICH ERROR OCCURED
R2= ERROR ADDRESS
R3= THE SHIFT IN OCTAL ON WHICH ERROR OCCURRED (1-100).
R4= EXPECTED DATA (SHIFTED IN FROM THE SHIFT TABLE).
R5= ACTUAL CONTENTS OF MAINTENANCE ADDRESS REG.

C. CONSOLE SWITCH SETTINGS

FUNCTION

SW11=0
SW11=1

NORMAL RUN [100 PASSES/TEST]
SUPPRESS SUBTEST ITERATIONS

SW12=0
SW12=1

CONTINUE SHIFTING ON SHIFT ERROR
ABORT SHIFTING ON SHIFT ERROR

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

SW13=0	PRINT ERROR MESSAGES
SW13=1	INHIBIT ERROR TYPEOUT
SW14=0	NORMAL RUN
SW14=1	LOOP ON CURRENT SUBTEST
SW15=0	CONTINUE ON ERROR
SW15=1	HALT ON ERROR

10. VT ' CHARACTER SET ' TEST (1)

A. TEST DESCRIPTION

THIS TEST DISPLAYS THE 'VT' CHARACTER SET IN NUMERICAL ORDER. THE CHARACTER SET IS REPEATED UNTIL THE ENTIRE SCREEN IS FILLED. THE TEST STARTS DISPLAYING A BLANK SCREEN. AS EACH CHARACTER IS CREATED IT IS DISPLAYED FOR APPROXIMATELY ONE SECOND AND THEN THE NEXT CHARACTER IS DISPLAYED. THE DISPLAY BUFFER IS SET UP SO THAT EACH LINE (EXCEPT THE LAST LINE) IS TERMINATED WITH EITHER VISIBLE 'END OF LINE' OR VISIBLE 'END OF PARAGRAPH'. THE LAST LINE IS TERMINATED WITH 'VISIBLE END OF SCREEN'. WHEN THE ENTIRE SCREEN HAS BEEN FILLED THE TEST IS RESTARTED.

NOTE: THE CHARACTER SET MAY CONTAIN BLANK SPACES. THESE ARE CODES RESERVED FOR FUTURE SPECIAL TYPESETTING CHARACTERS.

B. CONSOLE SWITCH SETTINGS	FUNCTION
-----	-----
SW11=0	NORMAL RUN
SW11=1	INHIBIT DISPLAY DELAY
SW12=0	NORMAL RUN
SW12=1	ADVANCE TO NEXT*
SW14=0	NORMAL RUN
SW14=1	HOLD ON CURRENT CHARACTER

*THIS ADVANCE WILL ONLY TAKE PLACE AFTER THE CURRENT TEST HAS BEEN COMPLETED.

11. VT ' FIELD CONTROL ' TEST (2)

A. TEST DESCRIPTION

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

THIS TEST DISPLAYS THE 'VT' CHARACTER SET USING ALL FIVE 'FIELD CONTROL' MODES: NORMAL, BLINK, BOLD, BLANK AND UNDERLINE. THE TEST BREAKS THE CHARACTER SET INTO THREE PARTS. EACH PART IS THEN DISPLAYED IN ALL 'S' MODES. THIS ENABLES THE USER TO VIEW EACH CHARACTER IN A STRAIGHT LINE IN EVERY MODE.

B.	<u>CONSOLE SWITCH SETTING</u>	<u>FUNCTION</u>
	SW12=0	NORMAL RUN
	SW12=1	ADVANCE TO NEXT TEST

12. VT 'CSR PRESET' TEST (3)

A. TEST DESCRIPTION

THIS TEST CONSISTS OF FOUR (4) PARTS. EACH PART DISPLAYS AN ENTIRE SCREEN OF THE 'VT' CHARACTER SET USING A 'CSR' PRESET FUNCTION. THE SELECTED PRESET FUNCTIONS IN ORDER ARE: BOLD, BLINK, BLANK, & UNDERLINE. AN IDENTIFYING MESSAGE WILL PRECEDE EACH PART.

B.	<u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
	SW12=0	NORMAL RUN
	SW12=1	ADVANCE TO NEXT TEST*
	SW14=0	NORMAL RUN
	SW14=1	HALT ON CURRENT PRESET

13. VT 'END OF LINE' TEST (4)

A. TEST DESCRIPTION

THIS TESTS THAT THE 'VISIBLE END OF LINE' CHARACTER CAN BE PLACED IN ANY OF THE '1024' CHARACTER POSITIONS ON THE SCREEN. THIS IS DONE BY STARTING ALL '16' LINES WITH A VISIBLE END OF LINE CHARACTER AND THEN INCREMENTING THESE CHARACTERS ACROSS THE SCREEN. AS THE 'EOL' CHARACTER IS MOVED AN ASTERISK IS DISPLAYED IN ITS PLACE.

B.	<u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
	SW11=0	NORMAL RUN

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

SW11=1	INHIBIT DISPLAY DELAY
SW12=0	NORMAL RUN
SW12=1	ADVANCE TO NEXT TEST*
SW14=0	NORMAL RUN
SW14=1	HOLD CURRENT 'EOL' POSITION

* THIS ADVANCE WILL ONLY TAKE PLACE AFTER THE CURRENT TEST HAS BEEN COMPLETED.

14. VT 'END OF SCREEN' TEST (5)

A. TEST DESCRIPTION

THIS TEST TESTS THAT THE 'VISIBLE END OF SCREEN' CHARACTER CAN BE PLACED IN ANY OF THE '1024' CHARACTER POSITIONS ON THE SCREEN. THIS IS DONE BY FIRST PRE-LOADING THE DATA BUFFER WITH QUESTION MARKS (THESE WILL BE DISPLAYED IF THE EOS IS SKIPPED). THE 'EOS' CHARACTER IS THEN INCREMENTED THRU THIS BUFFER REPLACING THESE QUESTION MARKS BEHIND THE 'EOS' WITH DOTS.

B. CONSOLE SWITCH SETTINGS	FUNCTION
-----	-----
SW11=0	NORMAL RUN
SW11=1	INHIBIT DISPLAY DELAY
SW12=0	NORMAL RUN
SW12=1	ADVANCE TO NEXT TEST
SW14=0	NORMAL RUN
SW14=1	HOLD CURRENT 'EOS' POSITION

15. VT 'BLANK CONTROL' TEST (6)

A. TEST DESCRIPTION

THIS TEST TESTS THE 'BLANK' FIELD CONTROL LOGIC BY INCREMENTING THE 'BLANK' CONTROL CHARACTER THRU A BUFFER OF ASTERISK CHARACTERS. THE TEST STARTS USING THE BLANK CONTROL CHARACTER AT THE START OF THE 'VT' BUFFER. THE SCREEN SHOULD START OFF BLANK AND AS THE BLANK CONTROL CHARACTER IS INCREMENTED THRU THE 'VT' BUFFER ASTERISKS

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

WILL START APPEARING ON THE SCREEN. THE TEST WILL RESTART
AFTER THE ENTIRE SCREEN HAS BEEN FILLED.

B.	<u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
	SW11=0	NORMAL RUN
	SW11=1	INHIBIT DISPLAY DELAY
	SW12=0	NORMAL RUN
	SW12=1	ADVANCE TO NEXT TEST
	SW14=0	NORMAL RUN
	SW14=1	HOLD CURRENT BLANK POSITION

16. VT 'ALIGNMENT' TEST (7)

A. TEST DESCRIPTION

THIS TEST DISPLAYS A FULL SCREEN OF THE CHARACTER 'E' TO
ENABLE A VISUAL ALIGNMENT OF THE 'VT'.

B.	<u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
	SW12=0	NORMAL RUN
	SW12=1	ADVANCE TO NEXT TEST

17. VT 'FOCUS' TEST (10)

A. TEST DESCRIPTION

THIS TEST DISPLAYS A FULL SCREEN OF ALTERNATE ASTERISKS
AND ONE'S TO ENABLE A VISUAL FOCUS OF THE 'VT'.

B.	<u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
	SW12=0	NORMAL RUN
	SW12=1	ADVANCE TO NEXT TEST

18. VT 'WORST CASE' TEST (11)

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

A. TEST DESCRIPTION

THIS TEST DISPLAYS A FULL SCREEN OF THE WORST CASE CHARACTER PATTERN '125 & 252'.

NOTE: CODE 125 WILL DISPLAY A "U" AND CODE 252 WILL DISPLAY EITHER A BLANK OR A SPECIAL TYPESETTING CHARACTER, THIS BEING DEPENDENT ON THE CHARACTER SET.

B. CONSOLE SWITCH SETTING FUNCTION

SW12=0	NORMAL RUN
SW12=1	ADVANCE TO NEXT TEST

19. VT 'CURSOR CONTROL' TEST (12)

A. TEST DESCRIPTION

THIS TEST TESTS THAT THE CURSOR CONTROL CHARACTER CAN BE PLACED IN ANY OF THE '1024' CHARACTER POSITIONS ON THE SCREEN. THIS IS DONE BY DISPLAYING A 'CURSOR EOS' AND INCREMENTING IT THRU A PRE-LOADED BUFFER OF QUESTION MARKS. AS THE CURSOR CONTROL CHARACTER IS MOVED ALONG ASTERISKS ARE FILLED IN BEHIND THE CURSOR CHARACTER. WHEN THE TEST IS COMPLETE THE ENTIRE SCREEN SHOULD BE FILLED WITH ASTERISKS AND WITH NO QUESTION MARKS VISIBLE.

B. CONSOLE SWITCH SETTINGS FUNCTION

SW11=0	NORMAL RUN
SW11=1	INHIBIT DISPLAY DELAY
SW12=0	NORMAL RUN
SW12=1	ADVANCE TO NEXT*
SW14=0	NORMAL RUN
SW14=1	HOLD ON CURRENT CHARACTER

* THIS ADVANCE WILL ONLY TAKE PLACE AFTER THE CURRENT TEST HAS BEEN COMPLETED.

20. VT 'ODD ADDRESS' TEST (13)

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

A. TEST DESCRIPTION

THIS TEST SIMPLY DISPLAYS A MESSAGE WHICH IS LOADED FROM AN ODD STARTING ADDRESS. THE MESSAGE SHOULD SAY: THIS MESSAGE IS BEING LOADED FROM AN ODD STARTING ADDRESS. HOW DOES IT LOOK?

21. VT 'RUNALL' TEST (14)

A. TEST DESCRIPTION

THIS TEST ENABLES TESTS '0-13,15' TO BE RUN IN ORDER. IF A '4' WAS SELECTED AS THE TUBE NUMBER, THESE TESTS WILL BE RUN SEQUENTIALLY ON ALL AVAILABLE TUBES.

22. ASCII KBD 'LOGIC' TEST (15)

A. TEST DESCRIPTION

THE 'KBD' LOGIC TEST CONSISTS OF '19' SUBTESTS WHICH EXERCISE AND DIAGNOSE THE 'KBD' LOGIC. EACH SUBTEST IS LOOPED '100' TIMES TO TEST LOGIC RELIABILITY. THESE TESTS ARE RUN AS A UNIT WHEN THE 'KBD' LOGIC TEST IS SELECTED BUT THEY MAY ALSO BE SELECTED INDIVIDUALLY BY THE 'FK/VT' SUBTEST SELECTOR (REFER TO SECTION 27).

THE 'KBD' LOGIC TEST DOES 'NOT' CHECKOUT THE LIGHT AND SWITCH LOGIC. THIS LOGIC IS CHECKED BY THE 'KBD FUNCTION LIGHT & SWITCH TEST' (SECTION 23).

B. RESTRICTIONS

THE 'VT' LOGIC MUST BE OPERATIONAL AS REQUESTS FOR 'KBD' KEYBOARD INPUTS ARE REQUESTED BY MESSAGES DISPLAYED ON THE 'VT'. CARE SHOULD BE TAKEN THAT ONLY ONE KEY BE STRUCK EACH TIME THIS REQUEST IS MADE TO AVOID FALSE ERROR INDICATIONS.

C. LOGIC ERRORS

THERE ARE TWO ERROR REPORTING SCHEMES USED FOR REPORTING 'KBD' LOGIC ERRORS, (1) WITH TELETYPE AND (2) WITH PROGRAM HALTS.

- 1. ON ENCOUNTERING A LOGIC ERROR (ALL DATA SWITCHES DOWN)

THE FAILING SUBTEST NO., THE ERROR ADDRESS AND
CONTENTS OF THE 'KBD' REGISTERS ARE TYPED OUT.

A. ERROR FORMAT

TST NO.	PASS	MA	CSR	DATA
A	B	C	D	E

WHERE:
 A= FAILING SUBTEST NO.
 B= PASS ON WHICH ERROR OCCURRED.
 C= ERROR ADDRESS (MEMORY ADDRESS)
 D= CONTENTS OF CONTROL & STATUS ADDRESS REGISTER
 E= CONTENTS OF DATA ADDRESS REGISTER.

2. ON ENCOUNTERING A LOGIC ERROR (DATA SW15 UP)
THE PROGRAM LOADS THE INTERNAL REGISTERS WITH
THE FOLLOWING ERROR DATA AND HALTS.

R0= FAILING SUBTEST NO.
 R1= PASS ON WHICH ERROR OCCURRED.
 R2= FAILING ERROR ADDRESS
 R3= CONTENTS OF CONTROL & STATUS REGISTER ADDRESS
 R4= CONTENTS OF THE DATA ADDRESS REGISTER.

D. <u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
SW10=0	REQUEST KEYWARD INPUT
SW10=1	INHIBIT KEYWARD INPUT REQUEST
SW11=0	NORMAL RUN [100 PASSES/TEST]
SW11=1	SUPPRESS SUBTEST ITERATIONS
SW13=0	PRINT ERROR MESSAGES
SW13=1	SUPPRESS ERROR MESSAGES
SW14=0	NORMAL RUN
SW14=1	LOOP ON CURRENT SUBTEST
SW15=0	CONTINUE ON ERROR
SW15=1	HALT ON ERROR

23. ASCII KBD 'LIGHT & SWITCH' TEST (16)

A. TEST DESCRIPTION

778
779
780
781
782
783
784
795
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

THIS TEST IS USED TO CHECK OUT THE KBD FUNCTION LIGHT AND SWITCH LOGIC. THIS IS AN OPERATOR INTERVENTION TEST WHICH MONITORS THE FUNCTION KEYS. WHEN A FUNCTION KEY OR KEYS ARE HELD DOWN ITS CORRESPONDING FUNCTION LIGHT IS LIT. IF THE KEY IS HELD DOWN A SECOND TIME THE CORRESPONDING FUNCTION LITE IS TURNED OFF. IT SHOULD BE A NOTED THAT THESE ARE INTERVAL SWITCHES AND THE SIGNAL LEVEL WHICH INDICATES THAT A KEY IS PRESSED WILL REMAIN TRUE FOR THE LENGTH OF TIME THE KEY IS HELD DOWN. SINCE THERE IS NO WAY FOR THE PROGRAM TO TELL IF A KEY IS BEING HELD DOWN OR IF IT HAS BEEN STRUCK A SECOND TIME THE LIGHT DATA WILL BE COMPLIMENTED EVERY TIME THE SIGNAL IS PRESENT. TO ALLEVIATE THIS PROBLEM THE PROGRAM ONLY READS THE SWITCHES EVERY '512' INTERVAL TIME FLAGS. SO IF A SWITCH IS HELD DOWN IT WILL BE TURNED ON AT THE BEGINNING OF THIS COUNT AND TURNED OFF AT THE END OF THE COUNT IF THE SWITCH REMAINS DOWN. TO SWITCH TO TESTING THE NEXT TUBE (IF RUNALL IS SELECTED) TYPE 'C' ON THE TUBE BEING TESTED.

24. ASCII KBD 'CHARACTER SET' TEST (17)

A. TEST DESCRIPTION

THIS TEST IS ALSO AN OPERATOR INTERVENTION TEST. IT IS USED TO VERIFY THAT ALL CHARACTERS CAN BE TRANSMITTED BY THE ASCII KEYBOARD. THE TEST REQUESTS THREE COMPLETE PASSES (LOWER CASE, UPPER CASE AND CONTROL) TO BE TYPED IN BY THE OPERATOR. A MESSAGE IS DISPLAYED ON THE 'VT' FOR THE CASE TO BE INPUTED. EACH PASS MUST START IN THE UPPER LEFT HAND CORNER OF THE KEYBOARD AND END WITH SPACE. AS EACH CHARACTER IS TRANSMITTED FROM THE KEYBOARD IT IS VERIFIED AGAINST A CHARACTER BUFFER AND DISPLAYED ON THE 'VT'.

B. ERRORS

IF THE TRANSMITTED CHARACTER CODE DOESN'T MATCH THE EXPECTED CHARACTER CODE THE AUDIO 'BEEP' IS SOUNDED AND THE RECEIVED CHARACTER IS DISPLAYED IN 'BOLD-UNDERLINED'.

C. CONSOLE SWITCH SETTINGS FUNCTION

SW14=0	NORMAL RUN
SW14=1	LOOP ON THE CURRENT CHARACTER
SW15=0	LOOP UNTIL CORRECT CODE RECEIVED
SW15=1	SKIP BAD CHARACTER

NOTE: FOR PROPER OPERATION SW15 SHOULD BE RE-SET TO '0'
AS SOON AS THE NEXT CHARACTER HAS BEEN INPUTED.

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

25. ASCII KBD 'REPEATIBILITY' TEST (20)

A. TEST DESCRIPTION

THIS TEST IS ALSO AN OPERATOR INTERVENTION TEST. THE TEST HAS TWO FUNCTIONS. IT CAN BE USED AS A BACKUP FOR THE 'CHARACTER SET' TEST TO TEST ANYONE SELECTED CHARACTER CODE OR, SECOND, IT CAN BE USED TO TEST KEYBOARD REPEATIBILITY BETWEEN ANY TWO SELECTED CHARACTER CODES.

B. STARTING SEQUENCE

THE TEST VERIFIES UP TO TWO CHARACTERS RECEIVED FROM THE KEYBOARD AGAINST THE CHARACTER CODE(S) LOADED BY THE OPERATOR. THESE CHARACTER CODES ARE LOADED ONE OF TWO WAY, (1) VIA TELETYPE OR (2) VIA 'SWR' ON PROGRAM HALTS.

- 1. TYPE: NNN[,NNN] (CR) (I.E. 101,141 UPPER CASE A AND LOWER A)

WHERE:
NNN IS THE CODE(S) TO BE VERIFIED.

- 2. WHEN TEST IS SELECTED THE PROGRAM WILL HALT. LOAD THE CHARACTER CODE TO BE VERIFIED IN THE 'SWR' AND PRESS CONTINUE. A SECOND HALT WILL OCCUR. IF TWO CHARACTERS ARE TO BE VERIFIED LOAD THE CODE OF THE SECOND CHARACTER IN THE 'SWR', OTHERWISE JUST PRESS CONTINUE.

C. ERRORS

IF THE TRANSMITTED CHARACTER CODE DOESN'T MATCH THE EXPECTED CHARACTER CODE THE AUDIO 'BEEP' IS SOUNDED AND THE RECEIVED CHARACTER IS DISPLAYED IN 'BOLD-UNDERLINED'.

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
99126. VT20 SYSTEM TEST (21)
*****A. TEST DESCRIPTION

THIS TEST IS DESIGNED TO EXERCISE THE VT20 IN A SYSTEM TYPE OF ENVIRONMENT. THE TEST UTILIZES THE ENTIRE VT20 SYSTEM (ALL TUBES) IN CONJUNCTION WITH A HOST COMPUTER. THE HOST COMPUTER MAY BE EITHER A PDP-11 OR A PDP-8. THE HOST IS TO BE CABLED VIA SERIAL LINE INTERFACE TO THE VT20 SYSTEM. ONE INTERFACE LINE IS REQUIRED FOR EACH UNIT. THE HOST DIAGNOSTIC, EITHER MD-11-DZVTE (FOR A PDP-11 HOST W/DL11'S), MD-11-DZVTG (FOR A PDP-11 HOST W/DH11'S) OR MD-08-DIVTB (FOR A PDP-8 HOST W/KL8J'S) MUST BE LOADED AND RUNNING BEFORE ATTEMPTING TO TRANSFER DATA VIA THIS TEST.

WHEN TEST 21 IS SELECTED, A MESSAGE REMINDER FOR SETTING UP THE HOST DIAGNOSTIC WILL BE DISPLAYED. THIS MESSAGE WILL THEN MOMENTARILY BE REPLACED WITH A TUBE STATUS HEADER. IN THIS HEADER WILL BE DISPLAYED: NO. OF CHARACTERS ON THE SCREEN, NO. OF CHARACTERS RECEIVED FROM THE HOST, NO. OF BLOCKS TRANSFERRED AND NO. OF ERRORS ENCOUNTERED. THIS MESSAGE ALSO INDICATES THAT THE TEST IS READY.

AS CHARACTERS ARE RECEIVED FROM THE ASCII KEYBOARD, THEY ARE TESTED FOR BEING PROGRAM CONTROL CHARACTERS, AND IF NOT, THEY ARE TREATED AS DATA AND DISPLAYED. THE PROGRAM WILL ACCEPT UP TO 384 DATA CHARACTERS OR 6 COMPLETE LINES.

B. DATA CREATION

DATA CAN BE AUTOMATICALLY GENERATED BY TYPING EITHER A (↑A) OR A (↑W). THE (↑A) WILL ENABLE FOR A INCREMENTAL CHARACTER SET OF THE CHARACTERS '40-177' TO BE CREATED UNTIL THE SCREEN BUFFER IS FULL. A (↑W) WILL ENABLE FOR A WORST CASE PATTERN OF '125-252' TO BE PRODUCED UNTIL THE SCREEN BUFFER IS FILLED.

C. TRANSMITTING MODES

THERE ARE TWO MODES USED TO TRANSMIT DATA TO THE HOST COMPUTER: SINGLE TRANSFER OR CONTINUOUS TRANSFER.

A SINGLE TRANSFER IS INITIATED BY TYPING 'ALTMODE' AFTER THE DATA HAS BEEN INPUTTED. ON RECEIVING THE ALTMODE THE INPUTED DATA IS TRANSFERED TO THE HOST COMPUTER. THE DATA IS BUFFERED THERE AND TRANSMITTED BACK TO THE VT20 ON RECEIVING THE END OF PARAGRAPH. AS THE DATA IS RECEIVED IT IS DISPLAYED BELOW THE ORIGINAL DATA. AFTER EXAMINING THE DATA THE OPERATOR MAY TYPE '↑C' TO

992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047

CLEAR THE SCREEN AND RESTART THE TEST OR START A CONTINUOUS TRANSFER.

A CONTINUOUS TRANSFER IS INITIATED AFTER THE DATA HAS BEEN INPUTTED VIA TYPING A '^T'. THIS MODE OPERATES IN THE SAME MANNER AS A SINGLE TRANSFER EXCEPT UPON RECEIVING THE DATA BACK FROM THE HOST COMPUTER IT IS AUTOMATICALLY RE-TRANSMITTED. TYPING A SECOND '^T' WILL STOP THE CONTINUOUS TRANSFER AFTER THE CURRENT TRANSFER HAS BEEN COMPLETED.

THIS DATA IS SENT IN THE FORMAT OF: 4 NULL CHARACTERS (000), A START CODE CHARACTER (377), DATA (ORIGINATED BY THE USER), AND A END OF PARAGRAPH CHARACTER (14). ON RECEIPT OF THE 'EOP' (END OF PARAGRAPH) CHARACTER, THE HOST COMPUTER WILL TRANSMIT THE DATA, IN THE FORMAT IT WAS RECEIVED, BACK TO THE VT20. ON RECEIPT OF THE START CODE CHARACTER (377), THE VT20 DISPLAYS A UP-ARROW(^) TO INDICATE THE DATA HAS RETURNED.

EACH CHARACTER IS VERIFIED AS IT IS RECEIVED AGAINST CORRESPONDING CHARACTER TRANSMITTED. IF A RECEIVED CHARACTER DOESN'T MATCH THE CORRESPONDING TRANSMITTED CHARACTER, IT IS DISPLAYED AS 'BOLD-UNDERLINED'. THIS ERROR WOULD ALSO BE RECORDED IN THE ERROR HEADER.

ERROR CHECKING CAN BE INHIBITED BY SETTING DATA 'SW00' TO A '1'(UP). ALL DATA THEN RECEIVED FROM THE HOST COMPUTER WILL SIMPLY BE DISPLAYED. THIS SWITCH SHOULD ONLY BE SET WHEN THE HOST COMPUTER IS ORGINATING DATA USING THE SEND MODE (REFER TO THE HOST DIAGNOSTIC).

THERE IS ALSO ANOTHER OPTION AVAILABLE TO THE USER RUNNING WITH MD-11-DZVTG.

SW00 & SW01 CAN BE SET. THIS WILL ENABLE THE DATA BEING RECEIVED FROM THE HOST WILL BE DISPLAYED IN OCTAL FORM. THE DATA WILL ALSO BE ECHOED BACK TO THE HOST. THIS OPTION IS USED WHEN RUNNING THE HOST VERIFY TEST. REFER TO MD-11-DZVTG FOR FULL DETAILS OF THE TEST.

IF A TRANSFER IS INITIATED AND FOR ONE REASON OR ANOTHER IT FAILS TO RETURN BACK TO THE VT20, THE TEST WILL APPEAR HUNG SINCE NO KEYBOARD COMMANDS WILL BE RECOGNIZED. IN THIS CASE AND ONLY IN THIS CASE TYPE '^E' TO ESCAPE THE TRANSFER MODE AND RESTART.

IF '^E' IS TYPED IN THE MIDDLE OF A TRANSFER, IT WILL CAUSE MULTIPLE RECEIVER ERRORS.

C.	CONSOLE SWITCH SETTINGS	FUNCTION
	-----	-----
	SW15=0	*CONTINUE TRNANSFERING ON ERROR
	SW15=1	HALT TRANSFERING ON ERROR
	SW00=0	VERIFY RECEIVER DATA
	SW00=1	**INHIBIT VERIFYING RECEIVER DATA

1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103

SW00 & SW01=1

***DISPLAY RECEIVED CHARACTER IN
OCTAL FORM AND ECHO IT BACK TO
THE HOST.

- * SW15 APPLIES TO CONTINUOUS TRANSFERS ONLY AND THE HALT WILL OCCUR AT THE END OF THE CURRENT TRANSFER.
- ** THIS SWITCH MUST BE SET IF THE HOST IS TRANSMITTING DATA USING THE SEND MODE.
- *** THIS OPTION SHOULD ONLY BE USED WHEN RUNNING THE VERIFY TEST OF MD-11-DZVTG.

D. ASCII KRB CONTROL SWITCHES

KEY	FUNCTION
RUBOUT	DELETE LAST CHARACTER.
ALTMODE	INITIATE SINGLE TRANSFER.
↑A (CTRL A)	GENERATE INCREMENTAL CHARACTER SET.
↑T (CTRL T)	ENABLE/DISABLE CONTINUOUS TRANS.
↑W	GENERATE WORST CASE CHARACTER PATTERN
↑C (CTRL C)	CLEAR SCREEN AND RESTART
↑E (CTRL E)	*ESCAPE AND RESTART

* USE WITH CAUTION AS NOTED ABOVE.

E. ERRORS

TO INDICATE TRANSMITTER, RECEIVER AND KEYBOARD ERRORS THE FOLLOWING CHARACTERS WILL BE DISPLAYED USING THE BLINK MODE.

CHARACTER	MEANING
K	ASCII KBD ERROR FLAG SET (OVERRUN ERROR)
T	ILLEGAL TRANSMITTER INTERRUPT
R	ILLEGAL RECEIVER INTERRUPT
O	DL11 RECEIVER 'OVERRUN' ERROR
F	DL11 RECEIVER 'FRAMING' ERROR
P	DL11 RECEIVER 'PARITY' ERROR

27. VT / K B D S U B T E S T S E L E C T O R (22)

A. TEST DESCRIPTION

THIS TEST IS DESIGNED TO ALLOW THE OPERATOR TO LOOP ON ANY SELECTED 'KBD' OR 'VT' LOGIC SUBTEST WITHOUT

1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159

RUNNING THE ENTIRE LOGIC TEST

B. STARTING SEQUENCE

THIS TEST ACCEPTS AN OCTAL ADDRESS OF A 'SUBTEST' TO BE EXECUTED. THE PROGRAM THEN SETS UP THE PROPER PARAMETERS AND EXECUTES THE SELECTED 'SUBTEST'. THE SUBTEST ADDRESS IS LOADED ONE OF TWO WAYS, (1) VIA TELETYPE OR (2) VIA 'SWR' ON PROGRAM HALT.

1. WHEN SELECTED THE PROGRAM WILL TYPE THE MESSAGE 'TEST ADDR.?'. THE OPERATOR SHOULD THEN TYPE IN THE 'SCOPE ADDRESS' OF THE TEST TO BE EXECUTED.
2. WHEN THE TEST IS SELECTED THE PROGRAM WILL HALT. LOAD THE ADDRESS OF THE TEST TO BE EXECUTED INTO THE 'SWR' AND PRESS CONTINUE.

C. RESTRICTIONS

DATA "SW11" MUST BE DOWN [0] TO INHIBIT THE PROGRAM FROM ESCAPING THE SELECTED TEST AND CONTINUING THRU THE LOGIC TEST.

D. ERRORS

REFER TO THE RESPECTIVE 'KBD' OR 'VT' ERROR SECTION.

29. EXECUTION TIMES

TEST	PASS TIME	NOTES
----	-----	-----
0	7.5 MINS	
1	4 MINS	
2	N/A	*
3	30 SECS	
4	25 SECS	
5	6 MINS	
6	6 MINS	
7	N/A	*
10	N/A	*
11	N/A	*
12	6 MINS	
13	N/A	*
14	TOTAL OF ABOVE + TEST 15	
15	75 SECS	
16	N/A	
17	N/A	

1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215

20 N/A
21 N/A

* IN RUN-ALL (TEST 14), THESE RUN FOR 3 SECS EACH.

%

```
.TITLE VT20 SYSTEM DIAGNOSTIC TEST PROGRAM
;MAINDEC-11-DBVTA-D-D
;COPYRIGHT 1972,1976
;DIGITAL EQUIPMENT CORP. MAYNARD MASS. 01754
```

;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:

```
SW15=100000      ;=1, HALT ON ERROR
SW14=40000       ;=1, LOOP ON CURRENT TEST
SW13=20000       ;=1, SUPPRESS ERROR TYPEOUT
SW12=10000       ;=1, HALT SHIFTING ON SHIFT ERROR.
SW11=4000        ;=1, SUPPRESS 'SUBPROGRAM' ITERATIONS
SW10=2000        ;=1, INHIBIT ASCII KRB MANUAL INTERVENTION
SW09=1000
SW08=400
SW07=200
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1
```

;REGISTER DEFINITIONS

```
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7
```

;INSTRUCTIONS DEFINITIONS

```
FKCSR4=%4
VTCSR4=%4
VTMAR5=%5
POP1SP=5726
PUSH1SP=5746
PUSH2SP=24646
POP2SP=22626
NOP=240
X=2
```

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

000000
000001
000002
000003
000004
000005
000006
000007

000004
000004
000005
005726
005746
024646
022626
000240
000002

```

1216          000002          Y=2
1217
1218          ;LOAD TRAP CATCHER INTO LOC'S 0-1000
1219          000000          .=0
1220          .REPT          200
1221          .+2
1222          4
1223          .ENDR
1224          000020          .=20
1225 000020 016456          ERTRAP          ;ERROR TRAP REPORTER ROUTINE
1226          000024          .=24
1227 000024 016014          PWRFAL          ;POWER FAIL HANDLER
1228 000026 000340          340
1229          000030          .=30
1230 000030 001200          EMTSRV          ;EMT TRAP, EMT DISPATCH SERVICE
1231 000032 000340          340
1232 000034 017446          FKERR          ;TRAP TRAP, LOGIC ERROR TRAP
1233 000036 000340          340
1234          000052          .=52
1235 000052 020000          20000
1236          000060          .=60
1237 000060 014350          XTTYIN          ;TELEPRINTER KEYBOARD ROUTINE
1238          000200          .=200
1239 000200 000137 001624          JMP          INITTY          ;PROGRAM STARTING ADDR. IF TTY AVAILABLE.
1240 000204 000137 001634          JMP          NOTTY          ;PROGRAM STARTING ADDR. IN 'NO' TTY.
1241 000210 000137 002136          JMP          INITBA          ;PROGRAM RE-START ADDR. TO SELECT NEW UNIT.
1242
1243          ;TRAP EQUIVALENCE TABLE:
1244
1245          104400          ERROR=TRAP          ;LOGIC TEST ERROR ROUTINE
1246          104000          DISPLAY=EMT          ;'VT' MESSAGE DISPLAY ROUTINE
1247          104CC1          SCOPE=EMT+1          ;LOGIC TEST SCOPE SUBROUTINE
1248          104002          SAVREG=EMT+2          ;SUBROUTINE TO SAVE 'R0-R5' ON STACK
1249          104003          GETREG=EMT+3          ;SUBROUTINE TO GET 'R0-R5' FROM STACK
1250          104004          DELAY=EMT+4          ;3 SEC. DISPLAY DELAY
1251          104005          EOSBUF=EMT+5          ;SUBROUTINE TO LOAD VT BUFER W/ 'EOS'
1252          104006          ENDTST=EMT+6          ;SUBROUTINE TO CHECK DATA SW.'S
1253          104007          SETLNE=EMT+7          ;SUBROUTINE TO SET UP A LINE TO CHAR.'S
1254          104010          LDLINE=EMT+10          ;SUBROUTINE TO LOAD A 'VT' BUFFER LINE
1255          104011          PRELOAD=EMT+11          ;SUBROUTINE TO PRE-LOAD 'VT' DATA BUFFER
1256          104012          PRINT=EMT+12          ;SUBROUTINE TO PRINT ASCII MESSAGES.
1257          104013          TTYIN=EMT+13          ;SUBROUTINE TO INPUT VIA KEYBOARD
1258          104014          PRTOCT=EMT+14          ;SUBROUTINE TO PRINT A 6 DIGIT OCTAL NO.
1259          104015          ASEMBL=EMT+15          ;SUBROUTINE TO ASSEMBLE CHARACTERS INTO OCTAL VALUE
1260          104016          SPACE=EMT+16          ;SUBROUTINE TO PRINT SPACES
1261          104017          TSTTKS=EMT+17          ;SUBROUTINE TO TEST FOR KEYBOARD FLAGS
1262          104020          DELAYL=EMT+20          ;SUBROUTINE TO SETUP A LONG DISPLAY DELAY
1263          104021          WAITSY=EMT+21          ;SUBROUTINE TO WAIT FOR VERTICAL SYNC
1264          104022          NULL=EMT+22          ;SUBROUTINE TO TRANSMIT A NULL PRINTER CHAR.
1265          104023          THEND=EMT+23          ;SUBROUTINE ENTERED AT END OF DISP. TEST
1266          104024          BINDEC=EMT+24          ;SUBROUTINE TO CONVERT 'BCD' TO DECIMAL
1267
1268          ;*****
1269          ;EMT DISPATCH SERVICE ROUTINE
1270          ;ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
1271          ;TO THE SELECTED SUBROUTINE.

```



```

1272
1273
1274 001200 011646
1275 001202 162716 000002
1276 001206 017616 000000
1277 001212 005716
1278 001214 001001
1279 001216 000000
1280 001220 006316
1281 001222 042716 177001
1282 001226 062716 001240
1283 001232 017616 000000
1284 001236 000136
1285
1286
1287
1288
1289 001240 017246
1290 001242 020272
1291 001244 016122
1292 001246 016176
1293 001250 015540
1294 001252 015656
1295 001254 015732
1296 001256 017024
1297 001260 015632
1298 001262 016252
1299 001264 014754
1300 001266 014350
1301 001270 015056
1302 001272 017104
1303 001274 014276
1304 001276 014330
1305 001300 015550
1306 001302 015476
1307 001304 017360
1308 001306 015726
1309 001310 016546
1310
1311
1312
1313 001312 177776
1314 001314 177560
1315 001316 177562
1316 001320 177564
1317 001322 177566
1318 001324 177570
1319 001326 177571
1320
1321
1322 001330 170330
1323 001332 170332
1324 001334 170334
1325 001336 170336
1326 001340 000320
1327 001342 000322

```

```

      . = 1200
EMTSRV: MOV      (SP), -(SP)      ; GET PC FOR TO RETURN
          SUB      #2, (SP)      ; PC OF EMT
          MOV      @ (SP), (SP)  ; GET EMT
          TST      (SP)          ; IS EMT VALID?
          BNE      EMTOK
          HALT
EMTOK:  ASL      (SP)            ; INVALID EMT
          BIC      #177001, (SP) ; MULTIPLY EMT ARG BY '2'
          ADD      #EMTTAB, (SP) ; CLEAR UNWANTED BITS
          MOV      @ (SP), (SP)  ; POINTER TO SUBROUTINE ADDRESS
          JMP      @ (SP)+       ; SUBROUTINE ADDRESS
                                       ; GO TO SUBROUTINE

```

;EMT DISPATCH TABLE

```

EMTTAB: VTMES      ; 'VT' MESSAGE DISPLAY ROUTINE
        SCOPEC     ; LOGIC TEST SCOPE ROUTINE
        XSAVRG    ; SUBROUTINE TO SAVE 'R0-R5' ON STACK
        XGETRG    ; SUBROUTINE TO GET 'R0-R5' FROM STACK
        XDELAY   ; SUBROUTINE TO SETUP 3 SEC. DISPLAY DELAY
        XEOSBF   ; SUBROUTINE TO LOAD VT BUFFER W/ VISIBLE 'EOS'
        XENDT    ; SUBROUTINE TO CHECK DATA SW.'S
        XSTLNE   ; SUBROUTINE TO SET UP A 'VT' BUFFER LINE
        XLDLNE   ; SUBROUTINE TO LOAD A 'VT' BUFFER LINE
        XPRELD   ; SUBROUTINE TO PRE-LOAD THE DATA BUFFER
        TYPMES   ; SUBROUTINE TO PRINT ASCII MESSAGES.
        XTTYIN   ; SUBROUTINE TO INPUT VIA KEYBOARD
        OCTPRT   ; SUBROUTINE TO PRINT A 6 DIGIT OCTAL NO.
        XASEMB   ; SUBROUTINE TO ASSEMBLE CHARACTERS INTO OCTAL VALUE
        XSPACE   ; SUBROUTINE TO PRINT SPACES
        TKSFLG   ; SUBROUTINE TO TEST FOR KEYBOARD FLAGS
        XDLAYL   ; SUBROUTINE TO SET UP LONG DISPLAY DELAY
        XWAITS   ; SUBROUTINE TO WAIT FOR VERT. SYNC
        XNULL    ; SUBROUTINE TO TRANSMIT A 'NULL' CHAR.
        XBINDEC  ; SUBROUTINE TO CONVERT 'BCD' TO DECIMAL

```

;REGISTER ADDRESSES

```

PSW:    177776 ; ADDRESS OF PROCESSOR STATUS REG.
TKS:    177560 ; ADDRESS OF KEYBOARD STATUS REG.
TKB:    177562 ; " " " " BUFFER "
TPS:    177564 ; " " " " PRINTER STATUS REG.
TPB:    177566 ; " " " " PRINTER BUFFER REG.
SWR:    177570 ; " " " " SWITCH REG.
SWRO:   177571 ; " " " " HIGH BYTE

```

; 'FK' UNIT 0'S REGISTER ADDRESSES.

```

FKOLDB: 170330 ; ADDRESS OF UNIT 0'S 'LIGHT DATA B'
FKOLDA: 170332 ; ADDRESS OF UNIT 0'S 'LIGHT DATA A'
FKOCSR: 170334 ; ADDRESS OF UNIT 0'S 'CONTROL & STATUS' REG.
FKODAT: 170336 ; ADDRESS OF UNIT 0'S 'KEYBOARD DATA' REG.
FKOINT: 320    ; ADDRESS OF UNIT 0'S 'INTERRUPT VECTOR'
FKOLVL: 322    ; ADDRESS OF UNIT 0'S 'INTERRUPT LEVEL'

```

```

1328
1329
1330
1331 001344 170340
1332 001346 170342
1333 001350 170344
1334 001352 170346
1335 001354 000324
1336 001356 000326
1337
1338
1339
1340 001360 170360
1341 001362 170362
1342 001364 170364
1343 001366 170366
1344 001370 000350
1345 001372 000352
1346
1347
1348
1349 001374 170370
1350 001376 170372
1351 001400 170374
1352 001402 170376
1353 001404 000354
1354 001406 000356
1355
1356
1357
1358 001410 170300
1359 001412 170302
1360 001414 170304
1361 001416 170306
1362 001420 000330
1363 001422 000332
1364
1365
1366
1367 001424 170310
1368 001426 170312
1369 001430 170314
1370 001432 170316
1371 001434 000340
1372 001436 000342
1373
1374
1375
1376 001440 170320
1377 001442 170322
1378 001444 170324
1379 001446 170326
1380 001450 000360
1381 001452 000362
1382
1383

```

;'FK' UNIT 1'S REGISTER ADDRESSES.

```

FK1LDB: 170340 ;ADDRESS OF UNIT 1'S 'LIGHT DATA B'
FK1LDA: 170342 ;ADDRESS OF UNIT 1'S 'LIGHT DATA A'
FK1CSR: 170344 ;ADDRESS OF UNIT 1'S 'CONTROL & STATUS' REG.
FK1DAT: 170346 ;ADDRESS OF UNIT 1'S 'KEYBOARD DATA' REG.
FK1INT: 324 ;ADDRESS OF UNIT 1'S 'INTERRUPT VECTOR'
FK1LVL: 326 ;ADDRESS OF UNIT 1'S 'INTERRUPT LEVEL'

```

;'FK' UNIT 2'S REGISTER ADDRESSES

```

FK2LDB: 170360
FK2LDA: 170362
FK2CSR: 170364
FK2DAT: 170366
FK2INT: 350
FK2LVL: 352

```

;'FK' UNIT 3'S REGISTER ADDRESSES

```

FK3LDB: 170370
FK3LDA: 170372
FK3CSR: 170374
FK3DAT: 170376
FK3INT: 354
FK3LVL: 356

```

;'VT' UNIT 0'S REGISTER ADDRESSES.

```

VTOCAR: 170300 ;ADDRESS OF UNIT 0'S CURSOR ADDRESS REG.
VTOSAR: 170302 ;ADDRESS OF UNIT 0'S STATING ADDRESS REG.
VTOCSR: 170304 ;ADDRESS OF UNIT 0'S COMMAND & STATUS REG.
VTOMAR: 170306 ;ADDRESS OF UNIT 0'S MAINTENANCE REG.
VTOINT: 330 ;ADDRESS OF UNIT 0'S INTERRUPT VECTOR
VTOLVL: 332 ;ADDRESS OF UNIT 0'S INTERRUPT LEVEL

```

;'VT' UNIT 1'S REGISTER ADDRESSES.

```

VT1CAR: 170310 ;ADDRESS OF UNIT 1'S CURSOR ADDRESS REG.
VT1SAR: 170312 ;ADDRESS OF UNIT 1'S STATING ADDRESS REG.
VT1CSR: 170314 ;ADDRESS OF UNIT 1'S COMMAND & STATUS REG.
VT1MAR: 170316 ;ADDRESS OF UNIT 1'S MAINTENANCE REG.
VT1INT: 340 ;ADDRESS OF UNIT 1'S INTERRUPT VECTOR
VT1LVL: 342 ;ADDRESS OF UNIT 1'S INTERRUPT LEVEL

```

;'VT' UNIT 2'S REGISTER ADDRESSES

```

VT2CAR: 170320
VT2SAR: 170322
VT2CSR: 170324
VT2MAR: 170326
VT2INT: 360
VT2LVL: 362

```

;'VT' UNIT 3'S REGISTER ADDRESSES

1384
 1385 001454 170350
 1386 001456 170352
 1387 001460 170354
 1388 001462 170356
 1389 001464 000370
 1390 001466 000372
 1391
 1392 001470 170330
 1393 001472 170332
 1394 001474 170334
 1395 001476 170336
 1396 001500 000320
 1397 001502 000322
 1398
 1399 001504 170300
 1400 001506 170302
 1401 001510 170304
 1402 001512 170306
 1403 001514 000330
 1404 001516 000332
 1405
 1406
 1407
 1408 001520 175610
 1409 001522 175620
 1410 001524 175630
 1411 001526 175640
 1412 001530 175612
 1413 001532 175622
 1414 001534 175632
 1415 001536 175642
 1416 001540 000300
 1417 001542 000302
 1418 001544 000310
 1419 001546 000312
 1420 001550 000400
 1421 001552 000402
 1422 001554 000410
 1423 001556 000412
 1424
 1425 001560 175614
 1426 001562 175624
 1427 001564 175634
 1428 001566 175644
 1429 001570 175616
 1430 001572 175626
 1431 001574 175636
 1432 001576 175646
 1433 001600 000304
 1434 001602 000306
 1435 001604 000314
 1436 001606 000316
 1437 001610 000404
 1438 001612 000406
 1439 001614 000414

VT3CAR: 170350
 VT3SAR: 170352
 VT3CSR: 170354
 VT3MAR: 170356
 VT3INT: 370
 VT3LVL: 372

FKLDB: 170330
 FK LDA: 170332
 FKCSR: 170334
 FKDATA: 170336
 FKINT: 320
 FKLVL: 322

;TEMPORARY STORAGE OF FK ADDRESSES

VTCAR: 170300
 VTSAR: 170302
 VTCSR: 170304
 VT MAR: 170306
 VTINT: 330
 VTLVL: 332

;TEMPORARY STORAGE OF VT ADDRESSES

;DL11 REGISTER ADDRESSES

RCSR0: 175610
 RCSR1: 175620
 RCSR2: 175630
 RCSR3: 175640
 RBUF0: 175612
 RBUF1: 175622
 RBUF2: 175632
 RBUF3: 175642
 RINT0: 300
 RLVLO: 302
 RINT1: 310
 RLVL1: 312
 RINT2: 400
 RLVL2: 402
 RINT3: 410
 RLVL3: 412

;ADDRESS OF UNIT 0'S DL11 REC. CSR
 ;ADDRESS OF UNIT 1'S DL11 REC. CSR
 ;ADDRESS OF UNIT 2'S DL11 REC. CSR
 ;ADDRESS OF UNIT 3'S DL11 REC. CSR
 ;ADDRESS OF UNIT 0'S DL11 REC. BUFFER
 ;ADDRESS OF UNIT 1'S DL11 REC. BUFFER

;ADDRESS OF UNIT 0'S REC. VECTOR

;ADDRESS OF UNIT 1'S REC. VECTOR

;ADDRESS OF UNIT 0'S TRANS. CSR
 ;ADDRESS OF UNIT 1'S DL11 TRANS. CSR

;ADDRESS OF UNIT 0'S DL11 TRANS. BUFFER
 ;ADDRESS OF UNIT 1'S DL11 TRANS. BUFFER

;ADDRESS OF UNIT 0'S DL11 TRANS. VECTOR

;ADDRESS OF UNIT 1'S DL11 TRANS. VECTOR

```

1440 001616 000416 XLVL3: 416
1441
1442 ;DEFINITIONS OF 'VT FIELD CONTROL' FUNCTIONS
1443
1444 000370 UNLINE=370 ;UNDERLINE PRESET
1445 000364 BOLD=364 ;BOLD PRESET
1446 000361 BLANK=361 ;BLANK PRESET
1447 000362 BLINK=362 ;BLINK PRESET
1448 000200 AUDIO=200 ;AUDIO PRESET
1449 000012 EOL=12 ;END OF LINE
1450 000014 EOP=14 ;END OF PARAGRAPH
1451 000031 EOS=31 ;END OF SCREEN
1452 000360 CLRFLD=360 ;CLR FIELD CONTROL
1453 000212 VISEOL=212 ;VISIBLE END OF LINE
1454 000214 VISEOP=214 ;VISIBLE END OF PARAGRAPH
1455 000231 VISEOS=231 ;VISIBLE END OF SCREEN
1456
1457 001620 000000 FCSET: 000000
1458 001622 000000 FCSET1: 000000
1459
1460 ;DEFINITIONS OF 'MAINTENANCE' FUNCTIONS ( SELECTED VIA THE 'CSR')
1461
1462 000000 CSR=000 ;COMMAND & STATUS REG.
1463 000002 DAR=2 ;DATA ADDRESS REG.
1464 000004 SRIDE=4 ;SHIFT REG. INPUT DATA (EVEN)
1465 000006 SRODE=6 ;SHIFT REG. OUTPUT DATA (EVEN)
1466 000010 CAR=10 ;CURSOR ADDRESS REG.
1467 000012 SAR=12 ;START ADDRESS REG.
1468 000014 SRIDO=14 ;SHIFT REG. INPUT DATA (ODD)
1469 000016 SRODO=16 ;SHIFT REG. OUTPUT DATA (ODD)
1470
1471 ;*****
1472 ;TEST INITIALIZATION ROUTINE. IF THE PROGRAM IS STARTED AT LOCATION
1473 ;'200' THE PROGRAM IS SET UP TO RUN VIA A TELETYPE MONITOR. IF THE PRO-
1474 ;GRAM IS STARTED AT LOCATION '204' THE PROGRAM USES PROGRAM 'HALTS'
1475 ;TO COLLECT NEEDED INFORMATION AND TO REPORT ERRORS.
1476 ;*****
1477
1478 001624 012737 000001 024154 INTTY: MOV #1,TTYSWH ;TTY AVAILABLE, SET SOFTWARE SW.
1479 001632 000402 BR ;
1480 001634 005037 024154 NOTTY: CLR TTYSWH ;CLR TTY AVAILABLE SOFTWARE SWITCH.
1481 001640 012777 000340 177444 MOV #340,@PSW
1482 001646 013706 024132 MOV STACK,SP ;INIT STACK POINTER=1000
1483 001652 000240 NOP
1484 001654 000240 NOP
1485 001656 000240 NOP ;INITIALIZATION SWITCH TEST REMOVED TO ALLOW RE-SELECT
1486 001660 000240 NOP
1487 001662 000240 NOP
1488 001664 012737 001702 000004 1$: MOV #CORSIZ,@#4 ;RESET TIMEOUT
1489 001672 012701 020000 MOV #20000,R1 ;TEST CORE SIZE
1490 001676 005721 TST (R1)+
1491 001700 000776 BR -2
1492 001702 162701 002000 CORSIZ: SUB #2000,R1 ;SAVE LAST 1K FOR 'VT' TEST
1493 001706 010137 024140 MOV R1,MEMSIZ
1494 001712 012737 000006 000004 MOV #6,@#4 ;RESET TIMEOUT TO 'ERTRAP'
1495 001720 012737 000004 000006 MOV #4,@#6

```

```

1496 001726 005737 024154      INITB: TST      TTYSWH      ;TTY AVAILABLE?
1497 001732 001106              BNE      INITC      ;YES, INIT VIA TTY.
1498 001734 000000              HALT                    ;SEE IF FORIEGN CHAR SET EXISTS
1499 001736 000240              NOP                    ;BY EXAMINING THE SWITCH REGISTER
1500 001740 017737 177360 001620  MOV      @SWR,FCSET    ;IF CLEAR THEN NOT FORIEGN SET
1501 001746 000000              HALT                    ;GET # OF TUBES FROM SWITCHES
1502 001750 000240              NOP                    ;
1503 001752 017737 177346 024136  MOV      @SWR, UNITNO
1504 001760 042737 177770 024136  BIC      #177770,UNITNO
1505 001766 005777 177332              TST      @SWR          ;IF BIT 15 IN SWR IS SET THEN
1506 001772 100402              BMI      4$            ;LET USER INPUT DEVICE ADDRESSES
1507 001774 000137 002436              JMP      INITCB        ;DEFAULT DEVICES & VECTORS REQUESTED
1508
1509 002000 013700 024136      4$:  MOV      UNITNO,R0
1510 002004 012705 001330              MOV      #FKOLDB,R5
1511 002010 000000              HALT                    ;HALT FOR FK VECTOR
1512 002012 017737 177306 001500  1$:  MOV      @SWR,FKINT
1513 002020 000240              NOP                    ;
1514 002022 000000              HALT                    ;HALT FOR FK ADDR.
1515 002024 017704 177274              MOV      @SWR,R4
1516 002030 004737 002700              JSR      PC,SETED
1517 002034 005300              DEC      R0
1518 002036 001364              BNE      1$
1519
1520 002040 012705 001410              MOV      #VTOCAR,R5
1521 002044 013700 024136              MOV      UNITNO,R0
1522 002050 000000              HALT                    ;HALT FOR VT VECTOR
1523 002052 017737 177246 001500  2$:  MOV      @SWR,FKINT
1524 002060 000240              NOP                    ;
1525 002062 000000              HALT                    ;HALT FOR VT ADDR.
1526 002064 017704 177234              MOV      @SWR,R4
1527 002070 004737 002700              JSR      PC,SETED
1528 002074 005300              DEC      R0
1529 002076 001364              BNE      2$
1530 002100 013700 024136              MOV      UNITNO,R0
1531 002104 012702 001540              MOV      #RINTO,R2
1532 002110 012705 001600              MOV      #XINTO,R5
1533 002114 000000              HALT                    ;HALT FOR DL11 VECTOR
1534 002116 017703 177202      3$:  MOV      @SWR,R3
1535 002122 012322              MOV      (3)+,(2)+
1536 002124 012322              MOV      (3)+,(2)+
1537 002126 012325              MOV      (3)+,(5)+
1538 002130 012325              MOV      (3)+,(5)+
1539 002132 005300              DEC      R0
1540 002134 001367              BNE      3$
1541 002136 000000      INITBA: HALT
1542 002140 017737 177160 024134  MOV      @SWR,UNITFG  ;GET TUBE NO. FROM SWITCHES
1543 002146 000543              BR                    ;SAVE TUBE NO.
1544 002150 005737 024166      INITC: TST      MTRSWH
1545 002154 001002              BNE      .+6           ;TYPE HEADER
1546 002156 104012              PRINT                    ;NO, ITS BEEN TYPED
1547 002160 021027              TITLE
1548 002162 104012              PRINT                    ;PRINT PROGRAM HEADER
1549 002164 021440              MFCP
1550 002166 104015              ASEMBL
1551 002170 042703 177776              BIC      #177776,R3
;
```

1552	002174	010337	001620		MOV	R3,FCSET	
1553	002200	104012			PRINT		;REQUEST NO. OF TUBES
1554	002202	024002			MED1		
1555	002204	104015			ASEMBL		;GET INPUT
1556	002206	010337	024136		MOV	R3,UNITNO	
1557	002212	104012	021160		PRINT,	MES1A	;TEXT 'ARE DEFAULT ADDR. & VECT. OK?'
1558	002216	104015			ASEMBL		;WAIT FOR HIS ANSWER
1559	002220	005703			TST	R3	;WAS 'CR' TYPED?
1560	002222	001505			BEQ	INITCB	;YES, USE DEFAULT ADDRESSES
1561	002224	022703	000001		CMP	#1,R3	;WAS 'Y' TYPED?
1562	002230	001502			BEQ	INITCB	;YES, IT SURE WAS
1563	002232	012705	001330	INITAC:	MOV	#FKOLDB,R5	;NO, LET USER DEFINE HIS OWN ADDRESSES
1564	002236	013700	024136	1\$:	MOV	UNITNO,R0	;WE HAVE ASK HIM FOR ADDRESSES
1565	002242	112737	000260	014750	MOVB	#260,MRUNN1	
1566	002250	104012		2\$:	PRINT		;FIRST GET FK ADDRESS
1567	002252	024033			MFKP		
1568	002254	104012			PRINT		;PRINT UNIT NO
1569	002256	014750			MRUNN1		
1570	002260	104015			ASEMBL		
1571	002262	010337	001500		MOV	R3,FKINT	
1572	002266	001770			BEQ	2\$	
1573	002270	004737	002700		JSR	PC,SETED	
1574	002274	105237	014750		INCB	MRUNN1	
1575	002300	005300			DEC	R0	
1576	002302	001362			BNE	2\$	
1577	002304	012705	001410		MOV	#VTOCAR,R5	;GET VT VECTORS AND ADDRESS
1578	002310	013700	024136		MOV	UNITNO,R0	;NUMBER OF TUBES
1579	002314	112737	000260	014750	MOVB	#260,MRUNN1	
1580							
1581	002322	104012		3\$:	PRINT		
1582	002324	024026			MUTP		
1583	002326	104012			PRINT		
1584	002330	024037			MAUFK		
1585	002332	104012			PRINT		
1586	002334	014750			MRUNN1		
1587	002336	104015			ASEMBL		
1588	002340	010337	001500		MOV	R3,FKINT	
1589	002344	001766			BEQ	3\$	
1590	002346	004737	002700		JSR	PC,SETED	
1591	002352	105237	014750		INCB	MRUNN1	
1592	002356	005300			DEC	R0	
1593	002360	001360			BNE	3\$	
1594							
1595	002362	013700	024136		MOV	UNITNO,R0	;ASK FOR DL11 VECTORS
1596	002366	012702	001540		MOV	#RINTO,R2	
1597	002372	012705	001600		MOV	#XINTO,R5	
1598	002376	112737	000260	014750	MOVB	#260,MRUNN1	
1599							
1600	002404	104012		4\$:	PRINT		;PRINT MESSAGE
1601	002406	024075			MDLVA		
1602	002410	104012			PRINT		
1603	002412	014750			MRUNN1		
1604	002414	104015			ASEMBL		
1605	002416	012322			MOV	(3)+,(2)+	
1606	002420	012322			MOV	(3)+,(2)+	
1607	002422	012325			MOV	(3)+,(5)+	

```

1608 002424 012325          MOV      (3)+,(5)+
1609 002426 105237 014750  INCB     MRUNN1
1610 002432 005300          DEC     RO
1611 002434 001363          BNE     4$
1612
1613 002436 005737 024154  INITCB: TST     TTYSWH
1614 002442 001635          BEQ     INITBA
1615 002444 104012          PRINT
1616 002446 021376          MESS
1617 002450 104015          ASEMBL ;REQUEST UNIT NO.
1618 002452 010337 024134  MOV     R3,UNITFG ;DECODE IT.
1619 002456 013737 024134 024126 INITD: MOV     UNITFG,PIMP ;SAVE IT
1620 002464 005037 024164  CLR     SWAPEM ;CLR SWAP SW.
1621 002470 022737 000004 024134  CMP     #4,UNITFG
1622 002476 001004          BNE     .+12
1623 002500 005037 024134  CLR     UNITFG ;YES, SET UP UNIT '0'
1624 002504 005237 024164  INC     SWAPEM ;SET SWAP SW.
1625 002510 004737 016320  JSR     PC,STUNIT ;SET UP SELECTED UNIT ADDRESSES
1626 002514 005737 024154  TST     TTYSWH ;TTY AVAILABLE?
1627 002520 001402          BEQ     MONITR ;NO
1628 002522 104012          PRINT ;YES
1629 002524 021263          MESS3 ;TEXT 'TYPE IN TST NO. TO BE EXECUTED'.
1630 ;*****
1631 ;TEST SELECTION ROUTINE. IF NO TTY IS AVAILABLE THE PROGRAM WILL HALT
1632 ;AND WAIT FOR A TEST NO. TO BE LOADED VIA THE SW.'S. IF A TTY IS
1633 ;AVAILABLE, A REQUEST IS MADE VIA THE PRINTER FOR THE SELECTED TEST
1634 ;NUMBER TO BE TYPED IN.
1635 ;*****
1636
1637 002526 000005  MONITR: RESET ;INITIALIZE ON ENTRY
1638 002530 104022          NULL ;NULL FOR PRINTER.
1639 002532 013706 024132  MOV     STACK,SP ;RESET STACK POINTER
1640 002536 005037 024164  CLR     SWAPEM
1641 002542 022737 000004 024126  CMP     #4,PIMP ;DOES HE WANT TO SWAP?
1642 002550 001006          BNE     1$
1643 002552 005037 024134  CLR     UNITFG
1644 002556 005237 024164  INC     SWAPEM
1645 002562 004737 016320  JSR     PC,STUNIT
1646 002566 012737 001726 024130 1$: MOV     #INITB,AVECTR ;SET UP '1A' RESTART ADDR.
1647 002574 004737 016710  JSR     PC,CLRFKV ;CLR 'FK' INTR ADDR TO HALT
1648 002600 004737 016774  JSR     PC,CLRVTV ;CLR 'VT' INTR ADDR TO HALT
1649 002604 012737 000001 024166  MOV     #1,MTRSWH ;SET MONITOR SW.
1650 002612 005037 024150  CLR     RUNSWH ;CLR 'RUN ALL' SOFTWARE SW.
1651 002616 005037 024240  CLR     HOLDSO
1652 002622 005037 024162  CLR     DONESW
1653 002626 005737 024154  TST     TTYSWH ;TTY AVAILABLE?
1654 002632 001004          BNE     TTYMTR ;YES, REQUEST TEST NO.
1655 002634 000000          HALT ;GET TEST NO. FROM DATA SW'S
1656 002636 017703 176462  MOV     @SWR,R3 ;SAVE TEST NO.
1657 002642 000403          BR     TTYMT1 ;EXECUTE SELECTED TEST
1658 002644 104012  TTYMTR: PRINT
1659 002646 022774          CNTRLC
1660 002650 104015          ASEMBL ;DECODE IT
1661 002652 042703 177740  TTYMT1: BIC     #177740,R3 ;CLR UN-WANTED BITS
1662 002656 006303          ASL     R3
1663 002660 020327 000044  CMP     R3,#44

```

```

1664 002664 003002          BGT      .+6          ;TYPE '?' IF NUMBER TO BIG
1665 002666 000173 002750  JMP      @TEST00(R3) ;EXECUTE SELECTED TEST
1666 002672 104012          PRINT
1667 002674 023007          QMARK          ;TYPE '?' ON ILLEGAL NO.'S
1668 002676 000714          BR      MONITR+2    ;RE-START
1669 002700 010146          SETED: MOV     R1,-(6) ;SAVE R1
1670 002702 012701 000004  MOV     #4,R1
1671 002706 010425          IS:   MOV     R4,(5)+ ;MOV ADDR TO ADDR. POINTER
1672 002710 062704 000002  ADD     #2,R4
1673 002714 005301          DEC     R1
1674 002716 001373          BNE     IS
1675 002720 013725 001500  MOV     FKINT,(5)+ ;MOV VECTOR ADDR TO ADDR POINTER
1676 002724 062737 000002 001500  ADD     #2,FKINT
1677 002732 013725 001500  MOV     FKINT,(5)+
1678 002736 012601          MOV     (6)+,R1
1679 002740 062737 000002 001500  ADD     #2,FKINT
1680 002746 000207          RTS      PC
1681
1682          ;*****
1683          ;'VT20' TEST TABLE
1684          ;*****
1685 002750 003122          TEST00: VTLOGIC          ;VT 'LOGIC' TEST
1686
1687 002752 005166          TEST01: VTCHAR          ;VT 'CHARACTER DISPLAY' TEST
1688
1689 002754 005226          TEST02: VTMOD1          ;VT 'FIELD MODE' TEST '1'
1690
1691 002756 005322          TEST03: VTMOD2          ;VT 'CSR PRESET' TEST
1692
1693 002760 005454          TEST04: VTEOL          ;VT 'END OF LINE' TEST
1694
1695 002762 005574          TEST05: VTEOS          ;VT 'END OF SCREEN' TEST
1696
1697 002764 005660          TEST06: VTBLANK          ;VT 'BLANK CONTROL' TEST
1698
1699 002766 005740          TEST07: VTALINE          ;VT 'ALIGNMENT' TEST
1700
1701 002770 005766          TEST10: VTFOCUS          ;VT 'FOCUS' TEST
1702
1703 002772 006014          TEST11: VTWORST          ;VT 'WORST CASE' TEST
1704
1705 002774 006042          TEST12: VTCURSR          ;VT 'CURSOR CONTROL' TEST
1706
1707 002776 006136          TEST13: VTODD          ;VT 'ODD ADDRESS' TEST.
1708
1709 003000 003016          TEST14: VTRUN          ;VT 'RUN ALL' TEST
1710
1711 003002 006236          TEST15: FKLOGIC          ;FK 'LOGIC' TEST
1712
1713 003004 007416          TEST16: FKFUN          ;FK 'FUNCTION' TEST
1714
1715 003006 007576          TEST17: FKCHAR          ;FK 'CHARACTER' TEST
1716
1717 003010 010126          TEST20: FKREPT          ;FK 'REPEATIBILITY' TEST
1718
1719 003012 010344          TEST21: SYSTST          ;VT 'SYSTEM EXERCISER' TEST

```



```

1720
1721 003014 014162
1722
1723
1724
1725
1726
1727
1728
1729
1730 003016 005237 024150
1731 003022 000137 003122
1732
1733
1734
1735
1736 003026 013704 001510
1737 003032 013705 001512
1738 003036 005037 024444
1739 003042 005037 024446
1740 003046 012737 000100 020402
1741 003054 012737 017532 000034
1742 003062 012737 003142 020406
1743 003070 012777 000340 176214
1744 003076 012737 000001 024142
1745 003104 005037 024432
1746 003110 005037 024434
1747 003114 005037 020404
1748 003120 000207
1749
1750 003122 005737 024154
1751 003126 001403
1752 003130 104012
1753 003132 023013
1754 003134 104022
1755 003136 004737 003026
1756
1757
1758
1759
1760
1761 003142 000240
1762 003144 000240
1763 003146 012714 125212
1764 003152 000005
1765 003154 011401
1766 003156 042701 050000
1767 003162 001401
1768 003164 104400
1769
1770
1771
1772
1773
1774 003166 104001 000002
1775 003172 000005

```

```

TEST22: TESTX ;FK/VT 'LOGIC SUBTEST' SELECTION
;*****
;SUBROUTINE ENTERED VIA SELECTING 'VTRUN'. THIS ROUTINE SETS A
;SOFTWARE SW. WHICH ENABLES THE VT LOGIC TEST, ALL DISPLAY TESTS, AND
;THE ASCII KEYBOARD TEST TO BE RUN IN ORDER. THESE TESTS WILL BE RUN
;ON BOTH UNITS IF UNIT '2' WAS SELECTED.
;*****
VTRUN: INC RUNSWH ;SET SOFTWARE SW.
JMP VTLOGIC ;START 1ST TEST.
;*****
;VT20 LOGIC TEST (00)
;*****
SETUPV: MOV VTCSR,VTCSR4 ;LOAD 'R4' WITH THE 'CSR' ADDRESS
MOV VTMAR,VTMARS ;LOAD 'R5' WITH THE 'MAINT' ADDR. REG.
CLR MESPRT ;CLR PRINT INHIBIT SW.
CLR MESPT1 ;CLR PRINT INHIBIT SW.
MOV #100,ICOUNT
MOV #VTERR,@#34 ;LOAD THE TRAP 'TRAP' WITH LOGIC ERROR TRAP
MOV #VT1,RETURN ;SET UP 'SCOPE' RETURN ADDRESS
MOV #340,@PSW ;SET PROCESSOR PRIORITY TO '7'
MOV #1,TSTNUM ;SET UP TEST NO.
CLR SHFTSW
CLR SHFPRT
CLR SCOPEF
RTS PC

VTLOGIC: TST TTYSWH ;TTY AVAILABLE?
BEQ .+10 ;NO
PRINT MES20 ;TEXT 'VT20 LOGIC TEST'
NULL
VTREST: JSR PC,SETUPV ;INITIALIZE THE LOGIC TEST
;*****
;TEST THAT THE 'CSR' WAS INITIALIZED CORRECTLY
;*****
VT1: NOP ;'VT' TEST 1
NOP
MOV #125212,@VTCSR4
RESET
MOV @VTCSR4,R1 ;READ CONTENTS OF THE 'CSR'
BIC #50000,R1 ;IGNORE 'BIT 12' AND 'BIT 14'
BEQ .+4 ;RESET SHOULD HAVE CLEARED 'CSR'
ERROR ;'CSR' SHOULD = '0 OR 50000'
;*****
;TEST THAT THE 'MAINTENANCE' REGISTER WAS INITIALIZED TO '0'
;*****
VT2: SCOPE,2 ;'VT' TEST 2
RESET

```

```

1776 003174 011501
1777 003176 042701 050000
1778 003202 001401
1779 003204 104400
1780
1781
1782
1783
1784 003206 104001 000003
1785 003212 012714 125212
1786 003216 011401
1787 003220 042701 050000
1788 003224 022701 125212
1789 003230 001401
1790 003232 104400
1791 003234 005014
1792
1793
1794
1795
1796
1797 003236 104001 000004
1798 003242 012714 052504
1799 003246 104021
1800 003250 022714 052504
1801 003254 001401
1802 003256 104400
1803 003260 005014
1804
1805
1806
1807
1808
1809 003262 104001 000005
1810 003266 012714 157760
1811 003272 104021
1812 003274 022715 157760
1813 003300 001401
1814 003302 104400
1815 003304 005014
1816 003306 011501
1817 003310 042701 050000
1818 003314 005701
1819 003316 001401
1820 003320 104400
1821
1822
1823
1824
1825
1826 003322 104001 000006
1827 003326 012777 125252 176150
1828 003334 012777 000010 176146
1829 003342 022715 125252
1830 003346 001401
1831 003350 104400

```

```

MOV @VTMARS,R1 ;READ CONTENTS OF THE 'MAR'
BIC #50000,R1 ;IGNORE 'BIT 12' AND 'BIT 14'
BEQ .+4
ERROR ;RESET DIDN'T CLR THE 'MAINT' REG. (EXCEPT BIT '12' AND
:*****
:TEST FOR WRITING THE 'CSR' TO 125212
:*****
VT3: SCOPE,3 ;'VT' TEST 3
MOV #125212,@VTCSR4 ;WRITE 'CSR' TO '125212'
MOV @VTCSR4,R1 ;READ CONTENTS OF THE 'CSR'
BIC #50000,R1 ;IGNORE 'BIT 12' AND 'BIT 14'
CMP #125212,R1
BEQ .+4
ERROR ;'CSR' SHOULD = '125212'
CLR @VTCSR4
:*****
:TEST FOR WRITING THE 'CSR' TO '52504'
:*****
VT4: SCOPE,4 ;'VT' TEST 4
MOV #52504,@VTCSR4 ;WRITE 'CSR' TO '52504'
WAITSYNC
CMP #52504,@VTCSR4
BEQ .+4 ;**BEWARE** BIT 6 IS CLEARED ON TYPEOUT
ERROR ;'CSR' SHOULD = '52504'
CLR @VTCSR4 ;WRITE 'CSR' TO '0'
:*****
:TEST FOR READING THE 'CSR' VIA THE 'MAINT' REG.
:*****
VT5: SCOPE,5 ;'VT' TEST 5
MOV #157760,@VTCSR4 ;SELECT 'MAINT CSR' VIA THE 'CSR'
WAITSYNC
CMP #157760,@VTMARS ;MAINT SHOULD='CSR'
BEQ .+4 ;**BEWARE** BIT 6 IS CLEARED ON TYPEOUT
ERROR ;MAINT REG. NOT ='CSR' (157760)
CLR @VTCSR4 ;CLR 'CSR'
MOV @VTMARS,R1
BIC #50000,R1
TST R1 ;'MAR' SHOULD = 'CSR' (0)
BEQ .+4
ERROR ;MAINT REG. NOT='CSR' (0)
:*****
:WRITE THE 'CURSOR' ADDRESS REGISTER TO '125252'
:*****
VT6: SCOPE,6 ;'VT' TEST 6
MOV #125252,@VTCAR ;WRITE REG. TO ALTERNATE 1'S
MOV #CAR,@VTCSR ;SELECT 'CURSOR' ADDRESS VIA 'CSR'
CMP #125252,@VTMARS ;READ CURSOR ADDRESS VIA 'MAINT'
BEQ .+4
ERROR ;CURSOR ADDRESS SHOULD=125252

```

```

1832 003352 005077 176132
1833 003356 005077 176122
1834
1835
1836
1837
1838
1839 003362 104001 000007
1840 003366 012777 052525 176110
1841 003374 012777 000010 176106
1842 003402 022715 052525
1843 003406 001401
1844 003410 104400
1845 003412 005077 176072
1846 003416 005077 176062
1847
1848
1849
1850
1851
1852 003422 104001 000010
1853 003426 012777 177777 176050
1854 003434 012714 000010
1855 003440 005077 176040
1856 003444 005715
1857 003446 001401
1858 003450 104400
1859
1860
1861
1862
1863
1864 003452 104001 000011
1865 003456 012777 177777 176020
1866 003464 000005
1867 003466 012714 000010
1868 003472 005715
1869 003474 001401
1870 003476 104400
1871
1872
1873
1874
1875 003500 104001 000012
1876 003504 012703 000001
1877 003510 010377 175770
1878 003514 012714 000010
1879 003520 020315
1880 003522 001401
1881 003524 104400
1882 003526 005014
1883 003530 005077 175750
1884 003534 006103
1885 003536 001364
1886
1887

```

```

CLR @VTCAR
CLR @VTCAR
;*****
;WRITE THE 'CURSOR' ADDRESS REGISTER TO '52525'
;*****
VT7: SCOPE,7 ;'VT' TEST 7
MOV #52525,@VTCAR ;WRITE COMPLIMENT OF PART I
MOV #CAR,@VTCAR ;SELECT 'CURSOR' ADDRESS VIA 'CSR'
CMP #52525,@VTMARS ;READ 'CAR' VIA 'MAR'
BEQ .+4
ERROR ;'CAR' SHOULD=52525
CLR @VTCAR
CLR @VTCAR
;*****
;WRITE THE 'CURSOR' ADDRESS REGISTER TO '0'
;*****
VT10: SCOPE,10 ;'VT' TEST 10
MOV #-1,@VTCAR ;WRITE REG. TO -1
MOV #CAR,@VTCAR4 ;SELECT CURSOR ADDRESS
CLR @VTCAR ;WRITE REG. TO '0'
TST @VTMARS
BEQ .+4
ERROR ;CURSOR ADDRESS SHOULD='0'
;*****
;TEST THAT THE 'CURSOR ADDRESS' REG. IS CLEARED VIA 'RESET'
;*****
VT11: SCOPE,11 ;'VT' TEST 11
MOV #-1,@VTCAR ;WRITE REG. =-1
RESET ;ISSUE 'RESET' TO CLR REG.
MOV #CAR,@VTCAR4 ;SELECT 'CAR' VIA 'CSR'
TST @VTMARS ;TEST IF 'CAR' WAS CLEARED
BEQ .+4
ERROR ;RESET DIDN'T CLR 'CAR' REG.
;*****
;TEST FOR ROTATING A '1' THRU THE 'CAR'.
;*****
VT12: SCOPE,12 ;'VT' TEST 12
MOV #1,R3 ;LOAD THE TEST REG. WITH '1'
LOOP12: MOV R3,@VTCAR ;LOAD THE 'CAR' WITH THE TEST REG.
MOV #CAR,@VTCAR4
CMP R3,@VTMARS
BEQ .+4
ERROR ;'CAR' SHOULD = CONTAINS OF R3
CLR @VTCAR4
CLR @VTCAR
ROL R3 ;ROTATE BIT
BNE LOOP12
;*****
;TEST FOR WRITING THE 'STARTING ADDRESS' REG. TO '125252'

```

```

1888
1889
1890 003540 104001 000013
1891 003544 012777 024474 175734
1892 003552 012714 000012
1893 003556 012777 125252 175722
1894 003564 104021
1895 003566 022715 125252
1896 003572 001401
1897 003574 104400
1898
1899
1900
1901
1902
1903 003576 104001 000014
1904 003602 012777 024474 175675
1905 003610 012714 000012
1906 003614 012777 052525 175664
1907 003622 104021
1908 003624 022715 052525
1909 003630 001401
1910 003632 104400
1911
1912
1913
1914
1915 003634 104001 000015
1916 003640 012777 024474 175640
1917 003646 012714 000012
1918 003652 005077 175630
1919 003656 104021
1920 003660 005715
1921 003662 001401
1922 003664 104400
1923
1924
1925
1926
1927
1928
1929 003666 104001 000016
1930 003672 012777 024474 175606
1931 003700 012703 000001
1932 003704 012714 000012
1933 003710 010377 175572
1934 003714 104021
1935 003716 020315
1936 003720 001401
1937 003722 104400
1938 003724 006103
1939 003726 001366
1940
1941
1942
1943

```

```

;*****
VT13: SCOPE,13 ;'VT' TEST 13
MOV #VTBUFF,AVTSAR ;LOAD LEGAL ADDR. TO PREVENT TIMEOUT
MOV #SAR,AVTCSR4 ;SELECT 'SAR' TO MAINT VIA 'CSR'
MOV #125252,AVTSAR ;WRITE REG. TO ALTERNATE 1'S
WAITSYNC ;WAIT FOR VERTICAL SYNC.
CMP #125252,AVTMARS
BEQ .+4
ERROR ;STARTING ADDR. REG. SHOULD=125252

;*****
;TEST FOR WRITING THE 'STARTING ADDRESS' REG. TO '52525'
;*****
VT14: SCOPE,14 ;'VT' TEST 14
MOV #VTBUFF,AVTSAR ;LOAD LEGAL ADDR. TO PREVENT TIMEOUT
MOV #SAR,AVTCSR4 ;SELECT 'SAR' TO MAINT VIA 'CSR'
MOV #52525,AVTSAR ;WRITE COMPLIMENT OF PART I
WAITSYNC ;WAIT FOR VERTICAL SYNC.
CMP #52525,AVTMARS
BEQ .+4
ERROR ;'SAR' SHOULD =52525

;*****
;TEST FOR WRITING THE 'STARTING ADDRESS' REG. TO '0'
;*****
VT15: SCOPE,15 ;'VT' TEST 15
MOV #VTBUFF,AVTSAR ;LOAD LEGAL ADDR. TO PREVENT TIMEOUT
MOV #SAR,AVTCSR4 ;SELECT 'STARTING ADDR.' REG.
CLR AVTSAR ;WRITE REG. TO '0'
WAITSYNC ;WAIT FOR VERTICAL SYNC.
TST AVTMARS ;CHECK FOR '0'
BEQ .+4
ERROR ;STARTING ADDRESS REG. SHOULD='0'

;*****
;TEST FOR ROTATING A '1' THUR THE STARTING ADDRESS REG.
;NOTE: 'R3' IS USED AS A CHECKING ADDRESS AND WILL CONTAIN THE
;EXPECTED DATA ON A FAILURE.
;*****
VT16: SCOPE,16 ;'VT' TEST 16
MOV #VTBUFF,AVTSAR ;LOAD LEGAL ADDR. TO PREVENT TIMEOUT
MOV #1,R3 ;LOAD THE TEST REG. WITH '1'
LOOP16: MOV #SAR,AVTCSR4 ;SELECT 'STARTING ADDR.' REG.
MOV R3,AVTSAR ;LOAD 'SAR' FROM 'R3'
WAITSYNC ;WAIT FOR VERTICAL SYNC.
CMP R3,AVTMARS ;CHECK IF REG. LOADED CORRECTLY.
BEQ .+4
ERROR ;'SAR SHOULD = CONTAINS OF 'R3'
ROL R3 ;ROTATE BIT
BNE LOOP16 ;LOOP UNTIL CARRY IS SET

;*****
;TEST THAT 'RESET' WILL CLR THE 'STARTING ADDRESS' REG.
;*****

```

```

1944 003730 104001 000017      VT17: SCOPE,17      ;'VT' TEST 17
1945 003734 012777 024474 175544      MOV #VTBUFF,@VTSAR ;LOAD LEGAL ADDR. TO PREVENT TIMEOUT
1946 003742 012714 000012      MOV #SAR,@VTCR4 ;SELECT 'STARTING ADDRESS' REG.
1947 003746 012777 177777 175532      MOV #-1,@VTSAR ;WRITE REG. TO '-1'
1948 003754 000005      RESET ;RESET SHOULD CLR REG.
1949 003756 104021      WAITSYNC ;WAIT FOR VERTICAL SYNC.
1950 003760 012714 000012      MOV #SAR,@VTCR4 ;RE-SELECT THE 'SAR'
1951 003764 005715      TST @VTMARS
1952 003766 001401      BEQ .+4
1953 003770 104400      ERROR ;RESET DIDN'T CLR 'SAR'
1954
1955 ;*****
1956 ;TEST THAT THE 'DAR' (DATA ADDRESS REG.) IS LOADED FROM 'SAR' (START ADDR. REG.)
1957 ;*****
1958
1959 003772 104001 000020      VT20: SCOPE,20      ;'VT' TEST 20
1960 003776 012777 125252 175502      MOV #125252,@VTSAR ;LOAD THE 'SAR'
1961 004004 104021      WAITSYNC ;WAIT FOR VERTICAL SYNC.
1962 004006 012714 000002      MOV #DAR,@VTCR4 ;SELECT MAINT. 'DAR' VIA 'CSR'
1963 004012 022715 125252      CMP #125252,@VTMARS
1964 004016 001401      BEQ .+4
1965 004020 104400      ERROR ;'DAR' WASN'T LOADED FROM 'SAR'
1966 004022 005014      CLR @VTCR4
1967
1968 ;*****
1969 ;ISSUE RESET TO CLR TIME-OUT FROM PREVIOUS TEST
1970 ;*****
1971
1972 004024 104001 000021      VT21: SCOPE,21      ;'VT' TEST 21
1973 004030 000005      RESET
1974
1975 ;*****
1976 ;TEST THAT THE 'DAR' (DATA ADDRESS REG.) IS LOADED FROM 'SAR' (START ADDR. REG.)
1977 ;*****
1978
1979 004032 104001 000022      VT22: SCOPE,22      ;'VT' TEST 22
1980 004036 012777 052525 175442      MOV #052525,@VTSAR ;LOAD THE 'SAR'
1981 004044 104021      WAITSYNC ;WAIT FOR VERTICAL SYNC.
1982 004046 012714 000002      MOV #DAR,@VTCR4 ;SELECT MAINT. 'DAR'
1983 004052 022715 052524      CMP #052524,@VTMARS
1984 004056 001401      BEQ .+4
1985 004060 104400      ERROR ;'DAR' WASN'T LOADED FROM 'SAR'
1986 004062 005014      CLR @VTCR4
1987
1988 ;*****
1989 ;ISSUE RESET TO CLR TIME-OUT FROM PREVIOUS TEST
1990 ;*****
1991 004064 104001 000023      VT23: SCOPE,23      ;'VT' TEST 23
1992 004070 000005      RESET
1993
1994 ;*****
1995 ;TEST FOR ROTATING A '1' THUR THE 'DAR' (DAR IS LOADED FROM THE SAR).
1996 ;*****
1997 004072 104001 000024      VT24: SCOPE,24      ;'VT' TEST 24
1998 004076 012703 000002      MOV #2,R3 ;LOAD THE TEST REG. WITH '2'
1999 004102 010377 175400      LOOP24: MOV R3,@VTSAR ;LOAD THE 'SAR'

```

```

2000 004106 104021           WAITSYNC           ;WAIT FOR VERTICAL SYNC.
2001 004110 012714 000002   MOV #DAR,@VTCSR4   ;SELECT MAINT. 'DAR' VIA 'CSR'
2002 004114 020315           CMP R3,@VTMARS     ;TEST 'DAR'
2003 004116 001401           BEQ .+4
2004 004120 104400           ERROR             ;'DAR' WASN'T LOADED FROM 'SAR'
2005 004122 005014           CLR @VTCSR4
2006 004124 006103           ROL R3             ;ROTATE BIT
2007 004126 001365           BNE LOOP24        ;ROTATE THRU REG.
2008
2009
2010
2011
2012
2013 004130 104001 000025   VT25:  SCOPE,25    ;'VT' TEST 25
2014 004134 000005           RESET
2015
2016
2017
2018
2019
2020
2021 004136 104001 000026   VT26:  SCOPE,26    ;'VT' TEST 26
2022 004142 012777 020727 175336   MOV #SHFTBF+1,@VTSAR ;LOAD 'SAR'
2023 004150 104021           WAITSYNC           ;WAIT FOR VERTICAL SYNC.
2024 004152 012777 020727 175324   MOV #SHFTBF+1,@VTCAR ;LOAD 'CAR' = 'SAR'
2025 004160 012714 000004           MOV #SRIDE,@VTCSR4 ;FORCE 'NPR' TRANSFER
2026 004164 032714 010000           BIT #10000,@VTCSR4 ;TEST IF BIT 12 (CAR BIT) SET
2027 004170 001001           BNE .+4
2028 004172 104400           ERROR             ;'CAR' BIT IN 'CSR' DIDN'T SET W/'CAR' = 'SAR'
2029 004174 032714 040000           BIT #40000,@VTCSR4 ;TEST IF BIT14 CLEAR
2030 004200 001401           BEQ .+4           ;SHOULD BE WHEN SAR=CAR
2031 004202 104400           ERROR             ;BIT 14 NOT CLEAR
2032 004204 005014           CLR @VTCSR4
2033
2034
2035
2036
2037
2038 004206 104001 000027   VT27:  SCOPE,27    ;'VT' TEST 27
2039 004212 012777 020741 175266   MOV #SHFTBF+13,@VTSAR ;LOAD 'SAR'
2040 004220 104021           WAITSYNC           ;WAIT FOR VERTICAL SYNC.
2041 004222 012777 020737 175254   MOV #SHFTBF+11,@VTCAR ;LOAD 'CAR' = 'SAR'
2042 004230 012714 000004           MOV #SRIDE,@VTCSR4 ;FORCE 'NPR' TRANSFER
2043 004234 032715 010000           BIT #10000,@VTMARS ;TEST IF BIT 12 (CAR BIT) SET
2044 004240 001401           BEQ .+4
2045 004242 104400           ERROR             ;'CAR' SET W/ 'CAR' NOT = 'SAR'
2046 004244 032777 004000 175236   BIT #4000,@VTCAR   ;SEE IF BIT 14 IN CSR, SET SHOULD BE
2047 004252 001001           BNE .+4           ;WHEN CAR <> SAR WITH ECO FOR BIT 14
2048 004254 000240           NOP              ;NOP IF NO ECO, PUT IN 104400 IN THIS
2049
2050
2051
2052
2053
2054
2055 004260 104001 000030   VT30:  SCOPE,30    ;'VT' TEST 30

```

2056 004264 012777 021024 175214
 2057 004272 104021
 2058 004274 012714 000004
 2059 004300 032714 020000
 2060 004304 001001
 2061 004306 104400
 2062 004310 005014
 2063
 2064
 2065
 2066
 2067
 2068
 2069
 2070 004312 104001 000031
 2071 004316 000005
 2072 004320 012737 000010 020402
 2073 004326 012737 023544 020026
 2074 004334 012737 000004 024430
 2075 004342 004737 004752
 2076 004346 004737 005004
 2077 004352 104400
 2078 004354 000415
 2079
 2080
 2081
 2082
 2083
 2084
 2085 004356 012737 023555 020026
 2086 004364 012737 000006 024430
 2087 004372 004737 004752
 2088 004376 005722
 2089 004400 005200
 2090 004402 004737 005004
 2091 004406 104400
 2092 004410 000240
 2093
 2094
 2095
 2096
 2097
 2098
 2099 004412 104001 000032
 2100 004416 000005
 2101 004420 012737 023544 020026
 2102 004426 012737 000014 024430
 2103 004434 004737 004752
 2104 004440 004737 005004
 2105 004444 104400
 2106 004446 000415
 2107
 2108
 2109
 2110
 2111

```

MOV #SHFEND,AVTSAR ;LOAD 'SAR'
WAITSYNC ;WAIT FOR VERTICAL SYNC.
MOV #SRIDE,AVTCSR4 ;FORCE 'NPR' TRANSFER
BIT #20000,AVTCSR4 ;TEST IF BIT 13 (EOS)
BNE .+4 ;'EOS' DIDN'T SET IN 'CSR'
ERROR
CLR AVTCSR4

:*****
:TEST THE 'SHIFT REG. INPUT DATA EVEN' GETS TRANSFERRED FROM THE DATA BUFFER
:TO THE 'MAR'. A SERIES OF '64' SHIFTS ARE DONE AND THE 'MAR' IS COMPARED
:TO THE DATA BUFFER AFTER EACH SHIFT.
:*****

VT31: SCOPE,31 ;'VT' TEST 31
RESET
MOV #10,ICOUNT ;RESET ITERATION COUNT TO '8'
MOV #MES37,SHFMES ;SET UP TO PRINT ON ERROR
MOV #SRIDE,SELECT ;SELECT 'SHIFT INPUT DATA EVEN'
JSR PC,INITSF ;INITIALIZE SHIFT TEST
JSR PC,SHIFT ;SUBROUTINE TO FORCE '64' 'NPR' XFER
ERROR ;'SHIFT INPUT DATA EVEN' SHIFT ERROR
BR TAGVB ;EXIT ON ERROR

:*****
:AT THIS POINT '64' SHIFTS HAVE BEEN MADE TO THE 'SHIFT REG. INPUT DATA
:EVENT'. THE NEXT TEST EXECUTES '63' MORE SHIFTS VALIDATING THE
:'SHIFT REG. OUTPUT DATA EVEN' INPUTTED FROM THE PREVIOUS TEST.
:*****

MOV #MES38,SHFMES
MOV #SRODE,SELECT ;SELECT 'SHIFT OUTPUT DATA EVEN'
JSR PC,INITSF ;INITIALIZE SHIFT TEST
TST (R2)+ ;UPDATE BUFFER COMPARE POINTER BY '2'
INC R0 ;SHIFT ONLY 63 TIMES FOR OUTPUT
JSR PC,SHIFT ;SUBROUTINE TO FORCE THE '63' NPR XFERS
ERROR ;'SHIFT OUTPUT DATA EVEN' SHIFT ERROR
TAGVB: NOP

:*****
:TEST THE 'SHIFT REG. INPUT DATA ODD' GETS TRANSFERRED FROM THE DATA BUFFER
:TO THE 'MAR'. A SERIES OF '64' SHIFTS ARE DONE AND THE 'MAR' IS COMPARED
:TO THE DATA BUFFER AFTER EACH SHIFT.
:*****

VT32: SCOPE,32 ;'VT' TEST 32
RESET
MOV #MES37,SHFMES
MOV #SRIDO,SELECT ;SELECT 'SHIFT INPUT DATA ODD'
JSR PC,INITSF ;INITIALIZE SHIFT TEST
JSR PC,SHIFT ;SUBROUTINE TO FORCE '1' 'NPR' XFER
ERROR ;'SHIFT INPUT DATA ODD' SHIFT ERROR
BR TAGVC ;EXIT ON ERROR

:*****
:AT THIS POINT '64' SHIFTS HAVE BEEN MADE TO THE 'SHIFT REG. INPUT DATA
:ODD'. THIS NEXT TEST EXECUTES '63' MORE SHIFTS VALIDATING THE
:'SHIFT REG. OUTPUT DATA ODD' INPUTTED FROM THE PREVIOUS TEST.

```

```

2112
2113
2114 004450 012737 023555 020026
2115 004456 012737 000016 024430
2116 004464 004737 004752
2117 004470 005722
2118 004472 005200
2119 004474 004737 005004
2120 004500 104400
2121 004502 000240
2122
2123
2124
2125
2126
2127 004504 104001 000033
2128 004510 012737 000100 020402
2129 004516 005037 024432
2130 004522 012777 173000 174756
2131 004530 004737 016740
2132 004534 004544
2133 004536 104021
2134 004540 104021
2135 004542 104400
2136 004544 013706 024132
2137 004550 004737 016774
2138 004554 005714
2139 004556 100401
2140 004560 104400
2141 004562 000005
2142
2143
2144
2145
2146
2147 004564 104001 000034
2148 004570 012777 173000 174710
2149 004576 004737 016740
2150 004602 004624
2151 004604 012777 000240 174500
2152 004612 012700 170000
2153 004616 005200
2154 004620 001376
2155 004622 000403
2156 004624 104400
2157 004626 013706 024132
2158 004632 000005
2159 004634 004737 016774
2160
2161
2162
2163
2164
2165 004640 104001 000035
2166 004644 012701 000010
2167 004650 012700 020730

```

```

MOV #MES38, SHFMES
MOV #SRODO, SELECT ;SELECT 'SHIFT OUTPUT DATA ODD'
JSR PC, INITSF ;INITIALIZE SHIFT TEST
TST (R2)+ ;UPDATE BUFFER COMPARE POINTER BY '2'
INC RO ;SHIFT ONLY 63 TIMES FOR OUTPUT
JSR PC, SHIFT ;SUBROUTINE TO FORCE THE '64' NPR XFERS
ERROR ;'SHIFT OUTPUT DATA ODD' SHIFT ERROR

```

TAGVC: NOP

```

*****
;TEST 'NXM' FOR CAUSING AN INTERRUPT VIA LOADING A 'NXM' ADDRESS IN THE 'SAR'
;NOTE: THIS IS THE FIRST TIME THAT THE 'INTR. ENABLE BIT IS SET OR USED.
*****

```

```

VT33: SCOPE, 33 ;'VT' TEST 33
MOV #100, ICOUNT ;RESET ITERATION COUNT TO '100'
CLR SHFTSW ;CLR SOFTWARE SW.
MOV #173000, @VTSAR ;LOAD 'SAR' W/ 'NXM' ADDRESS
JSR PC, LDVTVT ;LOAD THE 'VT' VECTOR ADDRESS
TAGVA
WAITSYNC ;WAIT FOR INTERRUPT
WAITSYNC
ERROR ;SETTING 'SAR' TO 'NXM' ADDR. DIDN'T ERROR
TAGVA: MOV STACK, SP ;RESET STACK POINTER
JSR PC, CLRVTV ;RESET 'VT' VECTOR ADDR. TO HALT
TST @VTCRA4 ;TEST THE 'NXM' SET THE ERROR BIT
BMI .+4 ;'NXM' FAILED TO SET 'ERROR' BIT
ERROR
RESET

```

```

*****
;TEST THAT NO INTERRUPT OCCURS WITH PROC. AT PRIORITY '5'
*****

```

```

VT34: SCOPE, 34 ;'VT' TEST 34
MOV #173000, @VTSAR
JSR PC, LDVTVT ;LOAD 'VT' INTERRUPT VECTOR
TAGVD
MOV #240, @PSW
MOV #170000, RO
INC RO
BNE .-2
BR .+10 ;'OK', NO INTERRUPT OCCURRED
TAGVD: ERROR ;INTERRUPTED WITH PROC @ PRIOR '5'
MOV STACK, SP
RESET
JSR PC, CLRVTV

```

```

*****
;TEST THAT NO 'NXM' ERRORS OCCUR IF THE 'VT' IS RUNNING AND THE
;STARTING ADDRESS REG. IS CHANGED.
*****

```

```

VT35: SCOPE, 35 ;'VT' TEST 35
MOV #10, R1 ;SET UP A LOOP COUNTER
MOV #SHFTBF+2, RO

```



```

2168 004654 012777 020726 174624      MOV      #SHFTBF, @VTSAR
2169 004662 104021                      WAITSYNC ;WAIT FOR VERTICAL SYNC
2170 004664 004737 016740      JSR      PC, LDVTVT ;LOAD 'VT' INTERRUPT VECTOR ADDR.
2171 004670 004716                      TAGVF
2172 004672 005002      TAGVE: CLR      R2
2173 004674 005202                      INC      R2 ;EVERYTIME 'R2' OVERFLOWS THE
2174 004676 001376                      BNE     .-2 ;'SAR' IS RELOADED
2175 004700 010077 174602      MOV      R0, @VTSAR ;RELOAD 'SAR'
2176 004704 005720                      TST     (R0)+
2177 004706 104021                      WAITSYNC ;WAIT FOR VERTICAL SYNC
2178 004710 005301                      DEC      R1
2179 004712 001367                      BNE     TAGVE
2180 004714 000403                      BR      .+10 ;OK, NO INTERRUPTS OCCURRED
2181 004716 104400      TAGVF: ERROR ;'NXM' ERROR OCCURRED WHEN 'SAR' WAS CHANGED
2182 004720 013706 024132      MOV      STACK, SP
2183 004724 004737 016774      JSR      PC, CLRVTV
2184
2185 ;*****
2186 ;'VT' LOGIC TEST COMPLETE
2187 ;*****
2188
2189 004730 104001 000036      VT36:  SCOPE, 36 ;'VT' TEST 36
2190 004734 104012                      PRINT
2191 004736 021610                      MES7 ;TEXT 'TEST COMPLETE'
2192 004740 104006                      ENDTST ;END OF TEST
2193 004742 000401                      BR      .+4
2194 004744 000510                      BR      VTCHAR ;CONTINUE TO CHARACTER TEST
2195 004746 000137 003136      JMP      VTREST ;RESTART THE LOGIC TEST.
2196
2197 ;*****
2198 ;SUBROUTINE TO INITIALIZE THE 'SAR', 'CAR' AND SHIFT COUNTER FOR THE
2199 ;'VT' LOGIC SHIFT TEST.
2200 ;*****
2201
2202 004752 012777 020726 174526      INITSF: MOV      #SHFTBF, @VTSAR ;LOAD 'STARTING ADDR'
2203 004760 012703 020726      MOV      #SHFTBF, R3
2204 004764 104021                      WAITSYNC ;WAIT FOR VERTICAL SYNC.
2205 004766 012702 020726      MOV      #SHFTBF, R2
2206 004772 005000                      CLR      R0 ;CLR 'SHIFT' COUNTER
2207 004774 012737 000001 024432      MOV      #1, SHFTSW ;SET SOFTWARE SW.
2208 005002 000207                      RTS      PC
2209
2210 ;*****
2211 ;SUBROUTINE TO FORCE THE 'NPR' XFER FOR THE 'VT' LOGIC SHIFT TEST.
2212 ;THE ADDRESS 'SELECT' CONTAINS THE SELECTED SHIFT FUNCTION (SHIFT INPUT
2213 ;DATA EVEN, ETC.) WHICH IS EXECUTED.
2214 ;*****
2215 005004 005200      SHIFT: INC      R0 ;INCREMENT SOFTWARE SHIFT COUNTER
2216 005006 013714 024430      MOV      SELECT, @VTCSR4 ;FORCE 'NPR' TRANSFER
2217 005012 005723                      TST     (R3)+ ;UPDATE STARTING ADDRESS BY '2'
2218 005014 010377 174466      MOV      R3, @VTSAR ;SET UP TO GET NEXT BUFFER WORD.
2219 005020 104021                      WAITSYNC ;WAIT FOR VERTICAL SYNC.
2220 005022 012201                      MOV     (R2)+, R1 ;GET XFER'D WORD FROM DATA BUFFER
2221 005024 042701 000360      BIC     #360, R1 ;CLR UNWANTED BITS FROM CONTROL WORD
2222 005030 000301                      SWAB   R1 ;SWAP BYTES TO COMPARE TO 'MAR'
2223 005032 022700 000040      SHFLST: CMP     #32., R0 ;IS THIS THE '32' SHIFT?

```

```

2224 005036 001010          BNE      SHF64      ;NO, CHECK FOR '64' SHIFT
2225 005040 022715 020031  CMP      #20031,AVTMARS ;YES, END OF SCREEN SHOULD BE SET IN 'MAR'
2226 005044 001013          BNE      SHFBAD     ;NO, 32ND SHIFT IS BAD
2227 005046 004737 004752  SHF32: JSR      PC,INITSF ;RE-INITIALIZE SOFTWARE POINTERS FOR '32' MORE SHIFTS
2228 005052 012700 000040  MOV      #32,RO      ;RESET SHIFT COUNTER TO COUNT TO '64'
2229 005056 000752          BR       SHIFT      ;DO IT.
2230 005060 022700 000100  SHF64: CMP      #64,RO      ;IS THIS THE LAST SHIFT
2231 005064 001006          BNE      TSTSHF     ;NO, COMPARE DATA TO 'MAR' DIRECTLY
2232 005066 022715 020031  CMP      #20031,AVTMARS ;YES, 'EOS' SHOULD BE SET
2233 005072 001432          BEQ      EXITSF     ;EXIT
2234 005074 052701 020000  SHFBAD: BIS      #20000,R1 ;SET UP 'R1' TO = EXPECTED WORD
2235 005100 000412          BR       REPORT     ;REPORT ERROR
2236 005102 032701 000400  TSTSHF: BIT      #400,R1 ;COMPLIMENT BLANK BIT FOR COMPARE
2237 005106 001003          BNE      .+10
2238 005110 052701 000400  BIS      #400,R1
2239 005114 000402          BR       .+6
2240 005116 042701 000400  BIC      #400,R1
2241 005122 020115          CMP      R1,AVTMARS ;DOES 'MAR' = EXPT'D WORD?
2242 005124 001727          BEQ      SHIFT     ;YES, FORCE NEXT SHIFT
2243 005126 022700 000100  REPORT: CMP     #64,RO ;IS ERROR ON THE LAST SHIFT?
2244 005132 001404          BEQ      .+12      ;YES,EXIT
2245 005134 032777 010000 174162 BIT      #SW12,ASWR ;NO, IS SW SET TO CONTINUE SHIFTING?
2246 005142 001401          BEQ      .+4       ;YES
2247 005144 000207          RTS      PC        ;RETURN TO SUBTEST AND REPORT ERROR
2248 005146 104400          ERROR
2249 005150 022700 000040  CMP      #32,RO      ;IS THIS THE '32' SHIFT
2250 005154 001734          BEQ      SHF32     ;YES, RE-INITIALIZE BUFFER
2251 005156 000712          BR       SHIFT     ;NO,CONTINUE SHIFTING
2252 005160 062716 000004  EXITSF: ADD     #4,(SP) ;SET UP STACK TO SKIP OVER ERROR
2253 005164 000207          RTS      PC
2254
2255 ;*****
2256 ;VT 'CHARACTER DISPLAY' TEST (1)
2257 ;THIS TEST DISPLAYS ALL '160' VT CHARACTERS ONE AT A TIME UNTIL THE ENTIRE
2258 ;'VT' SCREEN IS FILLED. EACH CHARACTER IS DISPLAYED FOR APPROXIMATELY '3' SECONDS
2259 ;BEFORE THE NEXT CHARACTER IS DISPLAYED. ON EACH LINE,THE '64' CHARACTER IS DIS-
2260 ;PLAYED AS EITHER A 'VISIBLE END OF LINE OR A 'VISIBLE END OF PARAGRAPH'.
2261 ;DATA 'SW14' MAY BE SET AT ANYTIME TO HOLD ON ANY INDIVIDUAL CHARACTER.
2262 ;*****
2263 005166 012737 005166 020406 VTCHAR: MOV      #VTCHAR,RETURN
2264 005174 104000          DISPLAY
2265 005176 023204          MES24              ;TEXT "VT CHARACTER DISPLAY TEST".
2266 005200 104005          EOSBUF            ;LOAD 'VT' BUFFER WITH 'EOS'.
2267 005202 005037 024152  CLR      INTSWH     ;SUBROUTINE SOFTWARE SW.
2268 005206 000401          BR       .+4
2269 005210 104004          DELAY              ;3 SEC. DELAY.
2270 005212 004737 015170  JSR      PC,INTCHR  ;CREATE & DISPLAY NEXT CHAR.
2271 005216 000240          NOP                ;NORMAL RETURN TO HERE
2272 005220 000773          BR       .-10      ;END OF LINE RETURNS HERE
2273 005222 104006          ENDTST
2274 005224 000760          BR       VTCHAR    ;END OF SCREEN RETURN HERE
2275
2276 ;*****
2277 ;VT 'FIELD CONTROL' TEST (2)
2278 ;THIS TEST DISPLAYS THE 'VT' CHARACTER SET USING ALL '5' FIELD CONTROL MODES.
2279 ;(NORMAL,BLINK,BOLD,BLANK AND UNDERLINE). THE TEST DISPLAYS CHARACTERS

```

```

2280 ;'40-136' ON LINES '1-5' USING ALL '5' MODES. CHARACTERS '137-240 (WITH THE
2281 ;EXCEPTION OF CHARACTERS 212,214 & 231) ON LINES '6-10'. CHARACTERS
2282 ;'241-277' ARE DISPLAYED ON THE LAST '5' LINES.
2283 ;*****
2284
2285 005226 012737 005226 020406 VTMOD1: MOV #VTMOD1,RETURN
2286 005234 104000 DISPLAY
2287 005236 023240 MES25 ;TEXT 'VT FIELD CONTROL TEST'.
2288 005240 104005 EOSBUF ;LOAD 'VT' BUFFER WITH 'VISIBLE EOS'.
2289 005242 012737 024474 024460 MOV #VTBUFF,KSTOR1
2290 005250 005037 024152 CLR INTSWH ;CLR SOFTWARE SWITCH
2291 005254 012737 177775 024464 MOV #-3,KSTOR3 ;SET UP FOR '3' SETS OF CHAR.'S
2292 005262 004737 005274 JSR PC,VTMD1A ;LOAD CHAR.'S
2293 005266 104023 THEND
2294 005270 000756 BR VTMOD1
2295 005272 000413 BR VTMOD2 ;GO TO NEXT TEST IF SW. IS SET
2296
2297 005274 004737 015170 VTMD1A: JSR PC,INTCHR ;CREATE AND LOAD CHAR.
2298 005300 000775 BR .-4 ;LOOP UNTIL LINE IS FULL
2299 005302 000240 NOP ;END OF LINE RETURNS HERE
2300 005304 104007 SETLNE ;SET UP '4' LINES USING ALL MODES
2301 005306 010037 024460 MOV RO,KSTOR1 ;SAVE ST. ADDR. FOR NEXT BUFFERED LINE
2302 005312 005237 024464 INC KSTOR3 ;FINISHED?
2303 005316 001366 BNE VTMD1A ;NO, CREATE NEXT SET
2304 005320 000207 RTS PC ;YES, EXIT
2305 ;*****
2306 ;VT 'CSR PRESET' TEST (3)
2307 ;THIS TEST DISPLAYS THE 'VT' CHARACTER SET IN ALL FIELD CONTROL MODES
2308 ; (BOLD,BLINK,BLANK & UNDERLINE). THE FIELD CONTROL IS SELECTED VIA THE
2309 ; 'CSR' BEFORE EACH CHARACTER SET IS DISPLAYED.
2310 ;*****
2311
2312 005322 012737 005322 020406 VTMOD2: MOV #VTMOD2,RETURN
2313 005330 104000 DISPLAY
2314 005332 023270 MES26 ;TEXT 'VT' CSR PRESET TEST'.
2315 005334 104005 EOSBUF ;LOAD 'VT' BUFFER W/ 'EOS'.
2316 005336 104000 DISPLAY
2317 005340 023566 MES39 ;TEXT 'BOLD PRESET TEST'
2318 005342 012777 002001 174140 VTMD2A: MOV #2001,@VTCSR ;SELECT 'BOLD' PRESET & GO
2319 005350 004737 005434 JSR PC,VTMD2B ;LOAD CHAR.'S
2320 005354 104000 DISPLAY
2321 005356 023611 MES40 ;TEXT 'BLINK PRESET TEST'
2322 005360 012777 001001 174122 MOV #1001,@VTCSR ;SELECT 'BLINK' PRESET
2323 005366 004737 005434 JSR PC,VTMD2B ;LOAD CHAR.'S
2324 005372 104000 DISPLAY
2325 005374 023635 MES41 ;TEXT 'BLANK PRESET TEST'
2326 005376 012777 000401 174104 MOV #401,@VTCSR ;SELECT 'BLANK' PRESET
2327 005404 004737 005434 JSR PC,VTMD2B ;LOAD CHAR.'S
2328 005410 104000 DISPLAY
2329 005412 023661 MES42 ;TEXT 'UNDERLINE PRESET TEST'
2330 005414 012777 004001 174066 MOV #4001,@VTCSR ;SELECT 'UNDERLINE' PRESET
2331 005422 004737 005434 JSR PC,VTMD2B ;LOAD CHAR.'S
2332 005426 104006 ENDTST
2333 005430 000734 BR VTMOD2 ;RE-CYCLE
2334 005432 000410 BR VTEOL ;BRANCH TO NEXT TEST IS SW. SET
2335

```

2336 005434 005037 024152
 2337 005440 004737 015170
 2338 005444 000240
 2339 005446 000774
 2340 005450 104020
 2341 005452 000207

VTMD2B: CLR INTSWH ;CLR SOFTWARE SW.
 JSR PC,INTCHR ;CREATE & DISPLAY CHAR.
 NOP
 BR -6 ;LOOP UNTIL SCREEN IF FILLED
 DELAYL ;6 SEC. DELAY
 RTS PC ;EXIT

2342 ;*****
 2343 ;VT 'END OF LINE' TEST (4)
 2344 ;THIS TEST TESTS THAT THE 'END OF LINE' CHARACTER CAN BE PLACED IN
 2345 ;ANY OF THE '1024' CHARACTER POSITIONS ON THE SCREEN. THIS IS DONE BY
 2346 ;STARTING ALL 16 LINES WITH A 'VISIBLE END OF LINE' CHARACTER AND
 2347 ;THEN INCREMENTING THESE CHARACTERS ACROSS THE SCREEN.
 2348 ;*****
 2349

2350 005454 012737 005454 020406
 2351 005462 104000
 2352 005464 023315
 2353 005466 104005
 2354 005470 005001
 2355 005472 112720 000212
 2356 005476 012703 000077
 2357 005502 105020
 2358 005504 005303
 2359 005506 001375
 2360 005510 005201
 2361 005512 022701 000021
 2362 005516 001365
 2363 005520 104004
 2364
 2365
 2366
 2367
 2368
 2369

VTEOL: MOV #VTEOL, RETURN
 DISPLAY
 MES27 ;TEXT 'VT END OF LINE TEST'
 EOSBUF ;LOAD 'VT' BUFFER & START
 CLR R1 ;CLR LINE CNTR.
 RESCNT: MOV #VISEOL, (R0)+ ;TERMINATE LINE W/ 'EOL'
 MOV #63, R3 ;SAVE '63' BYTES BEFORE NEXT LINE
 CLRB (R0)+
 DEC R3
 BNE -4
 INC R1 ;INCREMENT LINE CNTR.
 CMP #17, R1 ;DONE '16' LINES ?
 BNE RESCNT ;NO, CREATE NEXT LINE
 DELAY ;DISPLAY THE 1ST 16 LINES OF 'EOL'

;AT THIS POINT 16 LINES EACH CONTAINING 1 'EOL' AND 63 SPACES HAVE
 ;BEEN SET UP IN MEMORY. THE FOLLOWING SUBROUTINE GOES THRU ONE
 ;BY ONE REPLACING THE 'EOL' WITH AN '*' AND MOVING THE 'EOL'
 ;INTO THE NEXT SPACE UNTIL THE ENTIRE LINE HAS BEEN CHECKED.

2370 005522 012705 000077
 2371 005526 012704 000020
 2372 005532 012700 024474
 2373 005536 122027 000212
 2374 005542 001375
 2375 005544 112740 000052
 2376 005550 105720
 2377 005552 112720 000212
 2378 005556 005304
 2379 005560 001366
 2380 005562 104004
 2381 005564 005305
 2382 005566 001357
 2383 005570 104006
 2384 005572 000730
 2385
 2386
 2387
 2388
 2389
 2390
 2391

SRHEOL: MOV #63, R5 ;SET UP FOR '63' SHIFTS
 MOV #16, R4
 MOV #VTBUFF, R0 ;RESET BUFFER ADDR. POINTER
 SHFEOL: CMPB (R0)+, #VISEOL ;SEARCH BUFFER FOR "EOL"
 BNE -4
 MOV #52, -(R0) ;REPLACE 'EOL' W/ '*'
 TSTB (R0)+ ;RESET POINTER
 MOV #VISEOL, (R0)+ ;MOVE 'EOL' OVER '1'
 DEC R4 ;DONE '16' LINES
 BNE SHFEOL ;NO, SHIFT NEXT 'EOL'
 DELAY ;3 SEC. DISPLAY DELAY
 DEC R5 ;DONE '63' SHIFTS
 BNE SRHEOL ;NO, LOOP AGAIN
 ENDTST
 BR VTEOL ;YES, RESTART TEST

2385 ;*****
 2386 ;VT 'END OF SCREEN' TEST (5)
 2387 ;THIS TEST TESTS THAT THE 'VISIBLE END OF SCREEN' CHARACTER CAN
 2388 ;BE PLACED IN ALL '1024' CHARACTER POSITIONS ON THE SCREEN. THIS
 2389 ;IS DONE BY FIRST PRE-LOADING THE BUFFER TO BE DISPLAYED WITH
 2390 ;QUESTION MARKS. THE 'EOS' CHARACTER IS THEN INCREMENTED THRU THIS
 2391 ;BUFFER REPLACING THE QUESTION MARKS BEHIND THE 'EOS' CHARACTER WITH

```

2392 ;DOTS.
2393 ;*****
2394
2395 005574 012737 005574 020406 VTEOS: MOV #VTEOS, RETURN
2396 005602 104000 DISPLAY
2397 005604 023341 MES30 ;TEXT 'VT END OF SCREEN TEST'
2398 005606 104011 PRELOAD ;PRE-LOAD THE BUFFER WITH '?'
2399 005610 077 .BYTE 77
2400 005611 077 .BYTE 77
2401 005612 012701 002000 MOV #1024, R1 ;SET UP CHAR. CNTR.
2402 005616 012700 024474 MOV #VTBUFF, R0 ;SET UP BUFFER POINTER
2403 005622 112710 000231 MOV #VISEOS, (R0) ;LOAD 1ST CHAR. WITH 'EOS'
2404 005626 012777 000001 173654 MOV #1, @VTCSR ;ST DISPLAY
2405 005634 104004 SRHEOS: DELAY
2406 005636 005301 DEC R1 ;DONE
2407 005640 001405 BEQ ENDEOS ;YES, EXIT
2408 005642 012720 000056 MOV #56, (R0)+ ;NO, REPLACE THE 'EOS' W/ DOT.
2409 005646 112710 000231 MOV #VISEOS, (R0) ;MOVE 'EOS' OVER '1' PLACE
2410 005652 000770 BR SRHEOS
2411 005654 104006 ENDEOS: ENDTST
2412 005656 000746 BR VTEOS ;YES, RESTART TEST
2413
2414 ;*****
2415 ;VT 'BLANK CONTROL' TEST (6)
2416 ;THIS TEST INCREMENTS THE 'BLANK' CONTROL CHARACTER (361) THRU A
2417 ;FULL SCREEN BUFFER OF '?'S. THE SCREEN STARTS BLANK AND END'S
2418 ;UP FULL OF *'S.
2419 ;*****
2420
2421 005660 012737 005660 020406 VTBLANK: MOV #VTBLANK, RETURN
2422 005666 104000 DISPLAY
2423 005670 023371 MES31 ;TEXT 'VT BLANKING' TEST
2424 005672 104011 PRELOAD ;PRELOAD DATA BUFFER
2425 005674 077 .BYTE 77 ;W/ '?'S
2426 005675 077 .BYTE 77
2427 005676 112710 000361 MOV #BLANK, (R0) ;SET UP BLANK CONTROL CHAR.
2428 005702 012701 002000 MOV #1024, R1 ;SET UP CNTR
2429 005706 012777 000001 173574 MOV #1, @VTCSR ;ST. DISPLAY
2430 005714 104004 DELAY ;3 SEC. DISPLAY DELAY
2431 005716 112720 000052 MOV #52, (R0)+ ;REPLACE 'BLANK' CHAR. W/*
2432 005722 112710 000361 MOV #BLANK, (R0) ;MOVE 'BLANK' CHAR. UP '1'
2433 005726 104004 DELAY
2434 005730 005301 DEC R1 ;MOVED 'BLANK' THUR BUFFER
2435 005732 001371 BNE MOVBLK ;NO
2436 005734 104006 ENDTST
2437 005736 000750 BR VTBLANK ;RESTART TEST
2438
2439 ;*****
2440 ;VT 'ALIGNMENT' TEST (7)
2441 ;THIS TEST DISPLAYS A FULL SCREEN OF THE CHAR. 'E' TO ENABLE A VISUAL
2442 ;ALIGNMENT OF THE 'VT'.
2443 ;*****
2444
2445 005740 012737 005740 020406 VTALIN: MOV #VTALIN, RETURN
2446 005746 104000 DISPLAY
2447 005750 023421 MES33 ;TEXT 'VT' 'ALIGNMENT' TEST

```

```

2448 005752 104011          PRELOAD          ;PRELOAD 'VT' DATA BUFFER
2449 005754      105          .BYTE 105          ;W/ E'S
2450 005755      105          .BYTE 105
2451 005756 005277 173526    INC      @VTCSR    ;ST. DISPLAY
2452 005762 104023          THEND
2453
2454 005764 000765          BR      VTALIN
2455 ;*****
2456 ;VT 'FOCUS' TEST (10)
2457 ;THIS TEST DISPLAYS A FULL SCREEN OF ALTERNATE *'S & 1'S TO ENABLE A VISUAL
2458 ;FOCUS OF THE 'VT'.
2459 ;*****
2460
2461 005766 012737 005766 020406 VTFOCUS:MOV      #VTFOCUS,RETURN
2462 005774 104000          DISPLAY
2463 005776 023445          MES34          ;TEXT 'VT FOCUS TEST'
2464 006000 104011          PRELOAD          ;LOAD 'VT' DATA BUFFER
2465 006002      052          .BYTE 52          ;WITH *'S
2466 006003      061          .BYTE 61          ;AND 1'S
2467 006004 005277 173500    INC      @VTCSR    ;ST. DISPLAY
2468 006010 104023          THEND
2469 006012 000765          BR      VTFOCUS
2470
2471 ;*****
2472 ;VT 'WORST CASE' TEST (11)
2473 ;THIS TEST DISPLAYS A FULL SCREEN OF THE WORST CASE PATTERN '125 & 252'.
2474 ;*****
2475
2476 006014 012737 006014 020406 VTWORST:MOV      #VTWORST,RETURN
2477 006022 104000          DISPLAY
2478 006024 023465          MES35          ;TEXT 'VT WORST CASE TEST'
2479 006026 104011          PRELOAD          ;LOAD 'VT' BUFFER W/ WORST CASE
2480 006030      125          .BYTE 125         ;CHAR. 125
2481 006031      252          .BYTE 252         ;CHAR. 252
2482 006032 005277 173452    INC      @VTCSR    ;ST. DISPLAY
2483 006036 104023          THEND
2484 006040 000765          BR      VTWORST
2485 ;*****
2486 ;VT 'CURSOR MOVEMENT' TEST (12)
2487 ;THIS TEST TESTS THAT THE CURSOR CONTROL CHARACTER CAN BE PLACED
2488 ;IN ANY POSITION ON THE SCREEN. THIS IS DONE BY DISPLAYING A 'CURSOR EOS'
2489 ;BEING INCREMENTED ACROSS THE SCREEN. AS THE CURSOR IS INCREMENTED ALONG
2490 ;ASTRICKS ARE FILLED IN BEHIND IT AND WHEN THE TEST IS COMPLETE THE
2491 ;ENTIRE SCREEN WILL BE FILLED WITH *'S.
2492 ;*****
2493
2494 006042 012737 006042 020406 VTCURSR:MOV      #VTCURSR,RETURN
2495 006050 104000          DISPLAY
2496 006052 023512          MES36          ;TEXT 'VT CURSOR MOVEMENT TEST'
2497 006054 104011          PRELOAD          ;PRE-LOAD THE BUFFER WITH '?'
2498 006056      077          .BYTE 77
2499 006057      077          .BYTE 77
2500 006060 012701 002000    MOV      #1024,R1  ;SET UP CHAR. CNTR.
2501 006064 012700 024474    MOV      #VTBUFF,RO ;SET UP BUFFER POINTER
2502 006070 010077 173410    MOV      RO,@VTCAR ;LOAD 'CURSOR' ADDRESS REG.
2503 006074 012710 000231    MOV      #VISEOS,(RO) ;LOAD 1ST CHAR. WITH 'EOS'

```

2504 006100 012777 000001 173402
 2505 006106 104004
 2506 006110 005301
 2507 006112 001407
 2508 006114 012720 000056
 2509 006120 010077 173360
 2510 006124 012710 000231
 2511 006130 000766
 2512 006132 104006
 2513 006134 000742
 2514
 2515
 2516
 2517
 2518
 2519
 2520
 2521 006136 012737 006136 020406
 2522 006144 104000
 2523 006146 022077
 2524 006150 005077 173334
 2525 006154 012777 022251 173324
 2526 006162 005277 173322
 2527 006166 104023
 2528 006170 000762
 2529 006172 000421
 2530
 2531
 2532
 2533
 2534
 2535
 2536 006174 013704 001474
 2537 006200 005037 024444
 2538 006204 012737 000400 020402
 2539 006212 012737 017446 000034
 2540 006220 012737 006246 020406
 2541 006226 012737 000001 024142
 2542 006234 000207
 2543
 2544 006236 104000
 2545 006240 021240
 2546 006242 004737 006174
 2547
 2548
 2549
 2550
 2551
 2552 006246 000240
 2553 006250 000240
 2554 006252 000005
 2555 006254 005714
 2556 006256 001401
 2557 006260 104400
 2558
 2559

```

SHFCUR: MOV #1, @VTCSR ;ST DISPLAY
          DELAY
          DEC R1 ;DONE
          BEQ ENDCUR ;YES, EXIT
          MOV #56, (R0)+ ;NO, REPLACE THE 'EOS' W/ DOT.
          MOV RO, @VTCAR ;UPDATE 'CAR' TO STAY WITH 'EOS'
          MOV #VISEOS, (R0) ;MOVE 'EOS' OVER 'I' PLACE
          BR SHFCUR
ENDCUR: ENDTST
          BR VTCURSR ;YES, RESTART TEST

;*****
;VT 'ODD ADDRESS' TEST (13)
;THIS TEST SIMPLY DISPLAYS A MESSAGE WHICH IS LOADED FROM AN ODD
;STARTING ADDRESS.
;*****
VTODD: MOV #VTODD, RETURN
        DISPLAY
        MES10 ;TEXT 'VT ODD ADDRESS TEST'
        CLR @VTCSR ;STOP DISPLAY
        MOV #MES11, @VTSAR ;LOAD 'SAR' W/ MESSAGE FROM ODD ADDRESS.
        INC @VTCSR ;START & DISPLAY MESSAGE
        THEND
        BR VTODD
        BR FKLOGIC ;CONTINUE TO FK LOGIC TEST

;*****
;FK11 FUNCTION KEYBOARD LOGIC TEST (15)
;THIS IS AN OPERATOR INTERVENTION TEST WHICH TESTS THE FUNCTION KEYBOARD LOGIC.
;REQUESTS ARE MADE FROM THE PROGRAM FOR INPUTS FROM THE FUNCTION KEYBOARD.
;*****
SETUPF: MOV FKCSR, FKCSR4 ;MOVE 'CSR' ADDR. TO R4
          CLR MESPRT ;CLR PRINT INHIBIT SW.
          MOV #400, ICOUNT ;INITIALIZE THE ITERATION COUNT
          MOV #FKERR, @#34 ;LOAD 'TRAP' VECTOR FOR ERROR HANDLER
          MOV #FKT1, RETURN ;SET UP THE RETURN ADDRESS FOR SCOPE
          MOV #1, TSTNUM ;INIT TEST NO. CNTR.
          RTS PC

FKLOGIC: DISPLAY
          MES2 ;TEXT 'FK LOGIC TEST;
          JSR PC, SETUPF ;INITIALIZE THE 'FK' LOGIC TEST

;*****
;ISSUE 'RESET' AND CHECK THAT THE 'CONTROL REG. IS INITIALIZED CORRECTLY
;*****
FKT1: NOP ;FK TEST '1'
       NOP
       RESET
       TST @FKCSR4 ;'RESET' SHOULD OF CLEARED REGISTER
       BEQ FKT2 ;BRANCH IF CLEARED
       ERROR ;FK 'CSR' WASN'T CLEARED VIA 'RESET'

;*****

```

```

2560
2561
2562
2563 006262 104001 000002
2564 006266 052714 177665
2565 006272 011400
2566 006274 042700 002000
2567 006300 022700 100201
2568 006304 001401
2569 006306 104400
2570 006310 012714 000012
2571
2572
2573
2574
2575
2576 006314 104001 000003
2577 006320 052714 177665
2578 006324 005014
2579 006326 032714 000001
2580 006332 001401
2581 006334 104400
2582 006336 012714 000012
2583
2584
2585
2586
2587 006342 104001 000004
2588 006346 052714 177665
2589 006352 000005
2590 006354 005714
2591 006356 001401
2592 006360 104400
2593
2594
2595
2596
2597
2598 006362 104001 000005
2599 006366 012700 176000
2600 006372 005200
2601 006374 001376
2602 006376 032714 002000
2603 006402 001401
2604 006404 104400
2605 006406 012714 000012
2606
2607
2608
2609
2610
2611 006412 104001 000006
2612 006416 012700 176000
2613 006422 052714 000001
2614 006426 005200
2615 006430 001376

```

```

;TEST THAT THE 'ERROR', 'ASCII FLAG', & 'GO' BITS CAN BE WROTE TO '1'
;*****
FKT2:  SCOPE,2 ;'FK' TEST 2
      BIS #177665, @FKCSR4 ;WRITE ALL 'CSR' BITS EXCEPT '6' TO 1'S
      MOV @FKCSR4, R0
      BIC #2000, R0 ;CLR INTERVAL TIMING BIT
      CMP #100201, R0 ;TEST IF 'BITS '15,7,0' SET
      BEQ .+4
      ERROR ;BITS '15,7&0' FAILED TO WRITE TO 1 IN 'CSR'
      MOV #12, @FKCSR4 ;CLR FLAGS

```

```

;*****
;TEST THAT THE 'GO' BIT CAN BE WROTE TO '0'
;*****

```

```

FKT3:  SCOPE,3 ;'FK' TEST 3
      BIS #177665, @FKCSR4 ;WRITE ALL 'CSR' BITS TO '1'
      CLR @FKCSR4 ;WRITE BITS TO '0'
      BIT #1, @FKCSR4 ;TEST IF 'GO' BIT CLEARED
      BEQ .+4
      ERROR ;THE 'GO' BIT DIDN'T CLR
      MOV #12, @FKCSR4 ;CLR FLAGS

```

```

;*****
;TEST THAT THE 'ERROR', 'ASCII FLAG' & 'GO' BITS CAN BE CLEARED VIA 'RESET'
;*****

```

```

FKT4:  SCOPE,4 ;'FK' TEST 4
      BIS #177665, @FKCSR4 ;WRITE BITS TO 1
      RESET
      TST @FKCSR4 ;TEST IF 'RESET' CLEARED 'CSR'
      BEQ .+4
      ERROR ;RESET FAILED TO CLR 'CSR'

```

```

;*****
;TEST THAT THE 'INTERVAL TIMER' FLAG DOESN'T SET IF 'GO' IS CLEARED
;*****

```

```

FKT5:  SCOPE,5 ;'FK' TEST 5
      MOV #-2000, R0 ;SET UP WAIT LOOP
      INC R0 ;WAIT FOR INTERVAL TIMER FLAG
      BNE .-2
      BIT #2000, @FKCSR4
      BEQ .+4
      ERROR ;INTERVAL TIMER FLAG SET W/GO CLEARED.
      MOV #12, @FKCSR4 ;CLR FLAGS

```

```

;*****
;TEST THAT THE 'INTERVAL TIMER' FLAG WILL SET IF GO IS SET
;*****

```

```

FKT6:  SCOPE,6 ;'FK' TEST 6
      MOV #-2000, R0 ;SET UP WAIT LOOP
      BIS #1, @FKCSR4 ;SET 'GO'
      INC R0 ;GIVE FLAG A CHANCE TO SET
      BNE .-2

```



```

2616 006432 032714 002000
2617 006436 001001
2618 006440 104400
2619 006442 012714 000012
2620
2621
2622
2623
2624
2625 006446 104001 000007
2626 006452 052714 000001
2627 006456 032714 002000
2628 006462 001775
2629 006464 042714 000001
2630 006470 052714 000002
2631 006474 032714 002000
2632 006500 001401
2633 006502 104400
2634
2635
2636
2637
2638 006504 104001 000010
2639 006510 052714 000001
2640 006514 032714 002000
2641 006520 001775
2642 006522 000005
2643 006524 005714
2644 006526 001401
2645 006530 104400
2646
2647
2648
2649
2650
2651 006532 104001 000011
2652 006536 012700 176000
2653 006542 004737 016654
2654 006546 006560
2655 006550 005200
2656 006552 001376
2657 006554 104400
2658 006556 000401
2659 006560 022626
2660 006562 004737 016710
2661
2662
2663
2664
2665
2666 006566 104001 000012
2667 006572 012777 006630 172700
2668 006600 012777 000340 172674
2669 006606 012700 176000
2670 006612 005077 172474
2671 006616 012714 000100

```

```

BIT #2000, @FKCSR4 ;TEST IF FLAG SET
BNE .+4
ERROR ;INTERVAL TIMER FLAG DIDN'T SET W/GO SET
MOV #12, @FKCSR4 ;CLR FLAGS

;*****
;TEST THAT THE INTERVAL TIMER FLAG CAN BE CLEARED VIA SETTING BIT 1
;*****
FKT7: SCOPE, 7 ;'FK' TEST 7
BIS #1, @FKCSR4 ;SET 'GO'
BIT #2000, @FKCSR4 ;WAIT FOR FLAG
BEQ .-4
BIC #1, @FKCSR4 ;CLR 'GO'
BIS #2, @FKCSR4 ;CLR INTERVAL TIMER FLAG
BIT #2000, @FKCSR4 ;SEE IF FLAG CLEARED
BEQ .+4
ERROR ;SETTING BIT '1' DIDN'T CLR INTERVAL FLAG.

;*****
;TEST THAT 'RESET' WILL CLR THE 'INTERVAL TIMER' FLAG.
;*****
FKT10: SCOPE, 10 ;'FK' TEST 10
BIS #1, @FKCSR4 ;SET 'GO'
BIT #2000, @FKCSR4 ;WAIT FOR FLAG
BEQ .-4
RESET ;CLR FLAG W/RESET
TST @FKCSR4 ;RESET SHOULD OF CLEARED REG.
BEQ .+4
ERROR ;'RESET' DIDN'T CLR INTERVAL FLAG

;*****
;TEST THAT THE INTERVAL TIMER FLAG CAUSES AN INTERRUPT WITH PROCESSOR PRIORITY 2 0
;*****
FKT11: SCOPE, 11 ;'FK' TEST 11
MOV #-2000, R0 ;SET UP A WAIT LOOP
JSR PC, LDFKVT ;SET UP 'FK' INTR. ADDR.
TAGA ;INTERRUPT SERVICE ADDR.
INC R0 ;WAIT FOR INTERRUPT
BNE .-2
ERROR ;INTERVAL TIMER FLAG DIDN'T CAUSE INTERRUPT
BR .+4
TAGA: POP2SP ;RESET THE STACK
JSR PC, CLRFKV ;CLR 'FK' INTR. ADDR.

;*****
;TEST THAT 'NO' INTERRUPT OCCURS IF 'GO' IS CLEARED
;*****
FKT12: SCOPE, 12 ;'FK' TEST 12
MOV #TAGAB, @FKINT ;SET UP THE INTERRUPT ADDRESS
MOV #340, @FKLVL
MOV #-2000, R0 ;SET UP A WAIT LOOP
CLR @PSW ;CLR PROC. PRIORITY
MOV #100, @FKCSR4 ;SET INTR. ENABLE

```

2672 006622 005200
 2673 006624 001376
 2674 006626 000402
 2675 006630 022626
 2676 006632 104400
 2677 006634 004737 016710

INC RO
 BNE .-2
 BR .+6
 TAGAB: POP2SP ;OK, NO INTERRUPT OCCURRED.
 ERROR ;RESET THE STACK
 JSR PC,CLRFKV ;INTERRUPT OCCURRED W/ 'GO' CLR
 ;CLR 'FK' INTR. ADDR.

2678
 2679
 2680
 2681

 ;TEST THAT 'NO' INTERRUPT OCCURS WITH PROC. @ PRIORITY '4'

2682 006640 104001 000013
 2683 006644 012777 006704 172626
 2684 006652 012777 000340 172622
 2685 006660 012700 176000
 2686 006664 012777 000200 172420
 2687 006672 012714 000101
 2688 006676 005200
 2689 006700 001376
 2690 006702 000402
 2691 006704 022626
 2692 006706 104400
 2693 006710 004737 016710

FKT13: SCOPE,13 ;'FK' TEST 13
 MOV #TAGB,@FKINT ;SET UP THE INTERRUPT ADDRESS
 MOV #340,@FKLVL
 MOV #-2000,RO ;SET UP A WAIT LOOP
 MOV #200,@PSW ;SET PROC. @ PRIORITY '4'
 MOV #101,@FKCSR4 ;SET INTR. ENABLE & GO.
 INC RO
 BNE .-2
 BR .+6
 TAGB: POP2SP ;OK, NO INTERRUPT OCCURRED.
 ERROR ;RESET THE STACK
 JSR PC,CLRFKV ;INTERRUPT OCCURRED W/ PROC. @ PRIORITY '4'
 ;CLR 'FK' INTR. ADDR.

2694
 2695
 2696
 2697
 2698 006714 104001 000014
 2699 006720 012777 006760 172552
 2700 006726 012777 000340 172546
 2701 006734 012700 176000
 2702 006740 012777 000240 172344
 2703 006746 012714 000101
 2704 006752 005200
 2705 006754 001376
 2706 006756 000402
 2707 006760 022626
 2708 006762 104400
 2709 006764 004737 016710

 ;TEST THAT 'NO' INTERRUPT OCCURRES W/ PROC. @ PRIORITY '5'

2710
 2711
 2712
 2713
 2714
 2715 006770 104001 000015
 2716 006774 005001
 2717 006776 004737 016654
 2718 007002 007010
 2719 007004 005201
 2720 007006 001376
 2721
 2722 007010 022626
 2723 007012 004737 016710
 2724 007016 022701 000400
 2725 007022 101402
 2726 007024 104400
 2727 007026 000404

FKT14: SCOPE,14 ;'FK' TEST 14
 MOV #TAGC,@FKINT ;SET UP THE INTERRUPT VECTOR
 MOV #340,@FKLVL
 MOV #-2000,RO ;SET UP WAIT LOOP
 MOV #240,@PSW ;SET PROC. @ PRIORITY '5'
 MOV #101,@FKCSR4 ;SET INTER. ENABLE & GO
 INC RO
 BNE .-2
 BR .+6
 TAGC: POP2SP ;OK, NO INTERRUPT OCCURRED
 ERROR ;RESET STACK POINTER
 JSR PC,CLRFKV ;FK INTERRUPTED WITH PROC. @ PRIORITY '5'
 ;CLR 'FK' INTR. ADDR.

2710
 2711
 2712
 2713
 2714

 ;TEST THAT THE INTERVAL TIMER FLAG IS SET APPROXIMATELY EVERY '500' USEC.

2715 006770 104001 000015
 2716 006774 005001
 2717 006776 004737 016654
 2718 007002 007010
 2719 007004 005201
 2720 007006 001376
 2721
 2722 007010 022626
 2723 007012 004737 016710
 2724 007016 022701 000400
 2725 007022 101402
 2726 007024 104400
 2727 007026 000404

FKT15: SCOPE,15 ;'FK' TEST 15
 CLR R1
 JSR PC,LDFKVT ;SET UP THE INTR. ADDR.
 TAGFA
 INC R1
 BNE .-2

2722 007010 022626
 2723 007012 004737 016710
 2724 007016 022701 000400
 2725 007022 101402
 2726 007024 104400
 2727 007026 000404

TAGFA: POP2SP ;RESET STACK
 JSR PC,CLRFKV ;CLR 'FK' VECTOR ADDR.
 CMP #400,R1 ;CHECK LOW LIMIT
 BLOS .+6 ;BRANCH IF HIGHER
 ERROR ;INTERVAL CLOCK FREQ. TO FAST
 BR FKT16 ;EXIT ON ERROR

```

2728 007030 022701 000600
2729 007034 103001
2730 007036 104400
2731
2732
2733
2734
2735 007040 104001 000016
2736 007044 052714 100200
2737 007050 105714
2738 007052 100401
2739 007054 104400
2740 007056 052714 000010
2741 007062 032714 100200
2742 007066 001401
2743 007070 104400
2744
2745
2746
2747
2748
2749 007072 104001 000017
2750 007076 052714 000201
2751 007102 005777 172370
2752 007106 105714
2753 007110 100001
2754 007112 104400
2755 007114 012714 000012
2756
2757
2758
2759
2760
2761 007120 104001 000020
2762 007124 052714 100001
2763 007130 005714
2764 007132 100401
2765 007134 104400
2766 007136 005777 172334
2767 007142 005714
2768 007144 100001
2769 007146 104400
2770 007150 005014
2771 007152 052714 000012
2772 007156 104001
2773 007160 000020
2774 007162 005037 020402
2775
2776
2777
2778
2779 007166 104001 000021
2780 007172 032777 002000 172124
2781 007200 001076
2782 007202 012737 007210 020406
2783 007210 052714 000001

```

```

CMP #600,R1 ;CHECK HIGH LIMIT
BHIS .+4 ;BRANCH IF LOWER
ERROR ;INTERVAL CLOCK FREQ. TO SLOW
;*****
;TEST THAT 'CLR ASCII' (BIT3) WILL CLEAR THE KEYBOARD FLAG & ERROR BIT.
;*****
FKT16: SCOPE,16 ;'FK' TEST 16
BIS #100200,@FKCSR4 ;SET THE KEYBOARD FLAG
TSTB @FKCSR4 ;TEST THAT DONE WAS SET
BMI .+4
ERROR ;'DONE' DIDN'T SET
BIS #10,@FKCSR4 ;SET 'CLR FLAG'
BIT #100200,@FKCSR4 ;TEST IF FLAGS CLEARED
BEQ .+4
ERROR ;SETTING BIT 3 DIDN'T CLR ASCII FLAGS
;*****
;TEST THAT THE KEYBOARD FLAG IS CLEARED WHEN THE KEYBOARD DATA IS READ
;*****
FKT17: SCOPE,17 ;'FK' TEST 17
BIS #201,@FKCSR4 ;SET THE KEYBOARD FLAG
TST @FKDATA ;READ KEYBOARD DATA
TSTB @FKCSR4 ;RE-TEST IF FLAG CLEARED
BPL .+4
ERROR ;READING KEYBOARD DATA DIDN'T CLR FLAG
MOV #12,@FKCSR4 ;CLR FLAGS
;*****
;TEST THAT THE 'ERROR' FLAG IS CLEARED WHEN THE KEYBOARD DATA IS READ.
;*****
FKT20: SCOPE,20 ;'FK' TEST 20
BIS #100001,@FKCSR4 ;SET THE ERROR BIT
TST @FKCSR4 ;TEST THAT ERROR SET
BMI .+4
ERROR ;ERROR WASN'T SET
TST @FKDATA ;READ THE KEYBOARD DATA
TST @FKCSR4 ;RE-TEST THE ERROR FLAG
BPL .+4
ERROR ;READING KEYBOARD DATA DIDN'T CLR 'ERROR'
CLR @FKCSR4 ;CLR 'GO'
BIS #12,@FKCSR4 ;CLR FLAGS
SCOPE 20
CLR ICOUNT ;RUN FOLLOWING TESTS '1' PASS ONLY
;*****
;TEST THAT THE KEYBOARD FLAG CAN BE SET VIA THE KEYBOARD.
;*****
FKT21: SCOPE,21 ;'FK' TEST 21
BIT #2000,@SWR ;IS MANUAL INHIBIT SW. SET?
BNE FKT24 ;YES, TEST COMPLETE
MOV #TAGFH,RETURN ;RE-SET SCOPE ADDRESS
TAGFH: BIS #1,@FKCSR4 ;SET 'GO'

```

```

2784 007214 104000
2785 007216 021506
2786 007220 005077 172264
2787 007224 012700 177773
2788 007230 005001
2789 007232 005201
2790 007234 001376
2791 007236 005200
2792 007240 001373
2793 007242 105714
2794 007244 100401
2795 007246 104400
2796 007250 012714 000012
2797
2798
2799
2800
2801
2802 007254 104001 000022
2803 007260 104000
2804 007262 021534
2805 007264 005077 172220
2806 007270 012700 177773
2807 007274 005001
2808 007276 005201
2809 007300 001376
2810 007302 005200
2811 007304 001373
2812 007306 105714
2813 007310 100001
2814 007312 104400
2815 007314 012714 000012
2816
2817
2818
2819
2820 007320 104001 000023
2821 007324 012714 000201
2822 007330 104000
2823 007332 021562
2824 007334 005077 172150
2825 007340 012700 177773
2826 007344 005001
2827 007346 005201
2828 007350 001376
2829 007352 005200
2830 007354 001373
2831 007356 005714
2832 007360 100401
2833 007362 104400
2834 007364 005014
2835 007366 052714 000012
2836
2837
2838
2839

```

```

          DISPLAY
          MES6                ;REQUEST CHAR.
          CLR                 ;CLEAR MESSAGE
          MOV                 @VTCSR
          # -5, R0
TAGH:    CLR                 R1
          INC                 R1
          BNE                 .-2
          INC                 R0
          BNE                 TAGH
          TSTB                @FKCSR4      ;TEST IF KEYBOARD FLAG SET
          BMI                 .+4
          ERROR                ;KEYBOARD FLAG DIDN'T SET
          MOV                 #12, @FKCSR4 ;CLR FLAGS

;*****
;TEST THAT THE 'ASCII' FLAG ISN'T SET IF 'GO' IS CLEARED
;*****

FKT22:   SCOPE, 22           ;'FK' TEST 22
          DISPLAY
          MES6A              ;REQUEST CHAR.
          CLR                 ;CLEAR MESSAGE
          MOV                 @VTCSR
          # -5, R0
TAGAH:   CLR                 R1
          INC                 R1
          BNE                 .-2
          INC                 R0
          BNE                 TAGAH
          TSTB                @FKCSR4      ;TEST IF KEYBOARD FLAG SET
          BPL                 .+4
          ERROR                ;ASCII FLAG SET W/ 'GO' CLEARED
          MOV                 #12, @FKCSR4 ;CLR FLAGS

;*****
;TEST THAT THE 'ERROR' FLAG IS SET IF THE KEYBOARD FLAG ISN'T CLEARED
;*****

FKT23:   SCOPE, 23           ;'FK' TEST 23
          MOV                 #201, @FKCSR4 ;SET 'GO'
          DISPLAY
          MES6B              ;REQUEST CHAR.
          CLR                 ;CLEAR MESSAGE
          MOV                 @VTCSR
          # -5, R0
TAGI:    CLR                 R1
          INC                 R1
          BNE                 .-2
          INC                 R0
          BNE                 TAGI
          TST                 @FKCSR4      ;TEST IF 'ERROR' SET
          BMI                 .+4
          ERROR                ;ERROR DIDN'T SET ON 2ND CHAR.
          CLR                 @FKCSR4
          BIS                 #12, @FKCSR4 ;CLR FLAGS

;*****
;END OF 'FK' LOGIC TEST.
;*****

```

```

2840
2841 007372 104001 000024 FKT24: SCOPE,24 ;'FK' TEST 24
2842 007376 104000 DISPLAY ;TEXT 'LOGIC OK;
2843 007400 021610 MES7 ;END OF TEST
2844 007402 104006 ENDTST
2845 007404 000402 BR ;+6
2846 007406 000137 003122 JMP VTLOGIC ;RESTART THE 'VT' LOGIC TEST
2847 007412 000137 006236 JMP FKLOGIC ;RE-RUN 'FK' LOGIC TEST
2848
2849 ;*****
2850 ;FK11 FUNCTION KEYBOARD LIGHT & SWITCH TEST. (16)
2851 ;THIS IS AN OPERATOR INTERVENTION TEST WHICH MONITORS THE FUNCTION KEYS.
2852 ;WHEN A FUNCTION KEY OR KEYS ARE STRUCK ITS CORRESPONDING FUNCTION LIGHT
2853 ;IS LIT. IF THE KEY IS STRUCK A SECOND TIME THE CORRESPONDING FUNCTION
2854 ;LIGHT IS TURNED OFF.
2855 ;*****
2856 007416 012737 007416 020406 FKFUN: MOV #FKFUN, RETURN
2857 007424 104000 DISPLAY
2858 007426 021105 MES1 ;TEXT 'FK FUNCTION KEYBOARD LIGHT TEST'
2859 007430 005000 CLR R0 ;CLR WORKINGS REG'S
2860 007432 005001 CLR R1
2861 007434 005002 CLR R2
2862 007436 004737 016654 JSR PC,LDFKVT ;LOAD THE INTR. ADDR.
2863 007442 007450 SRRVFK ;INTR SERVICE ROUTINE
2864 007444 000001 WAIT ;WAIT FOR INTERRUPT
2865 007446 000776 BR -.2
2866
2867 ;ROUTINE TO 'SERVICE' FUNCTION KEYBOARD INTERRUPTS
2868
2869 007450 005202 SRRVFK: INC R2 ;INTR. CNTR. (EVERY 1000, LITE DATA UPDATED)
2870 007452 105777 172016 TSTB #FKCSR
2871 007456 100004 BPL 1$
2872 007460 022777 000003 172010 CMP #3,#FKDATA ;CONTROL C TYPED?
2873 007466 001436 BEQ EXITF1 ;IF SO-EXITF1
2874 007470 022702 001000 1$: CMP #1000,R2 ;SERVICED '512' FLAGS?
2875 007474 001023 BNE EXITF0 ;NO, EXIT
2876 007476 005002 CLR R2 ;CLR CNTR.
2877 007500 017704 171766 MOV #FKLDA,R4 ;READ SWITCH DATA 'A'
2878 007504 001406 BEQ SRVFOB ;SW. DATA PRESENT?
2879 007506 010037 024436 MOV R0,SAVOLD ;SAVE 'OLD' KEYBOARD DATA
2880 007512 040400 BIC R4,R0 ;CLR ALL CORRESPONDING BITS IN 'OLD' NO.
2881 007514 043704 024436 BIC SAVOLD,R4 ;CLR ALL CORRESPONDING BITS IN 'NEW' NO.
2882 007520 050400 BIS R4,R0 ;SET REMAINING BITS FOR 'NEW' VALUE
2883 007522 017704 171742 SRVFOB: MOV #FKLDB,R4 ;READ SWITCH DATA 'B'
2884 007526 001406 BEQ EXITF0 ;SW. DATA PRESENT?
2885 007530 010137 024436 MOV R1,SAVOLD
2886 007534 040401 BIC R4,R1 ;CLR ALL CORRESPONDING BITS IN 'OLD' WRD.
2887 007536 043704 024436 BIC SAVOLD,R4 ;CLR ALL CORRESPONDING BITS IN 'NEW' WRD.
2888 007542 050401 BIS R4,R1 ;UPDATE 'OLD' WRD WITH 'NEW' WRD.
2889 007544 010077 171722 EXITF0: MOV R0,#FKLDA ;TRANSMIT UPDATED LIGHT DATA 'A'
2890 007550 010177 171714 MOV R1,#FKLDB ;TRANSMIT UPDATED LIGHT DATA 'B'
2891 007554 052777 000012 171712 BIS #12,#FKCSR ;CLR FLAGS.
2892 007562 000002 RTI
2893 007564 052777 000012 171702 EXITF1: BIS #12,#FKCSR
2894 007572 104006 ENDTST
2895 007574 000710 BR FKFUN

```

2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951

007576 012737 007576 020406
007604 104000
007606 021312
007610 012701 020463
007614 005037 024144
007620 005037 024440
007624 005037 024442
007630 005037 024460
007634 004737 016654
007640 007646
007642 000001
007644 000776

007646 105777 171622
007652 100071
007654 117702 171616
007660 032777 040000 171436
007666 001007
007670 005737 024144
007674 001403
007676 005777 171422
007702 100001
007704 105721
007706 120211
007710 001073
007712 005737 024144
007716 001404
007720 005037 024144
007724 112720 000360
007730 022702 000033
007734 001403
007736 005737 024460
007742 001402
007744 052702 000100
007750 110220
007752 005737 024440
007756 001012
007760 022701 020551

```
*****
:KEYBOARD CHARACTER TEST (17)
:THIS TEST VERIFIES THAT ALL CHARACTERS CAN BE TRANSMITTED VIA THE FK11 KEYBOARD.
:WHEN THE TEST IS STARTED, THE TEXT "READY FOR INPUT" IS DISPLAYED ON THE 'VT'.
:THE OPERATOR THEN TYPES IN THREE PASSES OF THE ENTIRE KEYBOARD
:(LOWER CASE, SHIFT AND CONTROL) STARTING AT THE TOP LEFT HAND SIDE OF THE
:KEYBOARD FOR EACH PASS. EACH CHARACTER TRANSMITTED FROM THE KEYBOARD IS VERIFIED
:AGAINST A CHARACTER BUFFER AND DISPLAYED ON THE 'VT'. IF THE TRANSMITTED
:CHARACTER DOESN'T MATCH THE EXPECTED CHARACTER AN AUDIO 'BEEP' IS HEARD
:AND THE CHARACTER IS DISPLAYED IN BOLD UNDERLINED. ALL CHARACTERS RECEIVED
:FROM THIS POINT ON WILL BE TRANSMITTED IN THIS MODE UNTIL EITHER THE CORRECT
:CHARACTER CODE IS RECEIVED OR DATA 'SW15' IS SET TO A '1' ALLOWING THE PROGRAM
:TO CONTINUE. DATA 'SW14' MAY ALSO BE SET TO A '1' AT ANYTIME TO LOOP ON
:THE CURRENT CHARACTER.
*****
```

```
FKCHAR: MOV #FKCHAR, RETURN
        DISPLAY
        MES4 ;TEXT 'FK CHARACTER TEST'
        MOV #CHRTAB-1, R1 ;SET UP CHAR. TABLE POINTER
        CLR FIELDSW ;CLR FIELD CONTROL SOFTWARE SW.
        CLR TEMP1 ;SET UP SOFTWARE SW.'S
        CLR TEMP2
        CLR KSTOR1
        JSR PC, LDFKVT ;LOAD 'FK' VECTOR ADDR.
        CHKFK0
        WAIT ;WAIT FOR 'FK' INTERRUPTS
        BR -2

;ENTERED HERE ON 'FK' INTERRUPTS
CHKFK0: TSTB @FKCSR ;ASCII FLAG SET?
        BPL EXTCHR ;NO, INTERVAL TIMER FLAG
        MOV @FKDATA, R2 ;READ & SAVE DATA
        BIT #SW14, @SWR ;IS THE LOOP SW. SET?
        BNE CHKFK1+2 ;YES, DON'T UPDATE POINTER
        TST FIELDSW ;NO, IS THE ERROR FLAG SET?
        BEQ CHKFK1 ;NO, UPDATE POINTER
        TST @SWR ;YES, IS SCOPE SW. SET?
        BPL CHKFK1+2 ;YES, LOOP ON CURRENT CHAR.
        TSTB (R1)+ ;NO, UPDATE POINTER
        CMPB R2, (R1) ;DOES RECV'D CHAR.=EXPT'D CHAR.
        BNE CHRERR ;NO CHAR. ERROR
        TST FIELDSW ;IS FIELD SW. SET?
        BEQ CHKFK2 ;NO
        CLR FIELDSW ;YES, CLR IT
        MOVB #360, (R0)+ ;CLR FIELD CONTROL CHAR.
        CMP #33, R2
        BEQ .+10
        TST KSTOR1 ;DOING CONTROL PASS?
        BEQ .+6 ;NO, CONTINUE
        BIS #100, R2 ;YES, ADD BIT SO CHAR. IS VISIBLE
        MOVB R2, (R0)+ ;MOV RECV'D CHAR. TO 'VT' BUFFER.
        TST TEMP1
        BNE CHKFK4
        CMP #UPCASE-1, R1 ;STARTING 'UPPER CASE' PASS?
```

```

2952 007764 001007          BNE      CHKFK4          ;NO
2953 007766 104000          DISPLAY
2954 007770 023077          MES22          ;TEXT "INPUT UPPER CASE"
2955 007772 012737 007576 024130  MOV      #FKCHAR,AVECTR ;RE-SET '↑A' RESTART ADDR.
2956 010000 005237 024440          INC      TEMP1          ;SET PRINT INHIBIT SW.
2957 010004 005737 024442          CHKFK4: TST      TEMP2
2958 010010 001012          BNE      EXTCHR
2959 010012 022701 020637          CMP      #CONTRL-1,R1 ;STARTING CONTROL PASS?
2960 010016 001007          BNE      EXTCHR          ;NO
2961 010020 104000          DISPLAY
2962 010022 023143          MES23          ;TEXT "INPUT CONROL CHAR.S"
2963 010024 012737 007576 024130  MOV      #FKCHAR,AVECTR ;RE-SET '↑A' RESTART ADDR.
2964 010032 005237 024442          INC      TEMP2          ;SET DISPLAY INHIBIT SW.
2965 010036 022701 020653          EXTCHR: CMP      #CNTRIT,R1 ;STARTING CONTROL PASS
2966 010042 001002          BNE      .+6            ;NO, CONTINUE
2967 010044 005237 024460          INC      KSTOR1         ;YES, SET UP TO DISPLAY INVISIBLE CHAR.
2968 010050 052777 000002 171416  BIS      #2,#FKCSR      ;CLR INTERVAL TIMER FLAG
2969 010056 022701 020726          CMP      #CHREND+1,R1 ;FINISHED?
2970 010062 001401          BEQ      .+4            ;YES, RE-START TEST
2971 010064 000002          RTI
2972 010066 004737 016710          JSR      PC,CLRFKV      ;CLR INTERRUPT VECTOR
2973 010072 022626          POP2SP
2974 010074 104006          ENDTST
2975 010076 000637          BR       FKCHAR         ;RESTART TEST
2976 010100 052777 000200 171402  CHRERR: BIS      #200,#VTCSR ;ISSUE AUDIO BEEP ON ERROR
2977 010106 005737 024144          TST      FIELD5W        ;IS FIELD SW SET?
2978 010112 001306          BNE      CHKFK2         ;YES
2979 010114 005237 024144          INC      FIELD5W        ;NO, SET 'FIELD' SW.
2980 010120 112720 000374          MOV8     #374,(R0)+     ;SELECT 'BOLD-UNDERLINE'
2981 010124 000701          BR       CHKFK2         ;SET FIELD CONTROL SOFTWARE SW.
2982
2983 ;*****
2984 ;FUNCTION KEYBOARD REPEATABILITY TEST (20)
2985 ;THIS TEST VERIFIES THAT A SELECTED CHARACTER OR CHARACTERS (MAXIMUM OF TWO)
2986 ;ARE RECEIVED CORRECTLY FROM THE KEYBOARD. WHEN THE 'REPEATABILITY' TEST IS
2987 ;SELECTED THE PROGRAM HALTS. LOAD THE CODE FOR THE CHARACTER TO BE VERIFIED
2988 ;IN THE DATA SWITCHES AND PRESS CONTINUE. THE PROGRAM WILL HALT AGAIN.
2989 ;LOAD THE CODE FOR THE 2ND CHARACTER TO BE VERIFIED. IF ONLY
2990 ;ONE CHARACTER IS BEING CHECKED, PRESS CONTINUE THE 2ND TIME WITHOUT CHANG-
2991 ;ING THE SWITCH SETTING. AS INPUTTED CHARACTERS ARE VERIFIED THEY ARE
2992 ;ALSO DISPLAYED ON THE 'VT'. IF AN ERROR IS DETECTED, THE AUDIO 'BEEP' IS
2993 ;SOUNDED AND THE RECEIVED CHARACTER IS DISPLAYED AS 'BOLD-UNDERLINED'.
2994 ;*****
2995 010126 005737 024154          FKREPT: TST      TTYSWH ;TTY AVAILABLE?
2996 010132 001404          BEQ      FKRPTA         ;NO, HALT TO GET CHAR. CODE
2997 010134 104012          PRINT
2998 010136 022506          MES15
2999 010140 104015          ASEMBL ;REQUEST CHAR. CODES
3000 010142 000406          BR       FKRPTB         ;DECODE CODES
3001 010144 000000          FKRPTA: HALT
3002 010146 117703 171152          MOV8     @SWR,R3        ;GET 1ST CHAR. FROM THE SWITCHES
3003 010152 000000          HALT
3004 010154 117704 171144          MOV8     @SWR,R4        ;SAVE 1ST CHAR.
3005 010160 042703 000200          FKRPTB: BIC      #200,R3 ;GET 2ND CHAR. FROM THE SWITCHES
3006 010164 042704 000200          BIC      #200,R4        ;SAVE '2ND' CHAR.
3007 010170 012737 010160 020406          MOV      #FKRPTB,RETURN

```

```

3008 010176 104000          DISPLAY
3009 010200 022433          MES14          ;TEXT 'FK REPEATIBILITY TEST'
3010 010202 012737 010126 024130  MOV      #FKREPT,AVECTR
3011 010210 005037 024144          CLR      FIELD SW          ;CLR FIELD CONTROL SOFTWARE SW.
3012 010214 005005          CLR      R5                ;CLR WORKING REG.
3013 010216 004737 017062          JSR      PC,TTYENB         ;ENABLE TTY INTERRUPTS
3014 010222 004737 016654          JSR      PC,LDFKVT        ;LOAD 'FK' INTR. ADDR.
3015 010226 010234          FREPTO
3016 010230 000001          WAIT
3017 010232 000776          BR      -2                ;WAIT FOR 'FK' INTR.
3018
3019          ;ENTERED HERE ON 'FK' INTERRUPTS FOR THE REPEATIBILITY TEST
3020 010234 105777 171234  FREPTO: TSTB   @FKCSR          ;ASCII FLAG SET?
3021 010240 100035          BPL     EXREPT            ;NO, INTERVAL TIMER FLAG
3022 010242 117701 171230  MOVB   @FKDATA,R1        ;YES, READ & SAVE DATA
3023 010246 005705          TST     R5                ;COMPARE 1ST OR 2ND CHAR.?
3024 010250 001003          BNE     FREPT1           ;2ND
3025 010252 120103          CMPB   R1,R3             ;RCV'D = EXPT'D CHAR.?
3026 010254 001416          BEQ     REPTOK           ;YES
3027 010256 000402          BR      REPTER          ;NO, SET UP ERROR
3028 010260 120104  FREPT1: CMPB   R1,R4             ;RCV'D = EXPT'D CHAR.?
3029 010262 001413          BEQ     REPTOK           ;YES
3030 010264 052777 000200 171216  REPTER: BIS    #AUDIO,@VTCSR ;ISSUE AUDIO BEEP
3031 010272 005737 024144          TST     FIELD SW        ;FIELD SW. SET?
3032 010276 001014          BNE     FREPT2           ;YES
3033 010300 005237 024144          INC     FIELD SW        ;NO, SET IT
3034 010304 112720 000374          MOVB   #374,(R0)+       ;SEND BAD CHAR. AS 'BOLD-UNDERLINED'
3035 010310 000407          BR      FREPT2
3036 010312 005737 024144  REPTOK: TST     FIELD SW        ;FIELD SW. SET
3037 010316 001404          BEQ     FREPT2           ;YES
3038 010320 005037 024144          CLR     FIELD SW        ;NO, CLR IT
3039 010324 112720 000360          MOVB   #360,(R0)+       ;CLR FIELD CONTROL CHAR.
3040 010330 005105  FREPT2: COM     R5                ;SET UP SOFTWARE SW.
3041 010332 110120          MOVB   R1,(R0)+         ;TRANSMIT RCV'D CHAR.
3042 010334 052777 000002 171132  EXREPT: BIS    #2,@FKCSR     ;CLR INTERVAL TIMER FLAG
3043 010342 000002          RTI
3044
3045          ;*****
3046          ;VT20 SYSTEM EXERCISER TEST. (TEST 21)
3047          ;THIS TEST IS DESIGNED TO EXERCISE THE VT20 IN A SYSTEM ENVIRONMENT. THE
3048          ;TEST UTILIZES THE VT20 (DISPLAYS & KEYBOARDS) AND ALL ASSOCIATED DL11'S
3049          ;IN CONJUNCTION WITH A HOST COMPUTER. TO RUN THIS TEST THE DL11 DATA
3050          ;HANDLING ROUTINE MUST BE LOADED INTO THE HOST COMPUTER. THIS ROUTINE
3051          ;RECEIVES AND RE-TRANSMITS DATA TO VT20 BEING TESTED. WHEN THE SYSTEM TEST
3052          ;IS STARTED THE PROGRAM WAITS FOR CHARACTERS TO BE INPUTTED FROM THE
3053          ;KEYBOARD. AS THESE CHARACTERS ARE RECEIVED THEY ARE STORED IN A VT BUFFER
3054          ;AND DISPLAYED. THE PROGRAM WILL ACCEPT UP TO 384 CHARACTERS (6 COMPLETE
3055          ;LINES) FROM THE KEYBOARD. THESE 384 CHARACTERS, OR ANY PART THERE OF,
3056          ;CAN BE AUTOMATICALLY GENERATED BY TYPING A '↑A'. THE OPERATOR MAY AT
3057          ;THIS TIME OR ANYTIME PRIOR TO THIS, TYPE 'ALTMODE'. THIS WILL ENABLE
3058          ;THE INPUTTED CHARACTERS TO BE TRANSMITTED TO THE HOST COMPUTER.
3059          ;UPON RECEIVING THE COMPLETE DATA TRANSFER, THE HOST COMPUTER WILL
3060          ;RE-TRANSMIT THE DATA BACK TO THE VT20. THE RECEIVED DATA IS THEN
3061          ;VERIFIED AGAINST THE DATA ORGINALLY TRANSMITTED AND THEN DISPLAYED.
3062          ;IF THE RECEIVED DATA DOESN'T MATCH THE TRANSMITTED DATA, THE RECEIVED
3063          ;CHACACTER IS DISPLAYED AS 'BOLD-UNDERLINED'.
          ;THE SYSTEM TEST USES AN ERROR REPORTING SCHEME OF DISPLAYING ALL

```



```

; ERRORS AS THEY ARE ENCOUNTERED AS UNIQUE CURSOR CHARACTERS. THESE
; CHARACTERS IDENTIFY THE PARTICULAR ERROR DETECTED. TYPING A 't' AT
; ANYTIME WILL ENABLE THE TEST TO BE RE-STARTED.
; TYPING A 'T' AFTER THE DATA HAS BEEN INPUTTED WILL ENABLE
; FOR A CONTINUOUS TRANSFER OF DATA. TYPING 't' A SECOND TIME
; WILL STOP THE DATA TRANSFER (ON COMPLETION OF CURRENT TRANSFER)
; AND ALLOW THE PROGRAM TO RESUME NORMAL OPERATION.
; *****

```

```

3064
3065
3066
3067
3068
3069
3070
3071
3072
3073 010344 012737 010344 020406 SYSTST: MOV #SYSTST, RETURN
3074 010352 000005 RESET
3075 010354 012737 010344 024130 MOV #SYSTST, AVECTR ; SET UP THE RESTART ADDR.
3076 010362 012777 021651 171022 MOV #MES9, AVTOSAR ; DISPLAY TEST HEADER ON UNIT '0'
3077 010370 012777 021651 171030 MOV #MES9, AVT1SAR ; DISPLAY TEST HEADER ON UNIT '1'
3078 010376 022737 000004 024136 CMP #4, UNITNO ; 4 UNITS?
3079 010404 001013 BNE 2$
3080 010406 012777 021651 171042 MOV #MES9, AVT3SAR ; DISPLAY TEST HEADER ON UNIT '3'
3081 010414 005277 171040 INC AVT3CSR
3082 010420 012777 021651 171014 1$: MOV #MES9, AVT2SAR ; DISPLAY TEST HEADER ON UNIT '2'
3083 010426 005277 171012 INC AVT2CSR
3084 010432 000404 BR 3$
3085
3086 010434 022737 000003 024136 2$: CMP #3, UNITNO ; 3 UNITS?
3087 010442 001766 BEQ 1$
3088 010444 005277 170744 3$: INC AVT0CSR
3089 010450 005277 170754 INC AVT1CSR
3090 010454 104020 DELAYL
3091 010456 104020 DELAYL
3092 010460 104005 EOSBUF
3093 010462 012777 000340 170622 SYSIT1: MOV #340, APSW ; RE-LOAD DATA BUFFER W/ 'EOS'
3094 010470 012700 024170 MOV #RECSWO, RO ; SET PROC. PRIO. 37
3095 010474 005020 CLR (RO)+ ; SET UP TO CLR SOFTWARE USER SW.'S
3096 010476 022700 024424 CMP #START, RO ; DONE?
3097 010502 001374 BNE -6 ; NO
3098 010504 012777 011766 171026 MOV #RECVO, ARINTO ; LOAD DL11 REC. VECTOR ADDR.'S
3099 010512 012777 000200 171022 MOV #200, ARLVLO ; BR LEVEL '4'
3100 010520 012777 012000 171016 MOV #RCV1, ARINT1
3101 010526 012777 000200 171012 MOV #200, ARLV1
3102 010534 012777 011706 171036 MOV #TRANO, AXINTO ; LOAD DL11 TRANS. VECTOR ADDR.'S
3103 010542 012777 000200 171032 MOV #200, AXLVLO
3104 010550 012777 011722 171026 MOV #TRANI, AXINT1
3105 010556 012777 000200 171022 MOV #200, AXLV1
3106 010564 012777 011626 170546 MOV #SERFKO, AFK0INT ; LOAD 'FK' VECTOR ADDRESSES
3107 010572 012777 000200 170542 MOV #200, AFK0VL
3108 010600 012777 011642 170546 MOV #SERFK1, AFK1INT
3109 010606 012777 000200 170542 MOV #200, AFK1VL
3110 010614 012700 024474 MOV #VTBUFF, RO ; LOAD 'VT' BUFFER POINTERS
3111 010620 010002 MOV RO, R2
3112 010622 004737 011610 JSR PC, DISPMS ; DISPLAY HEADER MESSAGE
3113 010626 012702 026444 MOV #VTBUFF+1000., R2
3114 010632 010077 170554 MOV RO, AVTOSAR ; LOAD 'VT' STARTING ADDR. REG.'S
3115 010636 010277 170564 MOV R2, AVT1SAR
3116 010642 004737 011610 JSR PC, DISPMS
3117 010646 012700 024563 MOV #VTBUFF+55., RO
3118 010652 010037 024404 MOV RO, BUFORD ; R0 BUFFER POINTER FOR UNIT '0'
3119 010656 010037 024414 MOV RO, BUFORD1 ; R1 BUFFER POINTER FOR UNIT '0'

```

3120	010662	012702	026533		MOV	#VTBUFF+1055.,R2	
3121	010666	010237	024406		MOV	R2, BUF1R0	;R0 BUFFER PCINTER FOR UNIT '1'
3122	010672	010237	024416		MOV	R2, BUF1R1	;R1 BUFFER POINTER FOR UNIT '1'
3123	010676	012737	010726	000004	MOV	#DL11A, @#4	;SET UP TIME-OUT FOR MISSING DL'S
3124	010704	012737	000340	000006	MOV	#340, @#6	
3125	010712	012777	000100	170600	MOV	#100, @RCSR0	;ENABLE DL11 REC. INTR'S
3126	010720	012777	000100	170632	MOV	#100, @XCSR0	;ENABLE DL11 TRANS, INTR.'S
3127	010726	012737	010750	000004	DL11A: MOV	#DL11B, @#4	
3128	010734	012777	000100	170560	MOV	#100, @RCSR1	
3129	010742	012777	000100	170612	MOV	#100, @XCSR1	
3130	010750	022737	000004	024136	DL11B: CMP	#4, UNITNO	;4 UNITS?
3131	010756	001133			BNE	TAG2X	;NO TEST FOR 3
3132	010760	012777	012024	170566	MOV	#REC3, @RINT3	;LOAD DL11 REC VECTOR FOR UNIT '3'
3133	010766	012777	000200	170562	MOV	#200, @RLVL3	;BR LEVEL '4'
3134	010774	012777	011752	170612	MOV	#TRAN3, @XINT3	;LOAD DL11 XMIT VECTOR FOR UNIT '3'
3135	011002	012777	000200	170606	MOV	#200, @XLVL3	;BE LEVEL '4'
3136	011010	012777	011672	170366	MOV	#SERFK3, @FK3INT	;LOAD FK VECTOR FOR UNIT '3'
3137	011016	012777	000200	170362	MOV	#200, @FK3LVL	;BR LEVEL '4'
3138	011024	012702	032364		MOV	#VTBUFF+3000.,R2	
3139	011030	010277	170422		MOV	R2, @VT3SAR	;LOAD 'VT'3 STARTING ADDR. REG
3140	011034	004737	011610		JSR	PC, DISPMS	
3141	011040	012702	032453		MOV	#VTBUFF+3055.,R2	
3142	011044	010237	024412		MOV	R2, BUF3R0	;BUFFER POINTER FOR UNIT '3'
3143	011050	010237	024422		MOV	R2, BUF3R1	
3144	011054	012737	011076	000004	MOV	#DL11C, @#4	
3145	011062	012777	000100	170436	MOV	#100, @RCSR3	;ENABLE DL11 FOR UNIT '3' TO INTERRUPT
3146	011070	012777	000100	170470	MOV	#100, @XCSR3	
3147	011076	012777	000101	170354	DL11C: MOV	#101, @VT3CSR	;START VT
3148	011104	012777	000101	170266	MOV	#101, @FK3CSR	;ENABLE FK INTERRUPT
3149	011112	012777	012012	170430	TAG1X: MOV	#REC2, @RINT2	;LOAD DL11 REC VECTOR FOR UNIT '2'
3150	011120	012777	000200	170424	MOV	#200, @RLVL2	;BR LEVEL '4'
3151	011126	012777	011736	170454	MOV	#TRAN2, @XINT2	;LOAD DL11 XMIT VECTOR FOR UNIT '2'
3152	011134	012777	000200	170450	MOV	#200, @XLVL2	;BR LEVEL '4'
3153	011142	012777	011656	170220	MOV	#SERFK2, @FK2INT	;LOAD FK VECTOR FOR UNIT '2'
3154	011150	012777	000200	170214	MOV	#200, @FK2LVL	;BR LEVEL '4'
3155	011156	012702	030414		MOV	#VTBUFF+2000.,R2	
3156	011162	010277	170254		MOV	R2, @VT2SAR	;LOAD 'VT'2 STARTING ADDR. REG
3157	011166	004737	011610		JSR	PC, DISPMS	
3158	011172	012702	030503		MOV	#VTBUFF+2055.,R2	
3159	011176	010237	024410		MOV	R2, BUF2R0	;BUFFER POINTER FOR UNIT '2'
3160	011202	010237	024420		MOV	R2, BUF2R1	
3161	011206	012737	011230	000004	MOV	#DL11D, @#4	
3162	011214	012777	000100	170302	MOV	#100, @RCSR2	;ENABLE DL11 FOR UNIT '2' TO INTERRUPT
3163	011222	012777	000100	170334	MOV	#100, @XCSR2	
3164	011230	012777	000101	170206	DL11D: MOV	#101, @VT2CSR	;START VT
3165	011236	012777	000101	170120	MOV	#101, @FK2CSR	;ENABLE FK INTERRUPTS
3166	011244	000404			BR	TAG3X	
3167							
3168	011246	022737	000003	024136	TAG2X: CMP	#3, UNITNO	;3 UNITS?
3169	011254	001716			BEQ	TAG1X	;YES, SET UP UNIT 2
3170	011256	012737	000006	000004	TAG3X: MOV	#6, @#4	;RESET THE TIMEOUT VECTOR
3171	011264	012737	000004	000006	MOV	#4, @#6	
3172	011272	012777	000101	170114	MOV	#101, @VT0CSR	;START VT'S
3173	011300	012777	000101	170122	MOV	#101, @VT1CSR	
3174	011306	012777	000101	170020	MOV	#101, @FK0CSR	;ENABLE 'FK' INTERRUPTS
3175	011314	012777	000101	170026	MOV	#101, @FK1CSR	

3176	011322	005077	167764		CLR	QPSW		:SET PROC. PRIO. QD
3177								
3178	011326	012700	024504		SERVICE: MOV	#VTBUFF+08.,R0		
3179	011332	013702	024230		MOV	CHRCT0,R2		
3190	011336	104024			BINDEC			:UPDATE THE CHARACTER COUNTER
3181	011340	012700	024522		MOV	#VTBUFF+22.,R0		
3182	011344	013702	024314		MOV	RECTR0,R2		
3183	011350	104024			BINDEC			:UPDTAE THE RECV'D CHAR. COUNTER
3184	011352	012700	024540		MOV	#VTBUFF+36.,R0		
3185	011356	013702	024304		MOV	TRANFC,R2		
3186	011362	104024			BINDEC			:UPDATE THE TRANSFER COUNTER
3187	011364	012700	024555		MOV	#VTBUFF+49.,R0		
3188	011370	013702	024334		MOV	ERFLG0,R2		
3189	011374	104024			BINDEC			:UPDATE THE ERROR COUNTER
3190	011376	012700	026454		MOV	#VTBUFF+1008.,R0		
3191	011402	013702	024232		MOV	CHRCT1,R2		
3192	011406	104024			BINDEC			
3193	011410	012700	026472		MOV	#VTBUFF+1022.,R0		
3194	011414	013702	024316		MOV	RECTR1,R2		
3195	011420	104024			BINDEC			
3196	011422	012700	026510		MOV	#VTBUFF+1036.,R0		
3197	011426	013702	024306		MOV	TRANF1,R2		
3198	011432	104024			BINDEC			
3199	011434	012700	026525		MOV	#VTBUFF+1049.,R0		
3200	011440	013702	024336		MOV	ERFLG1,R2		
3201	011444	104024			BINDEC			
3202	011446	023727	024136	000003	CMP	UNITNO,#3		:RUNNING '3' UNITS ?
3203	011454	002724			BLT	SERVICE		:NO, RETURN TO SERVICE
3204	011456	012700	030424		MOV	#VTBUFF+2008.,R0		
3205	011462	013702	024234		MOV	CHRCT2,R2		
3206	011466	104024			BINDEC			:UPDATE THE CHARACTER COUNTER
3207	011470	012700	030442		MOV	#VTBUFF+2022.,R0		
3208	011474	013702	024320		MOV	RECTR2,R2		
3209	011500	104024			BINDEC			:UPDTAE THE RECV'D CHAR. COUNTER
3210	011502	012700	030460		MOV	#VTBUFF+2036.,R0		
3211	011506	013702	024310		MOV	TRANF2,R2		
3212	011512	104024			BINDEC			:UPDATE THE TRANSFER COUNTER
3213	011514	012700	030475		MOV	#VTBUFF+2049.,R0		
3214	011520	013702	024340		MOV	ERFLG2,R2		
3215	011524	104024			BINDEC			:UPDATE THE ERROR COUNTER
3216	011526	023727	024136	000004	CMP	UNITNO,#4		:RUNNING '4' UNITS ?
3217	011534	002674			BLT	SERVICE		:NO, RETURN TO SERVICE
3218	011536	012700	032374		MOV	#VTBUFF+3008.,R0		
3219	011542	013702	024236		MOV	CHRCT3,R2		
3220	011546	104024			BINDEC			
3221	011550	012700	032412		MOV	#VTBUFF+3022.,R0		
3222	011554	013702	024322		MOV	RECTR3,R2		
3223	011560	104024			BINDEC			
3224	011562	012700	032430		MOV	#VTBUFF+3036.,R0		
3225	011566	013702	024312		MOV	TRANF3,R2		
3226	011572	104024			BINDEC			
3227	011574	012700	032445		MOV	#VTBUFF+3049.,R0		
3228	011600	013702	024342		MOV	ERFLG3,R2		
3229	011604	104024			BINDEC			
3230	011606	000647			BR	SERVICE		
3231								

```

3232
3233
3234 011610 012701 022157
3235 011614 111122
3236 011616 122721 000031
3237 011622 001374
3238 011624 000207
3239
3240
3241 011626 004737 012036
3242 011632 004737 012444
3243 011636 000137 012272
3244
3245
3246
3247 011642 004737 012102
3248 011646 004737 012444
3249 011652 000137 012316
3250
3251
3252
3253 011656 004737 012152
3254 011662 004737 012444
3255 011666 000137 012342
3256
3257
3258
3259 011672 004737 012222
3260 011676 004737 012444
3261 011702 000137 012366
3262
3263
3264 011706 004737 012036
3265 011712 004737 013130
3266 011716 000137 012272
3267
3268
3269
3270 011722 004737 012102
3271 011726 004737 013130
3272 011732 000137 012316
3273
3274
3275
3276 011736 004737 012152
3277 011742 004737 013130
3278 011746 000137 012342
3279
3280
3281
3282 011752 004737 012222
3283 011756 004737 013130
3284 011762 000137 012366
3285
3286
3287

```

```

;ENTERED HERE TO MOVE A MESSAGE INTO THE VT BUFFER

```

```

DISPMS: MOV      #MES80,R1
        MOV8    (R1),(R2)+
        CMPB   #EOS,(R1)+
        BNE    DISPMS+4
        RTS    PC

```

```

;ENTERED HERE ON 'FK' INTERRUPTS FROM UNIT '0'

```

```

SERFK0: JSR      PC,SETBFO           ;SET UP BUFFER POINTERS
        JSR      PC,SERVFK         ;SERVICE 'FK' INTERRUPT
        JMP      SAVBFO           ;SAVE BUFFER POINTERS ON RETURN

```

```

;ENTERED HERE ON 'FK' INTERRUPTS FROM UNIT '1'

```

```

SERFK1: JSR      PC,SETBF1         ;SET UP BUFFER POINTERS
        JSR      PC,SERVFK         ;SERVICE 'FK' INTERRUPT
        JMP      SAVBF1           ;SAVE BUFFER POINTERS ON RETURN

```

```

;ENTERED HERE ON 'FK' INTERRUPTS FROM UNIT '2'

```

```

SERFK2: JSR      PC,SETBF2         ;SET UP BUFFER POINTERS
        JSR      PC,SERVFK         ;SERVICE 'FK' INTERRUPT
        JMP      SAVBF2           ;SAVE BUFFER POINTERS

```

```

;ENTERED HERE ON 'FK' INTERRUPTS FROM UNIT '3'

```

```

SERFK3: JSR      PC,SETBF3         ;SET UP BUFFER POINTERS
        JSR      PC,SERVFK         ;SERVICE 'FK' INTERRUPT
        JMP      SAVBF3           ;SAVE BUFFER POINTERS

```

```

;ENTERED HERE ON DL11 TRANSMITTER INTERRUPTS FROM UNIT '0'

```

```

TRAN0:  JSR      PC,SETBFO         ;SET UP BUFFER POINTERS
        JSR      PC,TRANSMT
        JMP      SAVBFO

```

```

;ENTERED HERE ON DL11 TRANSMITTER INTERRUPTS FROM UNIT '1'

```

```

TRAN1:  JSR      PC,SETBF1         ;SET UP BUFFER POINTERS
        JSR      PC,TRANSMT
        JMP      SAVBF1

```

```

;ENTERED HERE ON DL11 TRANSMITTER INTERRUPTS FROM UNIT '2'

```

```

TRAN2:  JSR      PC,SETBF2         ;SET UP BUFFER POINTERS
        JSR      PC,TRANSMT         ;SERVICE TRANSMITTER INTERRUPT
        JMP      SAVBF2

```

```

;ENTERED HERE ON DL11 TRANSMITTER INTERRUPTS FROM UNIT '3'

```

```

TRAN3:  JSR      PC,SETBF3         ;SET UP BUFFER POINTERS
        JSR      PC,TRANSMT
        JMP      SAVBF3

```

```

;ENTERED HERE ON DL11 RECEIVER INTERRUPTS FROM UNIT '0'

```

```

3288 011766 004737 012036
3289 011772 004737 013362
3290 011776 000535
3291
3292
3293 012000 004737 012102
3294 012004 004737 013362
3295 012010 000542
3296
3297
3298
3299 012012 004737 012152
3300 012016 004737 013362
3301 012022 000547
3302
3303
3304
3305 012024 004737 012222
3306 012030 004737 013362
3307 012034 000554
3308
3309
3310
3311 012036 012637 024460
3312 012042 010046
3313 012044 010146
3314 012046 010246
3315 012050 010346
3316 012052 010446
3317 012054 013700 024404
3318 012060 013701 024414
3319 012064 005002
3320 012066 005003
3321 012070 012737 024563 024424
3322 012076 000177 012356
3323
3324
3325
3326 012102 012637 024460
3327 012106 010046
3328 012110 010146
3329 012112 010246
3330 012114 010346
3331 012116 010446
3332 012120 013700 024406
3333 012124 013701 024416
3334 012130 012702 000002
3335 012134 012703 000014
3336 012140 012737 026533 024424
3337 012146 000177 012306
3338
3339
3340
3341 012152 012637 024460
3342 012156 010046
3343 012160 010146
    
```

```

RECVO: JSR PC,SETBFO
        JSR PC,RECVIT
        BR SAVBFO
;ENTERED HERE ON DL11 RECEIVER INTERRUPTS FROM UNIT '1'

RECV1: JSR PC,SETBF1
        JSR PC,RECVIT
        BR SAVBF1
;ENTERED HERE ON DL11 RECEIVER INTERRUPTS FROM UNIT '2'

RECV2: JSR PC,SETBF2 ;SET UP BUFFER POINTERS
        JSR PC,RECVIT ;SERVICE RECEIVER INTERRUPT
        BR SAVBF2
;ENTERED HERE ON DL11 RECEIVER INTERRUPTS FROM UNIT '3'

RECV3: JSR PC,SETBF3 ;SET UP BUFFER POINTERS
        JSR PC,RECVIT ;SERVICE RECEIVER INTERRUPT
        BR SAVBF3
;SUBROUTINE ENTERED TO SET UP BUFFER POINTERS FOR UNIT '0'

SETBFO: MOV (SP)+,KSTOR1 ;SAVE THE RETURN ADDRESS
        MOV RO,-(SP) ;SAVE THE WORKING REG.'S
        MOV R1,-(SP)
        MOV R2,-(SP)
        MOV R3,-(SP)
        MOV R4,-(SP)
        MOV BUFORD,RO
        MOV BUFOR1,R1
        CLR R2 ;R2 IS INDEX OFFSET FOR ADDRESSING
        CLR R3 ;R3 IS INDEX OFFSET FOR DEVICE ADDRESSING
        MOV #VTBUFF+55.,START
        JMP @KSTOR1 ;RETURN TO CALL
;SUBROUTINE ENTERED TO SET UP BUFFER POINTERS FOR UNIT '1'

SETBF1: MOV (SP)+,KSTOR1 ;SAVE THE RETURN ADDRESS
        MOV RO,-(SP) ;SAVE THE WORKING REG.'S
        MOV R1,-(SP)
        MOV R2,-(SP)
        MOV R3,-(SP)
        MOV R4,-(SP)
        MOV BUF1RO,RO
        MOV BUF1R1,R1
        MOV #2,R2 ;R2 IS INDEX OFFSET FOR ADDRESSING
        MOV #12.,R3 ;R3 IS INDEX OFFSET FOR DEVICE ADDRESSING
        MOV #VTBUFF+1055.,START
        JMP @KSTOR1 ;RETURN TO CALL
;SUBROUTINE ENTERED TO SET UP BUFFER POINTERS FOR UNIT '2'

SETBF2: MOV (SP)+,KSTOR1 ;SAVE THE RETURN ADDRESS
        MOV RO,-(SP) ;SAVE THE WORKING REG.'S
        MOV R1,-(SP)
    
```

```

3344 012162 010246      MOV      R2,-(SP)
3345 012164 010346      MOV      R3,-(SP)
3346 012166 010446      MOV      R4,-(SP)
3347 012170 013700 024410  MOV      BUF2R0,R0
3348 012174 013701 024420  MOV      BUF2R1,R1
3349 012200 012702 000004      MOV      #4,R2      ;R2 IS INDEX OFFSET FOR ADDRESSING
3350 012204 012703 000030      MOV      #24,R3     ;R3 IS INDEX OFFSET FOR DEVICE ADDR.
3351 012210 012737 030503 024424  MOV      #VTBUFF+2055.,START
3352 012216 000177 012236      JMP      @KSTOR1    ;RETURN TO CALL
3353
3354      ;SUBROUTINE ENTERED TO SET UP BUFFER POINTERS FOR UNIT '3'
3355
3356 012222 012637 024460  SETBF3: MOV      (SP)+,KSTOR1      ;SAVE THE RETURN ADDRESS
3357 012226 010046      MOV      R0,-(SP)      ;SAVE THE WORKING REG.'S
3358 012230 010146      MOV      R1,-(SP)
3359 012232 010246      MOV      R2,-(SP)
3360 012234 010346      MOV      R3,-(SP)
3361 012236 010446      MOV      R4,-(SP)
3362 012240 013700 024412  MOV      BUF3R0,R0
3363 012244 013701 024422  MOV      BUF3R1,R1
3364 012250 012702 000006      MOV      #6,R2
3365 012254 012703 000044      MOV      #36,R3
3366 012260 012737 032453 024424  MOV      #VTBUFF+3055.,START
3367 012266 000177 012166      JMP      @KSTOR1    ;RETURN TO CALL
3368
3369      ;SUBROUTINE ENTERED TO SAVE BUFFER POINTERS FOR UNIT '0'
3370
3371 012272 010037 024404  SAVBFO: MOV      R0,BUF0R0
3372 012276 010137 024414  MOV      R1,BUF0R1
3373 012302 012604      MOV      (SP)+,R4      ;RESTORE THE WORKING REG.'S
3374 012304 012603      MOV      (SP)+,R3
3375 012306 012602      MOV      (SP)+,R2
3376 012310 012601      MOV      (SP)+,R1
3377 012312 012600      MOV      (SP)+,R0
3378 012314 000002      RTI
3379
3380      ;SUBROUTINE ENTERED TO SAVE BUFFER POINTERS FOR UNIT '1'
3381
3382 012316 010037 024406  SAVBF1: MOV      R0,BUF1R0
3383 012322 010137 024416  MOV      R1,BUF1R1
3384 012326 012604      MOV      (SP)+,R4      ;RESTORE THE WORKING REG.'S
3385 012330 012603      MOV      (SP)+,R3
3386 012332 012602      MOV      (SP)+,R2
3387 012334 012601      MOV      (SP)+,R1
3388 012336 012600      MOV      (SP)+,R0
3389 012340 000002      RTI
3390
3391      ;SUBROUTINE ENTERED TO SAVE BUFFER POINTERS FOR UNIT '2'
3392
3393 012342 010037 024410  SAVBF2: MOV      R0,BUF2R0
3394 012346 010137 024420  MOV      R1,BUF2R1
3395 012352 012604      MOV      (SP)+,R4      ;RESTORE THE WORKING REG.'S
3396 012354 012603      MOV      (SP)+,R3
3397 012356 012602      MOV      (SP)+,R2
3398 012360 012601      MOV      (SP)+,R1
3399 012362 012600      MOV      (SP)+,R0

```

```

3400 012364 000002          RTI
3401
3402          ;SUBROUTINE ENTERED TO SAVE BUFFER POINTERS FOR UNIT '3'
3403
3404 012366 010037 024412  SAVBF3: MOV      R0, BUF3R0
3405 012372 010137 024422      MOV      R1, BUF3R1
3406 012376 012604          MOV      (SP)+, R4          ;RESTORE THE WORKING REG.'S
3407 012400 012603          MOV      (SP)+, R3
3408 012402 012602          MOV      (SP)+, R2
3409 012404 012601          MOV      (SP)+, R1
3410 012406 012600          MOV      (SP)+, R0
3411 012410 000002          RTI
3412
3413          ;SUBROUTINE ENTERED ON ENCOUNTERING SYSTEM ERRORS WHICH MOVES
3414          ;THE CHARACTER IN THE PC INTO THE DISPLAY BUFFER.
3415
3416 012412 005262 024334  FLGERR: INC      ERFLGO(R2)          ;SET ERROR FLAG
3417 012416 112720 000362      MOVB     #BLINK, (R0)+          ;SET UP TO BLINK CHAR.
3418 012422 117620 000000      MOVB     @ (SP), (R0)+          ;MOVE CHAR. INTO BUFFER
3419 012426 112720 000360      MOVB     #360, (R0)+          ;CLR PRESET FIELD
3420 012432 112710 000031      MOVB     #EOS, (R0)           ;TERMINATE W/ EOS.
3421 012436 062716 000002      ADD      #2, (SP)            ;SET UP STACK TO EXIT
3422 012442 000207          RTS      PC
3423
3424          ;FK INTERRUPT SERVICE ROUTINE
3425
3426 012444 105773 001334  SERVFK: TSTB     @FKOCSR(R3)          ;ASCII FLAG SET?
3427 012450 100421          BMI      SERVOD              ;YES, SERVICE KEYBOARD
3428 012452 005762 024274      TST      AUTSWO(R2)          ;IS THE AUTO CHAR. SW. SET?
3429 012456 001012          BNE      AUTOLL              ;YES, GENERATE A CHARACTER
3430 012460 005762 024250      TST      PATERN(R2)          ;GENERATING WORST CASE PATTERN?
3431 012464 001011          BNE      JMPWOR              ;YES, OUTPUT NEXT CHAR.
3432 012466 005762 024240      TST      HOLDSO(R2)          ;INTERVAL TIMER, IS ERROR HOLD SET?
3433 012472 001402          BEQ      LINKOD              ;NO, CLR INTERVAL TIME FLAG
3434 012474 004737 013760      JSR      PC, TESTSW          ;YES, TST SWIS TO CONTINUE
3435 012500 000137 013106      LINKOD: JMP      SERVOD              ;SWIS STILL SET, EXIT
3436 012504 000137 013304      AUTOLL: JMP      AUTOUT
3437 012510 000137 013350      JMPWOR: JMP      WORST1
3438
3439 012514 005773 001334  SERVOD: TST      @FKOCSR(R3)          ;ERROR FLAG SET?
3440 012520 100003          BPL      SERVOD              ;NO VALID CHAR.
3441 012522 004737 012412      JSR      PC, FLGERR          ;YES, ASCII ERRR FLAG SET
3442 012526 000113          IIR      113                ;DISPLAY A 'BLINKING K'
3443 012530 117304 001336      SERVOD: MOVB     @FKODAT(R3), R4    ;READ & SAVE CHAR.
3444 012534 122704 000003      CMPB     #3, R4              ;CHAR='↑C'?
3445 012540 001051          BNE      SERVAA              ;NO, CONTINUE
3446 012542 005762 024374      TST      BUSYO(R2)           ;YES, DL'S BUSY?
3447 012546 001403          BEQ      RELOAD              ;NO, RE-START
3448 012550 005262 024324      INC      STOPO(R2)           ;YES, SET STOP SW.
3449 012554 000554          BR       SERVOD              ;EXIT
3450 012556 013700 024424      RELOAD: MOV      START, R0        ;RESET BUFFER POINTER
3451 012562 112710 000031      MOVB     #EOS, (R0)          ;SET UP FOR NEW INPUT
3452 012566 005062 024364      CLR      XFERO(R2)
3453 012572 005062 024354      CLR      RESTRO(R2)
3454 012576 005062 024324      CLR      STOPO(R2)
3455 012602 005062 024220      CLR      STCODO(R2)

```

3456	012606	005062	024374	CLR	BUSY0(R2)	
3457	012612	005062	024262	CLR	BLKERR(R2)	
3458	012616	005062	024250	CLR	PATERN(R2)	
3459	012622	005062	024274	CLR	AUTSW0(R2)	
3460	012626	005062	024304	CLR	TRANF0(R2)	
3461	012632	005062	024314	CLR	RECTRO(R2)	
3462	012636	005062	024334	CLR	ERFLG0(R2)	
3463	012642	005062	024240	CLR	HOLDS0(R2)	
3464	012646	005062	024230	CLR	CHRCT0(R2)	; CLR CHAR. CNTR.
3465	012652	005062	024200	CLR	TRNSW0(R2)	
3466	012656	005062	024170	CLR	RECSW0(R2)	
3467	012662	000511		BR	SERV0D	; EXIT
3468	012664	122704	000024	SERVAA: CMPB	#24, R4	; CHAR='↑T' ?
3469	012670	001017		BNE	SERV0B	; NO, CONTINUE
3470	012672	005162	024364	COM	XFER0(R2)	; YES, COMPLIMENT TRANSFER SW.
3471	012676	001503		BEQ	SERV0D	; STOP AUTO TRANSFER.
3472	012700	005762	024374	TST	BUSY0(R2)	; CURRENTLY TRANSFERRING DATA?
3473	012704	001100		BNE	SERV0D	; YES, IGNORE SW.
3474	012706	005762	024354	SERVAB: TST	RESTRO(R2)	; HAS A RESTART ADDR BEEN LOADED?
3475	012712	001403		BEQ	.+10	; NO, TERMINATE INPUT BUFFER
3476	012714	016200	024354	MOV	RESTRO(R2), R0	; YES, RESET BUFFER POINTER
3477	012720	000423		BR	SERVBB	; SET UP TO START TRANSFER
3478	012722	112720	000014	MOVB	#EOP, (R0)+	; TERMINATE MESSAGE
3479	012726	000420		BR	SERVBB	
3480						
3481	012730	122704	000001	SERV0B: CMPB	#1, R4	; CHAR. = TO '↑A'
3482	012734	001556		BEQ	AUT0ST	; YES, AUTOMATICALLY GENERATE CHAR.'S
3483	012736	122704	000005	CMPB	#5, R4	; CHAR. = TO '↑E'
3484	012742	001705		BEQ	RELOAD	; YES, RELOAD
3485	012744	122704	000027	CMPB	#27, R4	; CHAR. = TO '↑W' ?
3486	012750	001574		BEQ	WORST	; YES, GENERATE WORST CASE PATTERN
3487	012752	005762	024374	TST	BUSY0(R2)	; DL11 BUSY?
3488	012756	001053		BNE	SERV0D	; YES, IGNORE CHAR.
3489	012760	122704	000033	CMPB	#33, R4	; REQUEST TRANSFER?
3490	012764	001024		BNE	SERV0C	; NO
3491	012766	000747		BR	SERVAB	
3492	012770	112710	000031	SERVBB: MOVB	#EOS, (R0)	
3493	012774	010062	024354	MOV	R0, RESTRO(R2)	; SAVE LAST ADDR. FOR AUTO RESTART
3494	013000	012762	000001 024374	MOV	#1, BUSY0(R2)	; SET BUSY SW.
3495	013006	012762	000001 024200	MOV	#1, TRNSW0(R2)	; SET TRANSFER, SOFTWARE SW.
3496	013014	013701	024424	MOV	START, R1	; SET UP BUFFER POINTER TO TRANSMIT DATA.
3497	013020	105741		TSTB	-(R1)	; SET UP BUFFER POINTER
3498	013022	012762	000004 024210	MOV	#4, NULCT0(R2)	
3499	013030	105072	001570	SERVBC: CLRB	QXBUFO(R2)	; TRANSMIT THE START CODE
3500	013034	000424		BR	SERV0D	; EXIT
3501	013036	122704	000177	SERV0C: CMPB	#177, R4	; CHAR. = 'RUBOUT' ?
3502	013042	001010		BNE	SERVCC	; NO, SAVE IT
3503	013044	005762	024230	TST	CHRCT0(R2)	; BUFFER EMPTY?
3504	013050	001416		BEQ	SERV0D	; YES, IGNORE IT
3505	013052	005362	024230	DEC	CHRCT0(R2)	; NO, DECEMENT COUNTER
3506	013056	112740	000031	MOVB	#EOS, -(R0)	; BACK UP BUFFER POINTER
3507	013062	000411		BR	SERV0D	; EXIT
3508	013064	026227	024230 000600	SERVCC: CMP	CHRCT0(R2), #384.	; BUFFER FULL?
3509	013072	103011		BHIS	SERV0X	; YES, IGNORE CHAR.
3510	013074	005262	024230	INC	CHRCT0(R2)	; INCREMENT CHAR. CNTR.
3511	013100	110420		MOVB	R4, (R0)+	; NO, SAVE CHAR.


```

3512 013102 112710 000031          MOVB   #EOS,(R0)
3513 013106 052773 000002 001334  SERVOD: BIS   #2,DFKCSR(R3)      ;CLR INTERVAL TIMER FLAG
3514 013114 000207          RTS     PC                ;EXIT
3515 013116 005062 024250          SERVDX: CLR  PATERN(R2)    ;CLEAR WORST CASE PATTERN SW.
3516 013122 005062 024274          CLR   AUTSWO(R2)        ;CLEAR AUTO SW.
3517 013126 000767          BR     SERVOD
3518
3519          ;DL11 TRANSMITTER SERVICE ROUTINE
3520
3521 013130 005762 024344          TRNSMT: TST  INTRNO(R2)   ;LEGAL INTERRUPT?
3522 013134 001003          BNE   .+10              ;YES, SERVICE IT.
3523 013136 005262 024344          INC   INTRNO(R2)       ;THIS IS THE INITIALIZE INTERRUPT.
3524 013142 000207          RTS     PC                ;EXIT
3525 013144 005762 024200          TST  TRNSWO(R2)        ;TRANSFER SW. SET?
3526 013150 001004          BNE   TRANOA           ;YES, LEGAL INTERRUPT
3527 013152 004737 012412          JSR   PC,FLGERR        ;NO, ILLEGAL TRANSMITTER INTERRUPT
3528 013156 000124          124
3529 013160 000207          RTS     PC                ;DISPLAY A 'BLINKING T'.
3530 013162 017737 166136 013270  TRANOA: MOV  @SWR,R3PRT   ;READ THE SWITCHES
3531 013170 042737 177774 013270  BIC   #177774,R3PRT
3532 013176 022737 000003 013270  CMP  #SWO0+SWO1,R3PRT  ;LOOPING DATA?
3533 013204 001430          BEQ   TRANOC+4         ;YES, EXIT
3534 013206 005362 024210          DEC  NULCTO(R2)
3535 013212 003306          BGT  SERVBC           ;XMITTED 4 NULL CHARS?
3536 013214 002406          BLT  TRANOB
3537 013216 112772 000377 001570  MOVB  #377,@XBUFO(R2)
3538 013224 005062 024314          CLR  RECTRO(R2)       ;RECV'D CHARACTER COUNTER
3539 013230 000207          RTS     PC
3540 013232 122721 000014          TRANOB: CMPB #EOP,(R1)+ ;HAS LAST CHAR. BEEN TRANSMITTED?
3541 013236 001011          BNE   TRANOC           ;NO, TRANSMIT NEXT CHAR.
3542 013240 005062 024200          CLR  TRNSWO(R2)       ;YES, CLR TRANSMIT SW.
3543 013244 005062 024262          CLR  BLKERR(R2)       ;CLEAR BLOCK ERROR SW.
3544 013250 005262 024170          INC  RECSWO(R2)       ;SET UP TO REC. DATA
3545 013254 013701 024424          MOV  START,R1        ;RESET POINTER TO COMPARE DATA
3546 013260 000207          RTS     PC
3547 013262 111172 001570          TRANOC: MOVB (R1),@XBUFO(R2) ;EXIT
3548 013266 000207          RTS     PC                ;TRANSMIT NEXT CHAR.
3549 013270 000000          R3PRT: .WORD 0
3550
3551          ;ENTERED HERE TO INITIALIZE THE AUTOMATIC CHARACTER GENERATOR
3552
3553 013272 012704 000041          AUTOST: MOV  #41,R4
3554 013276 010462 024274          MOV  R4,AUTSWO(R2)
3555 013302 000670          BR     SERVCC
3556
3557          ;ENTERED HERE TO GENERATE AND OUTPUT THE AUTOMATIC CHARACTERS
3558
3559 013304 005262 024274          AUTOUT: INC  AUTSWO(R2)   ;UPDATE THE CHARACTER
3560 013310 042762 177600 024274  BIC  #177600,AUTSWO(R2)
3561 013316 032762 000140 024274  BIT  #140,AUTSWO(R2)   ;IS CHARACTER LEGAL?
3562 013324 001003          BNE   .+10              ;YES
3563 013326 052762 000040 024274  BIS  #40,AUTSWO(R2)    ;NO, RESET IT
3564 013334 016204 024274          MOV  AUTSWO(R2),R4
3565 013340 000651          BR     SERVCC
3566
3567          ;ENTERED HERE TO INITIALIZE AND OUTPUT A WORST CASE CHARACTER

```

```

3568 ;PATTERN OF '125 & 252'.
3569
3570 013342 112762 000252 024250 WORST:  MOVB  #252,PATERN(R2)
3571 013350 105162 024250 WORST1: COMB  PATERN(R2)
3572 013354 116204 024250          MOVB  PATERN(R2),R4
3573 013360 000641          BR    SERVCC
3574
3575 ;DL11 RECEIVER SERVICE ROUTINE
3576
3577 013362 017204 001530 RECVID: MOV  @RBUF0(R2),R4
3578 013366 013705 024424          MOV  START,R5
3579 013372 062705 001604          ADD  #900.,R5
3580 013376 020500          CMP  R5,R0
3581 013400 003021          BGT  RECVON
3582 013402 017737 165716 014020          MOV  @SWR,R3PXT
3583 013410 042737 177774 014020          BIC  #177774,R3PXT
3584 013416 022737 000003 014020          CMP  #SW00+SW01,R3PXT
3585 013424 001003          BNE  1$
3586 013426 013700 024424          MOV  START,R0
3587 013432 000404          BR   RECVON
3588 013434 122704 000014          1$:  CMPB #EOP,R4
3589 013440 001522          BEQ  RECV0D
3590 013442 000207          RTS  PC
3591 013444 005762 024170          RECVON: TST  RECSW0(R2)
3592 013450 001010          BNE  RECV0A
3593 013452 032777 000001 165644          BIT  #SW00,@SWR
3594 013460 001004          BNE  RECV0A
3595 013462 004737 012412          STERR: JSR  PC,FLGERR
3596 013466 000122          122
3597 013470 000207          RTS  PC
3598
3599 013472 005704          RECV0A: TST  R4
3600 013474 100021          BPL  RECV0B
3601 013476 032704 040000          BIT  #40000,R4
3602 013502 001404          BEQ  .+12
3603 013504 004737 012412          JSR  PC,FLGERR
3604 013510 000117          117
3605 013512 000412          BR   RECV0B
3606 013514 032704 020000          BIT  #20000,R4
3607 013520 001404          BEQ  .+12
3608 013522 004737 012412          JSR  PC,FLGERR
3609 013526 000106          106
3610 013530 000403          BR   RECV0B
3611 013532 004737 012412          JSR  PC,FLGERR
3612 013536 000120          120
3613 013540 017737 165560 014020          RECV0B: MOV  @SWR,R3PXT
3614 013546 042737 177774 014020          BIC  #177774,R3PXT
3615 013554 022737 000003 014020          CMP  #SW00+SW01,R3PXT
3616 013562 001524          BEQ  RECVAC
3617 013564 105704          TSTB R4
3618 013566 001513          BEQ  RECV0E
3619 013570 005762 024220          TST  STCOD0(R2)
3620 013574 001014          BNE  RECVBB
3621 013576 122704 000377          CMPB #377,R4
3622 013602 001107          BNE  RECVAB
3623 013604 112720 000136          MOVB #136,(R0)+

```

:COMPLIMENT PATTERN

:READ DATA BUFFER

:IS BUFFER LIMIT EXCEEDED?
:IS BUFFER LIMIT EXCEEDED?
:NO, PROCESS CHARACTER
:IF SW0 AND SW1 ARE SET -
:DO NOT EXIT- RESET THE BUFFER
:AND CONTINUE.

:TRANSFER COMPLETE?
:YES, SET UP TO EXIT
:NO, IGNORE THE CHARACTER
:REC, SW, SET?
:YES, LEGAL REC. INTERRUPT
:IS SW '0' SET?
:YES, HOST SENDING DATA
:NO, ILLEGAL REC. INTR.
:DISPLAY A 'BLINKING R'.
:EXIT

:REC. ERROR BIT SET?
:NO, VALID CHAR.
:OVERRUN ERROR?
:NO
:YES, DATA OVERRUN ERROR
:DISPLAY A 'BLINKING O'
:FRAMING ERROR?
:NO
:YES, DISPLAY A 'BLINKING F'

:PARITY ERROR
:DISPLAY A BLINKING 'P'
:READ THE SWITCHES

:LOOP DATA BACK REQUESTED
:YES
:= NULL CHAR.?
:YES, IGNORE IT
:HAVE WE RECEIVED THE START CODE?
:YES, SAVE DATA
:NO, IS THIS IT?
:NO, CHECK 'SW0'
:YES, DISPLAY ST. CODE AS 't'

```

3624 013610 112720 000012          MOVB   #12,(R0)+      ;ST NEW LINE
3625 013614 112710 000031          MOVB   #EOS,(R0)     ;TERMINATE BUFFER.
3626 013620 005262 024220          INC    STCOD0(R2)    ;SET SW.
3627 013624 000474                   BR     RECVOE        ;EXIT
3628
3629 013626 122704 000014          RECVBB: CMPB   #EOP,R4      ;TRANSFER DONE?
3630 013632 001425                   BEQ    RECVDD        ;YES, SET UP TO EXIT
3631 013634 005262 024314          INC    RECTRO(R2)   ;KEEP TRACK OF NO. OF REC'D CHAR.S'S
3632 013640 120421                   CMPB   R4,(R1)+     ;COMPARE REC. DATA TO TRANS. DATA
3633 013642 001415                   BEQ    RECVOC        ;DATA OK
3634 013644 005762 024262          RECVBC: TST    BLKERR(R2) ;HAS A BLOCK ERROR BEEN DETECTED?
3635 013650 001002                   BNE    1$           ;YES, DON'T UPDATE ERROR COUNTER
3636 013652 005262 024334          INC    ERFLGO(R2)   ;DATA ERROR SET SOFT SWITCH
3637 013656 005262 024262          1$:    INC    BLKERR(R2)
3638 013662 112720 000374          MOVB   #374,(R0)+   ;DISPLAY CHAR IN "BOLD-UNDERLINE"
3639 013666 110420                   MOVB   R4,(R0)+
3640 013670 112720 000360          MOVB   #360,(R0)+   ;CLR FIELD PRESET
3641 013674 000401                   BR     .+4
3642 013676 110420          RECVOC: MOVB   R4,(R0)+   ;DISPLAY REC. CHAR.
3643 013700 112710 000031          MOVB   #EOS,(R0)
3644 013704 000207                   RTS    PC           ;EXIT
3645 013706 005262 024304          RECVDD: INC    TRANFO(R2) ;TRANSFER COUNTER
3646 013712 005062 024374          CLR    BUSY0(R2)
3647 013716 005062 024170          CLR    RECSW0(R2)   ;CLR. REC. SOFTWARE SW.
3648 013722 005062 024220          CLR    STCOD0(R2)   ;CLR ST. CODE SW.
3649 013726 005762 024364          TST    XFER0(R2)    ;IS AUTO TRANSFER SW. SET?
3650 013732 001431                   BEQ    RECVOE        ;NO, NORMAL TRANSFER EXIT
3651 013734 005762 024324          TST    STOPO(R2)    ;YES, STOP SW. SET?
3652 013740 001402                   BEQ    .+6           ;NO, START NEXT TRANSFER
3653 013742 000137 012556          JMP    RELOAD        ;YES, RESTART
3654 013746 016200 024354          MOV    RESTRO(R2),R0 ;NO, RESET BUFFER POINTER
3655 013752 005762 024334          TST    ERFLGO(R2)   ;ERROR FLAG SET?
3656 013756 001412                   BEQ    RECVDD        ;NO, START NEXT TRANSFER
3657 013760 005777 165340          TESTSW: TST    @SWR    ;YES, IS SW15 SET?
3658 013764 100411                   BMI    RECVEE        ;YES, INHIBIT FURTHER TRANSFERS
3659 013766 005762 024240          TST    HOLDS0(R2)   ;HAS SW15 BEEN SET AND NOW RESET?
3660 013772 001404                   BEQ    RECVDD        ;NO, IT HASN'T BEEN SET
3661 013774 005062 024334          CLR    ERFLGO(R2)   ;YES, RESET THE ERROR COUNTER TO '0'
3662 014000 005062 024240          CLR    HOLDS0(R2)
3663 014004 000137 012770          RECVDD: JMP    SERVBB   ;SET UP NEXT TRANSFER
3664 014010 012762 000001 024240  RECVEE: MOV    #1,HOLDS0(R2) ;SET TO HOLD XFER
3665 014016 000207                   RECVOE: RTS    PC
3666 014020 000000          R3PXT: .WORD    0
3667
3668
3669 014022 032777 000001 165274  RECVAB: BIT    #SW00,@SWR   ;IS SW '0' SET?
3670 014030 001322                   BNE    RECVOC        ;YES, JUST DISPLAY CHARACTER.
3671 014032 000704                   BR     RECVBC        ;NO, POST AS ILLEGAL CHAR.
3672
3673          ;ENTERED HERE IF BOTH 'SW00 & SW01' ARE SET
3674
3675 014034 005262 024314          RECVAC: INC    RECTRO(R2) ;UPDATE RECEIVE COUNT
3676 014040 005262 024304          INC    TRANFO(R2)   ;UPDATE TRANSMIT COUNT
3677 014044 005262 024200          INC    TRNSW0(R2)   ;SET THE TRANSMITTER SWITCH
3678 014050 110472 001570          MOVB   R4,@XBUFO(R2) ;TRANSMIT CHARACTER BACK
3679 014054 042704 177400          BIC    #177400,R4   ;SET UP TO DISPLAY OCTAL REPRESENTATION

```

```
3680 ;OF THE CHARACTER
3681 014060 012737 000003 024470 MOV #3,KSTOR5
3682 014066 012737 000376 015166 MOV #376,MASK
3683 014074 012703 000006 MOV #6,R3
3684 014100 000401 BR .+4
3685 014102 006104 MOVE1: ROL R4
3686 014104 006104 ROL R4
3687 014106 006104 ROL R4
3688 014110 005337 024470 DEC KSTOR5
3689 014114 002010 BGE MOVE2
3690 014116 110402 MOVB R4,R2
3691 014120 143702 015166 BICB MASK,R2
3692 014124 152702 000060 BISB #60,R2
3693 014130 110220 MOVB R2,(R0)+ ;PUT CHARACTER IN DISPLAY BUFFER
3694 014132 112710 000031 MOVB #E05,(R0) ;TERMINATE BUFFER
3695 014136 012737 000370 015166 MOVE2: MOV #370,MASK
3696 014144 005303 DEC R3 ;DONE
3697 014146 001355 BNE MOVE1 ;NO
3698 014150 112720 000040 MOVB #40,(R0)+ ;YES, PLACE 'SPACE' CODE IN BUFFER
3699 014154 112710 000031 MOVB #E05,(R0)
3700 014160 000207 RTS PC ;RETURN
3701 ;*****
3702 ;ROUTINE TO LOOP THRU A SINGLE LOGIC SUBTEST. ENTERED FROM THE 'MONITOR'
3703 ;VIA SELECTING TEST '?'. UPON ENTERING THIS SUBROUTINE ANOTHER 'HALT' IS EN-
3704 ;COUNTERED. LOAD THE ADDRESS OF THE SUBTEST TO BE EXECUTED IN THE DATA
3705 ;SWITCHES AND PRESS CONTINUE.
3706 ;NOTE THAT 'SW11' MUST BE '0' (DOWN) TO RUN THIS TEST.
3707 ;*****
3708
3709 014162 005737 024154 TESTX: TST TTYSWH ;TTY AVAILABLE?
3710 014166 001406 BEQ TSTA ;NO, HALT FOR TEST ADDR.
3711 014170 104012 PRINT
3712 014172 021634 MESB ;TEXT 'TEST ADDR.?'
3713 014174 104015 ASEMBL ;YES, GET ADDR. FROM TTY.
3714 014176 010337 024460 MOV R3,KSTOR1 ;SAVE ADDRESS
3715 014202 000404 BR TSTB ;EXECUTE TEST
3716 014204 000000 TSTA: HALT ;GET TEST ADDRESS
3717 014206 017737 165112 024460 MOV @SWR,KSTOR1 ;SAVE ADDRESS
3718
3719 014214 023727 024460 006174 TSTB: CMP KSTOR1,#SETUPF ;'FK' OR 'VT' LOGIC TEST?
3720 014222 101003 BHI .+10 ;SET UP FOR 'FK'
3721 014224 004737 003026 JSR PC,SETUPV ;SET UP FOR 'VT'
3722 014230 000402 BR .+6
3723 014232 004737 006174 JSR PC,SETUPF
3724 014236 062737 000002 024460 ADD #2,KSTOR1 ;ADD '2' TO POINT TO INSTRUCTION AFTER SCOPE
3725 014244 017737 010210 024142 MOV @KSTOR1,TSTNUM ;LOAD TEST NO.
3726 014252 062737 000002 024460 ADD #2,KSTOR1
3727 014260 005037 020404 XLOOP: CLR SCOPEF ;KEEP COUNT AT ZERO
3728 014264 012737 014260 020406 MOV #XLOOP,RETURN ;LOAD SCOPE LOOP RETURN POINTER
3729 014272 000177 010162 JMP @KSTOR1 ;JUMP TO TEST
3730 ;*****
3731 ;SUBROUTINE TO ISSUE N SPACES
3732 ;N IS ONE PLUS VALUE CONTAINED IN SPACEX
3733 ;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
3734 ;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE
3735 ;*****
```

```

3736
3737 014276 105777 165016 XSPACE: TSTB @TPS ;WAIT FOR TTY READY
3738 014302 100375 BPL .-4
3739 014304 012777 000240 165010 MOV #240,@TPB ;OUTPUT A SPACE
3740 014312 005337 014326 DEC SPACEX ;DECREMENT COUNT
3741 014316 003367 BGT XSPACE ;LOOP IF NOT DONE
3742 014320 005037 014326 CLR SPACEX ;RESET COUNT TO ZERO
3743 014324 000002 RTI ;RETURN
3744 014326 000000 SPACEX: 0
3745 ;*****
3746 ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET
3747 ;*****
3748
3749 014330 005737 024154 TKSFLG: TST TTYSWH ;USING THE 'TTY' ?
3750 014334 001404 BEQ .+12 ;NO, EXIT
3751 014336 105777 164752 TSTB @TKS ;FLAG SET?
3752 014342 100001 BPL .+4 ;NO, EXIT
3753 014344 104013 TTYIN ;YES, INQUIRE
3754 014346 000002 RTI
3755 ;*****
3756 ;KEYBOARD SERVICE ROUTINE
3757 ;*****
3758 014350 012704 014600 XTTYIN: MOV #INBUF,R4 ;SETUP CHARACTER BUFFER
3759 014354 005037 024146 CLR CHRCNT ;CLEAR CHARACTER COUNTER
3760 014360 005037 014602 CLR INBUF+2
3761 014364 105777 164724 INPUTA: TSTB @TKS ;CHARACTER READY?
3762 014370 100375 BPL INPUTA ;NO, WAIT IT OUT
3763 014372 017701 164720 MOV @TKB,R1 ;SAVE CHARACTER
3764 014376 042701 000200 BIC #200,R1 ;STRIPE PARITY BIT
3765 014402 120127 000060 CMPB R1,#60 ;IS IT A SPECIAL CHARACTER
3766 014406 100420 BMI SPCHR1 ;YES, TEST IT
3767 014410 122701 000137 CMPB #137,R1
3768 014414 100415 BMI SPCHR1
3769 014416 010124 INPUTB: MOV R1,(R4)+ ;SAVE CHARACTER
3770 014420 005237 024146 INC CHRCNT ;INCREMENT THE CHARACTER COUNT.
3771 014424 022737 000016 024146 CMP #14,CHRCNT
3772 014432 100450 BMI SPCHR5 ;TYPE '?' IF TOO MANT CHAR.
3773 014434 105777 164660 OUTPTA: TSTB @TPS ;ECHO CHARACTER
3774 014440 100375 BPL OUTPTA
3775 014442 110177 164654 MOVB R1,@TPB
3776 014446 000746 BR INPUTA ;WAIT FOR NEXT CHARACTER
3777
3778 ;SUBROUTINE TO TEST FOR SPECIAL CHARACTERS: '^C','+',^CR',',', OR ^RUBOUT^
3779 014450 122701 000003 SPCHR1: CMPB #3,R1 ;CHAR.= '^C'
3780 014454 001002 BNE .+6 ;NO, NOT '^C'
3781 014456 000137 002526 JMP MONITR ;YES, EXIT TO MONITOR
3782 014462 122701 000001 CMPB #1,R1 ;CHAR.= '^A'?
3783 014466 001005 BNE .+14 ;NOT '^A'
3784 014470 022626 POP2SP ;YES, RESTORE STACK
3785 014472 104012 PRINT
3786 014474 023001 CNTRLA ;TEXT '^A'
3787 014476 000177 007426 JMP @AVECTR
3788 014502 122701 000177 CMPB #177,R1 ;CHAR.= ^RUBOUT^
3789 014506 001011 BNE SPCHR3 ;IGNORE CHAR. & EXIT
3790 014510 005737 024146 TST CHRCNT ;IS RUBOUT LEGAL?
3791 014514 001723 BEQ INPUTA ;NO, IGNORE IT

```

```

3792 014516 005337 024146
3793 014522 012701 000134
3794 014526 005744
3795 014530 000741
3796 014532 122701 000054
3797 014536 001727
3798 014540 122701 000015
3799 014544 001003
3800 014546 104012
3801 014550 023005
3802 014552 000002
3803 014554 122701 000040
3804 014560 001725
3805 014562 105701
3806 014564 001002
3807 014566 000137 002526
3808 014572 104012
3809 014574 023007
3810 014576 000664
3811 014600 000000
3812 014632
3813
3814
3815
3816
3817
3818 014632 005037 024162
3819 014636 005737 024164
3820 014642 001001
3821 014644 000207
3822 014646 005077 164636
3823 014652 005237 024134
3824 014656 004737 016320
3825 014662 005737 024134
3826 014666 001414
3827 014670 005726
3828 014672 104012
3829 014674 023731
3830 014676 112737 000260 014750
3831 014704 153737 024134 014750
3832 014712 104012
3833 014714 014750
3834 014716 000002
3835 014720 104012
3836 014722 023711
3837 014724 005737 024240
3838 014730 001406
3839 014732 032777 010000 164364
3840 014740 001002
3841 014742 005726
3842 014744 000002
3843 014746 000207
3844 014750 000 015 012
3845 014753 100
3846
3847

```

```

DEC CHRCNT
MOV #134,R1 ;TYPE '\ ' TO INDICATE RUBOUT
TST -(R4) ;POP OFF LAST CHARACTER
BR OUTPTA ;WAIT FOR NEXT CHARACTER
SPCHR3: CMPB #54,R1 ;TEST FOR '
BEQ INPUTB ;LEGAL CHAR. SAVE IT
SPCHR4: CMPB #15,R1 ;=TO 'CARRIAGE RETURN' TO TERMINATE?
BNE SPCHRS ;NO, CONTINUE
PRINT ;YES, TYPE 'CR-LF'
CRLF
EXTTY: RTI ;EXIT
SPCHR5: CMPB #40,R1 ;TEST FOR SPACE
BEQ OUTPTA ;ECHO BUT DON'T SAVE
TSTB R1 ;TEST FOR A NULL CHAR.
BNE .+6 ;NO, ECHO IT
JMP MONITR ;ILLEGAL CHAR. RETURN TO MONITR
PRINT ;OTHERWISE TYPE '?'
QMARK
BR XTTYIN ;WAIT FOR NEW ENTRY
INBUF: 0 ;CHARACTER STORAGE BUFFER
.=.+30
;*****
;THIS ROUTINE CHECKS THE 'SWAP SW.' AND IF FOUND SET, SETS UP
;THE OPPOSITE UNIT TO BE TESTED.
;*****
TSWAP: CLR DONESW
TST SWAPEM ;SWAP SW. SET?
BNE .+4 ;YES, SWITCH UNITS
RTS PC ;NO, EXIT
CLR @VTCSR ;CLR GO BIT
INC UNITFG ;SELECT OPPOSITE UNIT
JSR PC,STUNIT ;SET IT UP
TST UNITFG ;BACK TO '0'
BEQ SWPEXT ;YES, CONTINUE TO NEXT TEST
POP1SP ;CLEAN UP STACK
PRINT MESS4 ;TEXT 'RUNNING UNIT 'X'
MOVB #260, MRUNN1
BISB UNITFG, MRUNN1
PRINT MRUNN1
RTI ;RE-RUN TEST W/ UNIT 'X'
SWPEXT: PRINT MESS4 ;TEXT 'RUNNING UNIT '0'
TST HOLDSO
BEQ .+16
BIT #SW12,@SWR
BNE .+6
POP1SP
RTS PC
MRUNN1: .BYTE 0,15,12,100
;*****

```

```

3848
3849
3850
3851
3852
3853 014754 004737 017062
3854 014760 104002
3855 014762 017602 000000
3856 014766 062716 000002
3857 014772 105777 164322
3858 014776 100375
3859 015000 122712 000100
3860 015004 001003
3861 015006 104003
3862 015010 104022
3863 015012 000002
3864 015014 122712 000045
3865 015020 001403
3866 015022 112277 164274
3867 015026 000761
3868 015030 012777 000015 164264
3869 015036 105777 164256
3870 015042 100375
3871 015044 012777 000012 164250
3872 015052 105722
3873 015054 000746
3874
3875
3876
3877
3878
3879 015056 004737 017062
3880 015062 104002
3881 015064 017601 000000
3882 015070 062716 000002
3883 015074 012737 000006 024466
3884 015102 012737 000376 015166
3885 015110 000401
3886 015112 006111
3887 015114 006111
3888 015116 006111
3889 015120 111102
3890 015122 143702 015166
3891 015126 052702 000260
3892 015132 132777 000200 164160
3893 015140 100374
3894 015142 110277 164154
3895 015146 012737 000370 015166
3896 015154 005337 024466
3897 015160 001354
3898 015162 104003
3899 015164 000002
3900 015166 000376
3901
3902
3903

```

```

;MESSAGE PRINT ROUTINE, ENTERED VIA EMT DISPATCH HANDLER.
;ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
;THE ADDRESS OF MESSAGE TO BE TYPED.
;*****
TYPMES: JSR      PC,TTYENB      ;ENABLE TTY INTERRUPTS
        SAVREG      ;SAVE REGISTERS ON STACK
        MOV      2(SP),R2      ;GET THE MESSAGE ADDRESS FROM START
        ADD      #2,(SP)      ;SET UP STACK TO EXIT
TYPERA: TSTB      2TPS
        BPL      TYPERA      ;WAIT FOR TTY DONE
        CMPB     #100,(R2)    ;TEST FOR 'a'
        BNE     TYPER1      ;BRANCH IF NO EQUAL
        GETREG     ;RESTORE REGISTERS FROM STACK.
        NULL     ;TRANSMIT 'NULL' CHARACTER
        RTI     ;OTHERWISE EXIT
TYPER1: CMPB     #45,(R2)    ;TEST FOR '%'
        BEQ     TYPECL      ;IF = TYPE 'CR-LF'
TYPER2: MOVB     (R2)+,2TPB  ;OUTPUT CHAR.
        BR      TYPERA
TYPECL: MOV      #15,2TPB   ;TYPE 'CR'
        TSTB     2TPS
        BPL     -4
        MOV     #12,2TPB
        TSTB     (R2)+      ;INCREMENT BUFFER
        BR      TYPERA
;*****
;SUBROUTINE TO TYPEOUT A '6' DIGIT OCTAL NO. THE 'PC' CONTAINS
;THE ADDRESS OF 'WORD' TO BE TYPED
;*****
OCTPRT: JSR      PC,TTYENB      ;ENABLE TTY INTERRUPTS
        SAVREG      ;SAVE REGISTERS ON STACK
        MOV      2(SP),R1      ;THE ADDRESS OF WORD TO BE TYPED
        ADD      #2,(SP)      ;SET UP STACK TO EXIT
        MOV      #6,KSTOR4
        MOV      #376,MASK    ;MASK FOR FIRST BIT
        BR      +4
MOVEIT: ROL      (R1)
        ROL      (R1)
        ROL      (R1)
        MOVB     (R1),R2
        BICB     MASK,R2
        BIS      #260,R2
        BITB     #200,2TPS
        BPL     -6
        MOVB     R2,2TPB      ;PRINT CHAR.
        MOV      #370,MASK    ;MASK FOR NEXT '5' DIGITS
        DEC      KSTOR4
        BNE     MOVEIT
        GETREG     ;RESTORE REGISTERS FROM STACK.
        RTI
MASK:   376
;*****
;THIS SUBROUTINE IS USED TO CREATE THE 'VT' CHARACTER SET IN NUMERICAL ORDER
;ALL LINES ARE SET UP TO BE TERMINATED WITH EITHER 'VISIBLE END OF LINE'

```

```

3904
3905
3906
3907
3908
3909
3909
3910
3911 015170 005737 024152
3912 015174 001014
3913 015176 012700 024474
3914 015202 012701 000040
3915 015206 012702 177701
3916 015212 012703 177760
3917 015216 012704 015520
3918 015222 005237 024152
3919 015226 005202
3920 015230 001013
3921 015232 012702 177701
3922 015236 112420
3923 015240 062716 000002
3924 015244 005203
3925 015246 001401
3926 015250 000207
3927 015252 062716 000002
3928 015256 000207
3929 015260 005737 001622
3930 015264 001405
3931 015266 112720 000040
3932 015272 005037 001622
3933 015276 000401
3934 015300 110120
3935 015302 005201
3936 015304 022701 000335
3937 015310 001003
3938 015312 012701 000040
3939 015316 000745
3940 015320 005037 001622
3941 015324 005737 001620
3942 015330 001043
3943 015332 020127 000177
3944 015336 003440
3945 015340 120127 000203
3946 015344 001435
3947 015346 120127 000207
3948 015352 001432
3949 015354 120127 000211
3950 015360 001427
3951 015362 120127 000216
3952 015366 001424
3953 015370 120127 000210
3954 015374 001421
3955 015376 120127 000237
3956 015402 001416
3957 015404 120127 000213
3958 015410 001413
3959 015412 120127 000215

```

```

;OR 'VISIBLE END OF SCREEN'. THIS SUBROUTINE WILL RETURN TO THE SUBROUTINE
; 'CALL+2' AFTER CREATING '1' VT CHARACTER CODE AND INSERTING IT INTO
; THE 'VT' DISPLAY BUFFER. IF THIS WAS THE LAST CHARACTER OF THE LINE, THE
; SUBROUTINE RETURNS TO THE 'CALL+4', AND IF IT WAS THE LAST CHARACTER OF THE
; BUFFER IT RETURNS TO 'CALL+6'.
;*****
INTCHR: TST      INTSWH      ;INITIALIZE SUBROUTINE?
        BNE     LDCHAR      ;NO LOAD CHAR.
        MOV     #VTBUFF,RO  ;SET UP BUFFER ADDR.
        MOV     #40,R1      ;CHAR. CODES START W/40 (SPACE).
        MOV     #-63.,R2    ;LOAD '63' CHAR.'S/LINE
        MOV     #-16.,R3    ;LOAD '16' LINES OF CHAR.'S
        MOV     #TERMTB,R4  ;LINE TERMINATOR TABLE (EOL,EOP)
LDCHAR: INC     INTSWH
        INC     R2          ;INCREMENT CHAR. CNTR.
        BNE     LDCHR1     ;NOT END OF LINE
LDCHR0: MOV     #-63.,R2    ;RESET CHAR. CNTR.
        MOV     (R4)+,(R0)+ ;LOAD LINE TERMINATOR (EOL,EOP,EOS)
        ADD     #2,(SP)
        INC     R3          ;INCREMENT LINE NO. CNTR.
        BEQ     .+4        ;EOS?
        RTS     PC         ;NO, EOL-RETURN TO 'CALL+4'
        ADD     #2,(SP)    ;YES, RETURN TO 'CALL+6'
        RTS     PC
LDCHR1: TST     FCSET1
        BEQ     .+14
        MOV     #40,(R0)+
        CLR     FCSET1
        BR     1$
        MOV     R1,(R0)+
        INC     R1
1$: LDCHEB: CMP     #335,R1  ;MOVE CHAR. TO BUFFER
        BNE     .+10      ;INCREMENT CHAR. VALUE
        MOV     #40,R1    ;CHAR. VALUE OVERFLO?
        BR     LDCHR0     ;NO
        CLR     FCSET1   ;YES, RESET CHAR. VALUE
        TST     FCSET1   ;RESET LINE COUNTER
        BNE     2$
        CMP     R1,#177   ;FORIEGN CHARACTER SET
        BLE     2$       ;YES-IGNOR THIS SECTION
        CMP     R1,#203  ;COUNT ABOVE 177?
        BEQ     2$       ;NO,IGNOR THIS SECTION
        CMP     R1,#207  ;IF NO FORIEGN CHARACTER SET
        BEQ     2$       ;THEN THERE ARE SOME CODES
        CMP     R1,#211  ;THAT WE DON'T WANT TO PRINT
        BEQ     2$       ;THIS IS THE INTELLIGENTS SECTION
        CMP     R1,#216
        BEQ     2$
        CMP     R1,#210
        BEQ     2$
        CMP     R1,#237
        BEQ     2$
        CMP     R1,#213
        BEQ     2$
        CMP     R1,#215

```



```

3960 015416 001410
3961 015420 120127 000217
3962 015424 001405
3963 015426 120127 000255
3964 015432 001402
3965 015434 005137 001622
3966 015440 120127 000212
3967 015444 001407
3968 015446 120127 000214
3969 015452 001404
3970 015454 120127 000231
3971 015460 001401
3972 015462 000207
3973 015464 005201
3974 015466 005037 001622
3975 015472 000137 015304

```

```

                BEQ      2$
                CMPB    R1,#217
                BEQ      2$
                CMPB    R1,#255
                BEQ      2$
                COM      FCSET1
2$:             CMPB    R1,#VISEOL          ;CHAR.=VISIBLE END OF LINE?
                BEQ      LDCHR2           ;YES SKIP IT
                CMPB    R1,#VISEOP       ;CHAR.=VISIBLE END OF PARAGRAPH?
                BEQ      LDCHR2           ;YES SKIP IT
                CMPB    R1,#VISEOS       ;CHAR.=VISIBLE END OF SCREEN?
                BEQ      .+4              ;NO,
                RTS      PC
LDCHR2:        INC      R1                ;YES, SKIP IT
                CLR      FCSET1
                JMP      LDCHEB           ;RETURN TO 'CALL+2'

```

```

*****
;SUBROUTINE TO SET UP A WAIT LOOP TO WAIT FOR VERTICAL SYNC (17 MSEC.).
*****

```

```

3980
3981 015476 004737 017062
3982 015502 012737 170000 024442
3983 015510 005237 024442
3984 015514 001375
3985 015516 000002

```

```

XWAITS:        JSR      PC,TTYENB        ;ENABLE TTY INTERRUPTS
                MOV      #-10000,TEMP2
                INC      TEMP2
                BNE      .-4
                RTI

```

;FOLLOWING IS A '16' BYTE BUFFER USED TO TERMINATE EACH LINE OF THE 'VT' BUFFER.

```

3986
3987
3988
3989 015520      212
3990 015521      214
3991 015522      212
3992 015523      214
3993 015524      212
3994 015525      214
3995 015526      212
3996 015527      214
3997 015530      212
3998 015531      214
3999 015532      212
4000 015533      214
4001 015534      212
4002 015535      214
4003 015536      212
4004 015537      231

```

```

TERMTB:        .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .BYTE    VISEOL
                .BYTE    VISEOP
                .EVEN

```

```

*****
;SUBROUTINE TO SET UP AN APPROXIMATE '3' SECOND DELAY. THE ROUTINE ALSO
;CONTINUOUSLY MONITORS DATA 'SW14' AND IF FOUND SET CAUSES AN INDEFINITE DELAY.
*****

```

```

4005
4006
4007
4008
4009
4010
4011
4012 015540 012737 175000 024442
4013 015546 000402
4014 015550 005037 024442
4015 015554 004737 017062

```

```

XDELAY:        MOV      #175000,TEMP2    ;SET UP SHORT DELAY
                BR       .+6
XDLAYL:        CLR      TEMP2            ;SET UP LONG DELAY
                JSR      PC,TTYENB      ;ENABLE TTY INTERRUPTS

```

4016 015560 012737 177777 024440
 4017 015566 005737 024426
 4018 015572 001010
 4019 015574 032777 040000 163522
 4020 015602 001374
 4021 015604 032777 004000 163512
 4022 015612 001006
 4023 015614 005237 024442
 4024 015620 001362
 4025 015622 005237 024440
 4026 015626 001357
 4027 015630 000002
 4028
 4029
 4030
 4031
 4032 015632 104002
 4033 015634 012701 177700
 4034 015640 013702 024460
 4035 015644 112220
 4036 015646 005201
 4037 015650 001375
 4038 015652 104003
 4039 015654 000002
 4040
 4041
 4042
 4043
 4044
 4045 015656 104002
 4046 015660 012700 024474
 4047 015664 012701 004374
 4048 015670 112720 000031
 4049 015674 005301
 4050 015676 001374
 4051 015700 012700 024474
 4052 015704 005077 163574
 4053 015710 010077 163572
 4054 015714 012777 000001 163566
 4055 015722 104003
 4056 015724 000002
 4057
 4058
 4059
 4060
 4061
 4062
 4063 015726 005237 024162
 4064 015732 104020
 4065 015734 004737 017062
 4066 015740 005737 024150
 4067 015744 001015
 4068 015746 032777 010000 163350
 4069 015754 001011
 4070 015756 013737 024162 024240
 4071 015764 004737 014632

```

XDLAY1: MOV #-1,TEMP1
TST DISPSW ;DISPLAYING MESSAGE?
BNE XDLAY3 ;YES INHIBIT HOLD
BIT #SW14,@SWR ;TEST DATA SW14
BNE #-6 ;IF SET, WAIT FOR CLEAR.
BIT #SW11,@SWR ;IS SW. '11' SET?
BNE XDLAY2 ;YES, INIBIT DELAY
XDLAY3: INC TEMP2
BNE XDLAY1
INC TEMP1
BNE XDLAY1
XDLAY2: RTI

;*****
;SUBROUTINE TO LOAD '64' CHAR.'S INTO 'VT' BUFFER.
;*****

XLDLNE: SAVREG ;SAVE REG.'S
MOV #-64,R1 ;LOAD '64' CHAR. BYTES
MOV KSTOR1,R2 ;FROM THIS ADDR. UP.
XLDLNA: MOVB (R2)+,(R0)+ ;'R0' IS POINTER TO A 'VT' BUFFER ADDR.
INC R1 ;FINISHED?
BNE XLDLNA ;NO, LOAD NEXT CHAR.
GETREG ;YES, RESTORE REG.'S
RTI

;*****
;SUBROUTINE TO LOAD THE 'VT' BUFFER W/ 'VISIBLE EOS' CHARACTER.
;*****

XEOSBF: SAVREG ;SAVE REG.'S
MOV #VTBUFF,R0 ;SET UP BUFFER ADDR. POINTER
MOV #2300,R1 ;LOAD '515' DATA WORDS
MOVB #EOS,(R0)+
DEC R1
BNE #-6
MOV #VTBUFF,R0 ;RE-SET POINTER
CLR @VTCAR ;CLR CURSOR ADDR. REG.
MOV R0,@VTSAR ;LOAD STARTING ADDR.
MOV #1,@VTCSR ;ST. DISPLAY
GETREG
RTI

;*****
;SUBROUTINE ENTERED AT THE END OF EVERY DISPLAY TEST TO CHECK STATUS
;OF THE DATA SWITCHES
;*****

XTHEND: INC DONESW
XENDT: DELAYL
JSR PC,TTYENB ;ISSUE DELAY
TST RUNSW ;ENABLE TTY INTERRUPTS
BNE RUNEM ;RUNING ALL TESTS?
BIT #SW12,@SWR ;YES, DO IT
BNE RUNEM ;MANUAL ADVANCE SW. SET?
MOV DONESW,HOLDSO ;YES
JSR PC,TSWAP ;SAVE SW.
;TEST FOR SWAP SW.

```

4072 015770 005737 024240
 4073 015774 001354
 4074 015776 000002
 4075 016000 104020
 4076 016002 004737 014632
 4077 016006 062716 000002
 4078 016012 000002
 4079
 4080

TST HOLDSO ; WAS SW. SET TO LOOP TEST?
 BNE XTHEND ; YES, LOOP IT.
 RTI ; NORMAL EXIT
 RUNEM: DELAYL
 JSR PC, TSWAP ; TEST SWAP SW.
 RUNEXT: ADD #2, (SP) ; ADVANCE TO NEXT TEST
 RTI

4081
 4082
 4083
 4084
 4085
 4086
 4087
 4088
 4089
 4090
 4091
 4092
 4093
 4094
 4095
 4096
 4097
 4098
 4099
 4100
 4101
 4102
 4103
 4104
 4105
 4106
 4107
 4108
 4109
 4110
 4111
 4112
 4113
 4114

```

;*****
;POWER FAIL HANDLER
;*****
PWRFAL: MOV R0, -(SP) ;SAVE REGISTERS ON STACK
        MOV R1, -(SP)
        MOV R2, -(SP)
        MOV R3, -(SP)
        MOV R4, -(SP)
        MOV R5, -(SP)
        MOV 24, -(SP)
        MOV SP, TEMP1 ;SAVE STACK POINTER
        MOV #PWRUP, @#24 ;POWER UP ROUTINE TO LOCATION 24
        HALT ;HALT
PWRUP: MOV #340, @PSW ;LOCK OUT INTERRUPTS
        MOV TEMP1, SP ;POWER DOWN ROUTINE TO LOCATION 24
        MOV (SP)+, @#24
        MOV (SP)+, R5 ;RESTORE REGISTERS
        MOV (SP)+, R4
        MOV (SP)+, R3
        MOV (SP)+, R2
        MOV (SP)+, R1
        MOV (SP)+, R0
        RESET
        CLR R1 ;POWER UP DELAY
        INC R1
        BNE .-2
        DISPLAY
        MES21
        JMP @RETURN
  
```

4115 016122 012637 024450
 4116 016126 012637 024452
 4117 016132 012637 024454
 4118 016136 012637 024456
 4119 016142 010146
 4120 016144 010246
 4121 016146 010346
 4122 016150 010446
 4123 016152 010546
 4124 016154 013746 024456
 4125 016160 013746 024454
 4126 016164 013746 024452
 4127 016170 013746 024450

```

;*****
;SUBROUTINE TO SAVE 'R1-R5' ON STACK
;*****
XSAVRG: MOV (SP)+, SAVEPC
        MOV (SP)+, SAVPSW
        MOV (SP)+, SAV2PC
        MOV (SP)+, SAV2SW
        MOV R1, -(SP)
        MOV R2, -(SP)
        MOV R3, -(SP)
        MOV R4, -(SP)
        MOV R5, -(SP)
        MOV SAV2SW, -(SP)
        MOV SAV2PC, -(SP)
        MOV SAVPSW, -(SP)
        MOV SAVEPC, -(SP)
  
```

```

4128 016174 000002
4129
4130
4131
4132
4133 016176 012637 024450
4134 016202 012637 024452
4135 016206 012637 024454
4136 016212 012637 024456
4137 016216 012605
4138 016220 012604
4139 016222 012603
4140 016224 012602
4141 016226 012601
4142 016230 013746 024456
4143 016234 013746 024454
4144 016240 013746 024452
4145 016244 013746 024450
4146 016250 000002
4147
4148
4149
4150
4151
4152 016252 005077 163232
4153 016256 012701 001000
4154 016262 011602
4155 016264 062716 000002
4156 016270 012700 024474
4157 016274 011220
4158 016276 005301
4159 016300 001375
4160 016302 112710 014431
4161 016306 012700 024474
4162 016312 010077 163170
4163 016316 000002
4164
4165
4166
4167
4168
4169 016320 123737 024134 024136
4170 016326 001414
4171 016330 022737 000001 024134
4172 016336 001417
4173 016340 022737 000002 024134
4174 016346 001420
4175 016350 022737 000003 024134
4176 016356 001421
4177 016360 005037 024134
4178 016364 012700 001330
4179 016370 012701 001410
4180 016374 000416
4181 016376 012700 001344
4182 016402 012701 001424
4183 016406 000411
    
```

```

RTI
;*****
;SUBROUTINE TO RESTORE 'R1-R5' FROM THE STACK
;*****
XGETRG: MOV (SP)+,SAVEPC
MOV (SP)+,SAVPSW
MOV (SP)+,SAV2PC
MOV (SP)+,SAV2SW
MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV SAV2SW,-(SP)
MOV SAV2PC,-(SP)
MOV SAVPSW,-(SP)
MOV SAVEPC,-(SP)
RTI

;*****
;SUBROUTINE TO PRE-LOAD THE 'VT' DATA BUFFER WITH DATA FROM THE 'PC'.
;*****
XPRELD: CLR @VTCSR ;STOP 'VT'
MOV #512,R1 ;SET UP '512' WORDS
MOV (SP),R2 ;GET ADDR. OF DATA FROM STACK
ADD #2,(SP) ;SET UP STACK TO EXIT
MOV @VTBUFF,R0 ;SET UP BUFFER ADDR. POINTER
MOV (R2),(R0)+ ;MOVE DATA TO BUFFER.
DEC R1 ;DONE?
BNE .-4 ;NO, MOVE NEXT WORD
MOVB #14431,(R0) ;YES, TERMINATE BUFFER W/ 'ECS'
MOV @VTBUFF,R0 ;RE-SET BUFFER ADDR. POINTER
MOV R0,@VTSAR ;LOAD 'STARTING ADDR. REG.
RTI ;EXIT

;*****
;SUBROUTINE TO SET UP THE 'FK' AND THE 'VT' DEVICE ADDRESSES
;*****
STUNIT: CMPB UNITFG,UNITNO ;TESTED ALL UNITS?
BEQ SETUTO ;YES, RE-SELECT UNIT '0'
CMP #1,UNITFG ;RUNNING UNIT #1?
BEQ SETUT1 ;YES
CMP #2,UNITFG ;RUNNING UNIT #2
BEQ SETUT2
CMP #3,UNITFG ;RUNNING UNIT #3
BEQ SETUT3
SETUTO: CLR UNITFG ;NO, LOAD UNIT '0'
MOV #FKOLDB,R0
MOV #VTOCAR,R1
BR LDUNIT
SETUT1: MOV #FK1LDB,R0
MOV #VT1CAR,R1
BR LDUNIT
    
```

4184	016410	012700	001360
4185	016414	012701	001440
4186	016420	000404	
4187	016422	012700	001374
4188	016426	012701	001454
4189	016432	012702	001470
4190	016436	012703	001504
4191	016442	012022	
4192	016444	012123	
4193	016446	022702	001504
4194	016452	001373	
4195	016454	000207	
4196			
4197			
4198			
4199			
4200			
4201			
4202			
4203	016456	000005	
4204	016460	104022	
4205	016462	000240	
4206	016464	011637	024156
4207	016470	022626	
4208	016472	011637	024160
4209	016476	162737	000004 024156
4210	016504	162737	000002 024160
4211	016512	005737	024154
4212	016516	001001	
4213	016520	000000	
4214	016522	104012	
4215	016524	022126	
4216	016526	104014	
4217	016530	024156	
4218	016532	104012	
4219	016534	022150	
4220	016536	104014	
4221	016540	024160	
4222	016542	000137	002526
4223			
4224			
4225			
4226			
4227			
4228	016546	005077	162540
4229	016552	042702	160000
4230	016556	012737	177774 016652
4231	016564	012704	016642
4232	016570	012701	177777
4233	016574	005201	
4234	016576	161402	
4235	016600	100375	
4236	016602	062402	
4237	016604	004737	016620
4238	016610	005237	016652
4239	016614	001365	

```

SETUT2: MOV #FK2LDB,R0
        MOV #VT2CAR,R1
        BR LDUNIT
SETUT3: MOV #FK3LDB,R0
        MOV #VT3CAR,R1
LDUNIT: MOV #FKLDB,R2
        MOV #VTCAR,R3
SETFKA: MOV (R0)+,(R2)+
        MOV (R1)+,(R3)+
        CMP #FKLVL+2,R2
        BNE SETFKA
        RTS PC
;*****
;SUBROUTINE ENTERED ON AN ILLEGAL TRAP. THE ROUTINE REPORTS WHERE IT
;TRAPPED 'FROM' AND WHERE IT TRAPPED 'TO'. IF NO TTY IS AVAILABLE
;A 'HALT' IS EXECUTED. HERE THE USER CAN EXAMINE THE 'FROMPC'
;LOCATION AND THE 'TOPC' LOCATION TO TRACK DOWN THE ERROR.
;*****
ERTRAP: RESET ;CLEAR ALL FLAGS
        NULL
        NOP
        MOV (SP),TOPC ;SAVE LOCATION WHERE IT TRAPPED 'TO'
        POP2SP
        MOV (SP),FROMPC ;SAVE WHERE IT TRAPPED FROM
        SUB #4,TOPC
        SUB #2,FROMPC
        TST TTYSWH ;IS TTY AVAILABLE?
        BNE .+4 ;YES
        HALT ;NO, HALT
        PRINT ;TEXT 'ILLEGAL TRAP'
        MESSA
        PRTOCT
        TOPC ;TYPE 'PC TRAPPED TOO'
        PRINT
        MES6C ;TEXT 'FROM'
        PRTOCT
        FROMPC ;WHERE IT TRAPPED FROM
        JMP MONITR ;RETURN TO THE MONITOR
;*****
;SUBROUTINE TO PRINT THE DECIMAL VALUE IN R2
;*****
XBINDEC: CLR @PSW ;ENABLE INTERRUPTS
        BIC #160000,R2
        MOV #-4,R3PBT
        MOV #DECPNT,R4
TYPT1: MOV #-1,R1
TYPT2: INC R1
        SUB (R4),R2
        BPL TYPT2
        ADD (R4)+,R2
        JSR PC,DECOUT
        INC R3PBT
        BNE TYPT1
    
```

4240	016616	000002		
4241	016620	005701		
4242	016622	001003		
4243	016624	112720	000060	
4244	016630	000207		
4245	016632	152701	000060	
4246	016636	110120		
4247	016640	000207		
4248				
4249	016642	001750		
4250	016644	000144		
4251	016646	000012		
4252	016650	000001		
4253	016652	000000		
4254				
4255				
4256				
4257				
4258				
4259	016654	017677	000000	162616
4260	016662	062716	000002	
4261	016666	012777	000200	162606
4262	016674	052777	000101	162572
4263	016702	005077	162404	
4264	016706	000207		
4265				
4266				
4267				
4268	016710	012777	000340	162374
4269	016716	012777	000012	162550
4270	016724	013777	001502	162546
4271	016732	005077	162544	
4272	016736	000207		
4273				
4274				
4275				
4276				
4277				
4278	016740	017677	000000	162546
4279	016746	062716	000002	
4280	016752	012777	000240	162536
4281	016760	052777	000101	162522
4282	016766	005077	162320	
4283	016772	000207		
4284				
4285				
4286				
4287				
4288				
4289	016774	012777	000340	162310
4290	017002	042777	000101	162500
4291	017010	013777	001516	162476
4292	017016	005077	162474	
4293	017022	000207		
4294				
4295				

```

RTI
DECOUT: TST R1
        BNE DEC1
        MOVB #60,(R0)+
        RTS PC
DEC1:   BISB #60,R1
        MOVB R1,(R0)+
        RTS PC

DECPNT: 1000.
        100.
        10.
        1.
R3PBT:  .WORD 0

;*****
;SUBROUTINE TO SET UP 'FK' VECTOR ADDR. & INTR ENABLE
;*****
LDFKVT: MOV @ (SP),@FKINT ;LOAD INTR SERVICE ADDR
        ADD #2,(SP)
        MOV #200,@FKLVL
        BIS #101,@FKCSR
        CLR @PSW ;ENABLE INTERRUPTS
        RTS PC
;*****
;SUBROUTINE TO RESET 'FK' VECTOR ADDRESSES TO HALT ON INTERRUPTS
;*****
CLRFKV: MOV #340,@PSW
        MOV #12,@FKCSR
        MOV FKLVL,@FKINT
        CLR @FKLVL
        RTS PC
;*****
;SUBROUTINE TO LOAD 'VT' VECTOR ADDR. & INTR ENABLE
;*****
LDVTVT: MOV @ (SP),@VTINT ;LOAD INTR SERVICE ADDR.
        ADD #2,(SP)
        MOV #240,@VTLVL
        BIS #101,@VTCSR ;SET 'GO' & INTR 'ENABLE'
        CLR @PSW
        RTS PC
;*****
;SUBROUTINE TO RESET 'VT' VECTOR ADDRESSES TO HALT ON INTERRUPTS
;*****
CLRVTV: MOV #340,@PSW
        BIC #101,@VTCSR
        MOV VTLVL,@VTINT
        CLR @VTLVL
        RTS PC
;*****

```

```

4296
4297
4298
4299
4300 017024 112720 000362
4301 017030 104010
4302 017032 112720 000364
4303 017036 104010
4304 017040 112720 000361
4305 017044 104010
4306 017046 112720 000370
4307 017052 104010
4308 017054 112720 000360
4309 017060 000002
4310
4311
4312
4313
4314
4315 017062 005737 024154
4316 017066 001405
4317 017070 012777 000100 162216
4318 017076 005077 162210
4319 017102 000207
4320
4321
4322
4323
4324
4325
4326 017104 104013
4327 017106 012701 014600
4328 017112 004737 017200
4329 017116 010403
4330 017120 022703 000701
4331 017124 001407
4332 017126 022703 154765
4333 017132 001401
4334 017134 000002
4335 017136 104012
4336 017140 022704
4337 017142 000402
4338 017144 104012
4339 017146 022655
4340 017150 000137 002526
4341
4342 017154 005721
4343 017156 006204
4344 017160 006204
4345 017162 006204
4346 017164 005337 024146
4347 017170 001752
4348 017172 010403
4349 017174 062716 000002
4350 017200 005004
4351 017202 005737 024146
    
```

```

;SUBROUTINE TO LOAD THE 'FIELD' CONTROL WITH THE '4' MODES AND SET UP
;'VT' BUFFER.
;*****
XSTLNE:  MOVB    #BLINK,(R0)+      ;LOAD 'BLINK'
        LDLINE   ;LOAD LINE OF 'BLINKING' CHAR'S
        MOVB    #BOLD,(R0)+      ;LOAD LINE OF 'BOLD' CHAR.'S
        LDLINE   ;LOAD LINE OF 'BLANK' CHAR.'S
        MOVB    #BLANK,(R0)+    ;LOAD LINE OF 'BLANK' CHAR.'S
        LDLINE   ;LOAD LINE OF 'UNDERLINED' CHAR.'S
        MOVB    #UNLINE,(R0)+
        LDLINE
        MOVB    #CLRFLD,(R0)+
        RTI
;*****
;SUBROUTINE TO TEST IF A TTY IS AVAILABLE, AND IF SO, ENABLE
;KEYBOARD INTERRUPTS.
;*****
TTYENB:  TST     TTYSWH           ;TTY AVAILABLE?
        BEQ     .+14             ;NO, EXIT
        MOV     #100,@TKS       ;YES, ENABLE TTY INTERRUPTS
        CLR    @PSW
        RTS     PC
;*****
;SUBROUTINE TO WAIT FOR AND ASSEMBLE CHARACTERS INPUT
;FROM THE KEYBOARD INTO OCTAL NUMBERS.
;*****
XASEMB:  TTYIN                    ;GET CHAR.'S FROM KEYBOARD
        MOV     #INBUF,R1       ;SET UP CHAR. BUFFER POINTER
        JSR    PC,STRIPN        ;STRIPE NO.
XASEM1:  MOV     R4,R3           ;RETURNS HERE IF ONLY '1' NO.
        CMP    #701,R3
        BEQ    WHY
        CMP    #154765,R3
        BEQ    .+4
        RTI
        PRINT  MES19
        BR     XASEXT
WHY:    PRINT  MES18
XASEXT:  JMP     MONITR
WORD2:   TST     (R1)+          ;ADVANCE POINTER PAST COMMA
        ASR    R4
        ASR    R4
        ASR    R4
        DEC    CHRCNT          ;DEC. CHAR. CNTR.
        BEQ    XASEM1         ;COMMA LAST CHAR.?
        MOV    R4,R3          ;NO, SAVE 1ST NO.
        ADD    #2,(SP)        ;SET UP STACK TO EXIT
STRIPN:  CLR     R4
        TST    CHRCNT          ;WAS ANY DATA INPUTTED?
    
```

4352 017206 001001
 4353 017210 000207
 4354 017212 022711 000054
 4355 017216 001756
 4356 017220 042711 177770
 4357 017224 062104
 4358 017226 005337 024146
 4359 017232 003001
 4360 017234 000207
 4361 017236 006304
 4362 017240 006304
 4363 017242 006304
 4364 017244 000756

BNE .+4 ;YES, PROCESS IT
 RTS PC ;NO, RETURN
 CMP #54,(R1) ;CHAR. = COMMA?
 BEQ WORD2 ;YES, SAVE 1ST NO.
 BIC #177770,(R1) ;NO, STRIPE NO. TO OCTAL
 ADD (R1)+,R4
 DEC CHRCNT ;FINISHED?
 BGT .+4 ;NO
 RTS PC ;YES, EXIT
 ASL R4
 ASL R4
 ASL R4
 BR STRIPN+2

 ;VT MESSAGE ROUTINE, ENTERED VIA EMT DISPATCH HANDLER.
 ;ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
 ;THE ADDRESS OF MESSAGE TO BE DISPLAYED.
 ;*****

4370
 4371 017246 004737 017062
 4372 017252 005077 162232
 4373 017256 104005
 4374 017260 017637 003000 024466
 4375 017266 011637 024130
 4376 017272 162737 000002 024130
 4377 017300 062716 000002
 4378 017304 104002
 4379 017306 013702 024466
 4380 017312 122712 000100
 4381 017316 001007
 4382 017320 104003
 4383 017322 005237 024426
 4384 017326 104020
 4385 017330 005037 024426
 4386 017334 000002
 4387 017336 122712 000045
 4388 017342 001402
 4389 017344 112220
 4390 017346 000761
 4391 017350 112720 000012
 4392 017354 105722
 4393 017356 000755

VTMES: JSR PC,TTYENB ;ENABLE TTY INTERRUPTS
 CLR @VTCR ;CLR 'GO'
 EOSBUF ;PRE-LOAD BUFFER
 MOV @ (SP),KSTOR4 ;GET THE MESSAGE ADDRESS FROM STACK
 MOV (SP),AVECTR ;GET ADDRESS OF CALL
 SUB #2,AVECTR ;SAVE AS 'A' RE-START ADDR.
 ADD #2,(SP) ;SET UP STACK TO EXIT
 SAVREG ;SAVE REG'S
 MOV KSTOR4,R2
 VTERA: CMPB #100,(R2) ;TEST FOR 'a'
 BNE VTER1 ;BRANCH IF NO EQUAL
 GETREG ;RESTORE REG'S
 INC DISPSW ;SET DISPLAY SW.
 DELAYL
 CLR DISPSW ;CLR DISPLAY SW.
 RTI ;OTHERWISE EXIT
 VTER1: CMPB #45,(R2) ;TEST FOR '%'
 BEQ VTCRLF ;IF = TYPE 'CR-LF'
 VTER2: MOVB (R2)+,(R0)+ ;SAVE CHAR. IN BUFFER
 BR VTERA
 VTCRLF: MOVB #12,(R0)+ ;SAVE 'EOL' IN BUFFER
 TSTB (R2)+ ;INCREMENT BUFFER
 BR VTERA

 ;SUBROUTINE TO TRANSMIT A 'NULL' CHAR. TO THE PRINTER.
 ;*****

4394
 4395
 4396
 4397
 4398
 4399 017360 005037 014326
 4400 017364 105237 014326
 4401 017370 100375
 4402 017372 012737 000002 014326
 4403 017400 004737 017436
 4404 017404 005077 161712
 4405 017410 004737 017436
 4406 017414 005337 014326
 4407 017420 001371

XNULL: CLR SPACEX
 IS: INCB SPACEX
 BPL IS
 MOV #2,SPACEX
 JSR PC,TSTTPS
 XNULL1: CLR @TPB ;TRANSMIT A NULL CHAR.
 JSR PC,TSTTPS
 DEC SPACEX
 BNE XNULL1

4408 017422 105237 014326
 4409 017426 100375
 4410 017430 005037 014326
 4411 017434 000002
 4412
 4413 017436 105777 161656
 4414 017442 100375
 4415 017444 000207
 4416
 4417
 4418
 4419
 4420
 4421
 4422

2\$: INCB SPACEX
 BPL 2\$
 CLR SPACEX
 RTI
 TSTTPS: TSTB @TPS
 BPL -4
 RTS PC

 : ENTERED WITH SYSTEM TRAP 'TRAP' CALL (ERROR) FROM THE 'FK' LOGIC TEST
 : THIS ROUTINE SETS UP REG'S 'R0-R5' WITH DATA
 : FROM THE 'FK' LOGIC TO BE EXAMINED ON A 'HALT' IF AN 'FK' LOGIC ERROR IS
 : DETECTED AND 'NO' TELEPRINTER IS AVAILABLE.

4423 017446 005737 024154
 4424 017452 001065
 4425 017454 052777 000200 162026
 4426 017462 005777 161636
 4427 017466 100014
 4428 017470 013700 024142
 4429 017474 013701 020404
 4430 017500 005201
 4431 017502 011602
 4432 017504 162702 000002
 4433 017510 017703 161760
 4434 017514 017704 161756
 4435 017520 005777 161600
 4436 017524 100001
 4437 017526 000000
 4438 017530 000002
 4439
 4440
 4441
 4442
 4443
 4444
 4445
 4446

FKERR: TST TTYSWH ; TTY AVAILABLE?
 BNE TYPFKR ; YES, TYPE FK ERRORS
 BIS #AUDIO, @VTCSR ; ISSUE 'AUDIO BEEP' ON ERROR
 TST @SWR ; TEST IF 'SWIS' IS SET
 BPL CKHALT ; NO, INHIBIT ERROR HALT.
 MOV TSTNUM, R0 ; SAVE 'TEST NO.' IN 'R0'
 MOV SCOPEF, R1 ; SAVE PASS COUNT
 INC R1
 MOV (SP), R2 ; SAVE 'ERROR PC' IN 'R2'
 SUB #2, R2
 MOV @FKCSR, R3 ; SAVE 'FK CSR' IN 'R3'
 MOV @FKDATA, R4 ; SAVE 'FK DATA' IN 'R4'
 CKHALT: TST @SWR ; CHECK IF SWIS IS SET
 BPL .+4 ; NO
 HALT ; YES, HALT
 RTI

 : ENTERED WITH SYSTEM TRAP 'TRAP' CALL (ERROR) FROM THE 'VT' LOGIC TEST
 : THIS ROUTINE SETS UP REG'S 'R0-R5' WITH DATA FROM THE 'VT' LOGIC TEST
 : TO BE EXAMINED ON A 'HALT' IF A 'VT' ERROR IS DETECTED AN 'NO' TELEPRINTER
 : IS AVAILABLE.

4447 017532 005737 024154
 4448 017536 001051
 4449 017540 005777 161560
 4450 017544 100367
 4451 017546 104002
 4452 017550 010003
 4453 017552 010104
 4454 017554 013700 024142
 4455 017560 013701 020404
 4456 017564 005201
 4457 017566 011602
 4458 017570 162702 000002
 4459 017574 005737 024432
 4460 017600 001005
 4461 017602 017703 161702
 4462 017606 017704 161700
 4463 017612 000402

VTERR: TST TTYSWH ; TTY AVAILABLE?
 BNE TYPVTR ; YES, TYPE 'VT' ERRORS
 TST @SWR ; TEST IF SWIS IS SET
 BPL CKHALT+4 ; NO, INHIBIT ERROR HALT
 SAVREG ; SAVE REGISTERS
 MOV R0, R3 ; SAVE SHIFT COUNT
 MOV R1, R4 ; SAVE EXPECTED DATA
 MOV TSTNUM, R0 ; SAVE FAILING TEST NO.
 MOV SCOPEF, R1 ; SAVE PASS COUNTER
 INC R1
 MOV (SP), R2 ; SAVE 'PC'
 SUB #2, R2
 TST SHFTSW ; RUNNING SHIFT TEST?
 BNE VTERR1 ; YES
 MOV @VTCSR, R3 ; NO, SAVE CONTENTS OF 'CSR'
 MOV @VTMAR, R4 ; SAVE CONTENTS OF THE 'MAR'
 BR VTERR2

4464 017614 017705 161672
4465 017620 000000
4466 017622 104003
4467 017624 000002
4468
4469
4470
4471
4472
4473
4474

VTERR1: MOV @VTMAR,R5 ;SAVE CONTENTS OF 'MAR'
VTERR2: HALT ;EXAMINE REG.'S
GETREG ;RESTORE REG.'S
RTI ;EXIT

;ENTERED WITH SYSTEM TRAP 'TRAP' CALL (ERROR) FROM THE 'FK' LOGIC TEST
;THIS ROUTINE IS USED TO PRINT OUT THE CONTENTS OF THE 'FK' REGISTERS
;WHEN A LOGIC ERROR IS DETECTED

4475 017626 012737 022370 020166
4476 017634 012737 000003 024440
4477 017642 012701 001474
4478 017646 011637 024464
4479 017652 004737 020120
4480 017656 000137 017520
4481
4482
4483
4484
4485
4486
4487

TYPFKR: MOV #MES13,ERRMES ;SET UP TO PRINT LOGIC ERROR HEADER
MOV #3,TEMP1 ;SET UP TO PRINT 'FK' REG'S
MOV #FKCSR,R1
MOV (SP),KSTOR3 ;PC OF FAILING ROUTINE
JSR PC,PRERR ;PRINT ERROR DATA
JMP CKHALT ;EXIT

;ENTERED WITH SYSTEM TRAP 'TRAP' CALL (ERROR) FROM THE 'VT' LOGIC TEST
;THIS ROUTINE IS USED TO PRINT OUT THE CONTENTS OF THE 'VT' REGISTERS
;WHEN A LOGIC ERROR IS DETECTED

4488 017662 104002
4489 017664 005737 024432
4490 017670 001422
4491 017672 005737 024446
4492 017676 001004
4493 017700 005237 024446
4494 017704 005037 024444
4495 017710 012737 022571 020166
4496 017716 010037 024472
4497 017722 010137 024470
4498 017726 012737 000001 024440
4499 017734 000413
4500 017736 042777 000100 161544
4501 017744 012737 022526 020166
4502 017752 012737 000002 024440
4503 017760 012701 001510
4504 017764 011637 024464
4505 017770 004737 020120
4506 017774 032777 020000 161322
4507 020002 001043
4508 020004 005737 024432
4509 020010 001414
4510 020012 013737 024472 024464
4511 020020 004737 020236
4512 020024 104012
4513 020026 000000
4514 020030 013737 024470 024464
4515 020036 004737 020236
4516 020042 032777 000007 161254
4517 020050 001412
4518 020052 042777 000017 161430
4519 020060 017702 161240

TYPVTR: SAVREG ;SAVE REGISTERS ON STACK
TST SHFTSW ;DOING SHIFT TEST?
BEQ VTNORM ;NO, REPORT AS NORMAL ERROR
TST MESPT1 ;TYPED HEADER?
BNE +12 ;YES
INC MESPT1
CLR MESPRT
MOV #MES17,ERRMES ;YES, SET UP SHIFT HEADER
MOV RO,KSTOR6 ;SAVE SHIFT COUNT
MOV R1,KSTOR5 ;SAVE EXPT'D DATA
MOV #1,TEMP1 ;PRINT '1' DATA WORD
BR VTDUMP
VTNORM: BIC #100,@VTCSR ;CLR 'VT' INTR ENABLE
MOV #MES16,ERRMES ;SET UP TO PRINT LOGIC ERROR HEADER
MOV #2,TEMP1 ;SET UP TO PRINT 'VT' REG'S
MOV #VTCSR,R1
VTDUMP: MOV (SP),KSTOR3 ;PC OF FAILING ROUTINE
JSR PC,PRERR ;PRINT ERROR DATA
BIT #SW13,@SWR ;IS PRINT INHIBIT SW. SET?
BNE EXTVTR ;YES, EXIT
TST SHFTSW ;DOING SHIFT TEST?
BEQ GETMAR ;NO, GET AND TYPE 'MAR'
MOV KSTOR6,KSTOR3 ;YES, SET UP TO TYPE SHIFT COUNT.
JSR PC,XPRTA ;TYPE SHIFT COUNT
PRINT ;PRINT IF DATA IS SHIFTED IN OR OUT
SHFMES: 0
MOV KSTOR5,KSTOR3 ;SET UP TO TYPE EXPT'D DATA
JSR PC,XPRTA ;TYPE EXPT'D DATA
GETMAR: BIT #7,@SWR ;ARE THE 'MAR' SELECT BITS SET?
BEQ TYPMAR ;NO, PRINT MAR AS IS
BIC #17,@VTCSR ;YES, RESELECT THE 'MAR'
MOV @SWR,R2 ;GET 'MAR' ADDR. FROM SW'S

4520 020064 042702 177770
 4521 020070 006302
 4522 020072 050277 161412
 4523 020076 017737 161410 024464
 4524 020104 004737 020236
 4525 020110 104022
 4526 020112 104003
 4527 020114 000137 017520
 4528
 4529
 4530
 4531
 4532
 4533
 4534 020120 005737 020462
 4535 020124 001402
 4536 020126 005237 020270
 4537 020132 032777 020000 161164
 4538 020140 001044
 4539 020142 162737 000002 024464
 4540 020150 042777 000100 161316
 4541 020156 005737 024444
 4542 020162 001004
 4543 020164 104012
 4544 020166 000000
 4545 020170 005237 024444
 4546 020174 104012
 4547 020176 023005
 4548 020200 104014
 4549 020202 024142
 4550 020204 104016
 4551 020206 013737 020404 024442
 4552 020214 005237 024442
 4553 020220 104014
 4554 020222 024442
 4555 020224 104016
 4556 020226 000403
 4557 020230 012102
 4558 020232 011237 024464
 4559 020236 104014
 4560 020240 024464
 4561 020242 104016
 4562 020244 005337 024440
 4563 020250 003367
 4564 020252 005737 020270
 4565 020256 001403
 4566 020260 104012
 4567 020262 023752
 4568 020264 000000
 4569
 4570 020266 000207
 4571 020270 000000

```

BIC      #177770,R2
ASL      R2
BIS      R2,AVTCSR      ;SET UP SELECTED 'MAR' REG.
TYPMAR:  MOV     AVTCSR,KSTOR3
          JSR     PC,XPRTA
          NULL
EXTVTR:  GETREG      ;TRANSMIT 'NULL' CHARACTER
          JMP     CKHALT      ;RESTORE REGISTERS FROM STACK.
          ;EXIT

;*****
;SUBROUTINE TO PRINT OUT ERROR INFORMATION ON 'FK' & 'VT' LOGIC
;ERRORS.
;*****

PRERR:   TST     XORFLG
          BEQ     .+6
          INC     XORHLT
          BIT     #20000,ASW      ;TEST SW-13 FOR INHIBIT PRINT OUT
          BNE     EXTPRT      ;INHIBIT,CHECK FOR HALT
          SUB     #2,KSTOR3
          BIC     #100,AFKCSR      ;CLR 'FK' INTR. ENABLE
          TST     MESPRT      ;HEADER BEEN TYPED?
          BNE     .+12      ;YES,
          PRINT
ERRMES:  O
          INC     MESPRT      ;PRINT LOGIC ERROR HEADER
          PRINT      ;SET PRINT INHIBIT SW.
          CRLF
          PRTCT  TSTNUM      ;PRINT FAILING TEST NO.
          SPACE
          MOV     SCOPEF,TEMP2
          INC     TEMP2
          PRTCT  TEMP2
          SPACE      ;PRINT THE FAILING PASS NO.
          BR
XPRT:    MOV     (R1)+,R2
          MOV     (R2),KSTOR3
XPRTA:   PRTCT  KSTOR3
          SPACE
          DEC     TEMP1
          BGT     XPRT
EXTPRT:  TST     XORHLT
          BEQ     1$
          PRINT
          MXORH
          HALT

1$:      RTS     PC
XORHLT:  000000
;*****
;SCOPE AND/OR ITERATION LOOP FOR EACH LOGIC TEST
;*****

```

4572
 4573
 4574
 4575

```

4576 020272 104017          SCOPEC: TSTTKS          ;CHECK FOR KEYBOARD FLAG
4577 020274 032777 040000 161022      BIT      #40000, @SWR    ;TEST SW-14 FOR SCOPE
4578 020302 001015          BNE      SCOPEB        ;YES, SCOPE
4579 020304 005737 020462          TST      XORFLG        ;SEE IF RUNNING WITH XOR
4580 020310 001037          BNE      XORT          ;
4581 020312 032777 004000 161004      SCOPE1: BIT      #4000, @SWR ;NO-TEST SW-11 FOR ITERATION
4582 020320 001013          BNE      SCOPEG        ;INHIBIT ITERATION
4583 020322 023737 020404 020402      CMP      SCOPEF, ICOUNT ;COMPARE CURRENT COUNT TO MAX NUMBER
4584 020330 100007          BPL      SCOPEG        ;EXIT-DONE
4585 020332 005237 020404          INC      SCOPEF        ;INCREMENT COUNT
4586 020336 022606          SCOPEB: CMP      (6)+, SP ;REPOSITION STACK
4587 020340 012677 160746          MOV      (6)+, @PSW    ;RESTORE PREVIOUS PROCESSOR STATUS
4588 020344 000177 000036          JMP      @RETURN       ;REPEAT TEST
4589 020350 005037 020404          SCOPEG: CLR      SCOPEF ;CLEAR COUNT
4590 020354 011601          MOV      @SP, R1      ;SAVE TEST NO.
4591 020356 011137 024142          MOV      (R1), TSTNUM ;
4592 020362 062716 000002          ADD      #2, (SP)     ;
4593 020366 011637 020406          MOV      @SP, RETURN  ;SAVE SCOPE RETURN POINTER
4594 020372 052777 000200 161110      BIS      #AUDIO, @VTCR ;ISSUE 'AUDIO BEEP' ON END OF TEST
4595 020400 000002          RTI                    ;RETURN INLINE-NEXT TEST
4596 020402 000200          ICOUNT: 200          ;ITERATION COUNT
4597 020404 000000          SCOPEF: 0            ;COUNT LOCATION FOR ITERATION LOOP
4598 020406 006246          RETURN: FKT1         ;
4599                                     ;*****BEGINNING OF XOR TESTING
4600 020410 013746 000004          XORT:  MOV      @#4, -(6) ;SAVE CONTENTS OF LOCATION 4
4601 020414 012737 020434 000004      MOV      #XORERR, @#4 ;SET UP FOR XOR TIMEOUT
4602 020422 005737 177060          TST      @#177060    ;IF XOR TIMES OUT IT DETECTED AN ERROR
4603 020426 012637 000004          MOV      (6)+, @#4   ;NO TIMEOUT-RESTORE LOCATION 4
4604 020432 000727          BR      SCOPE1       ;RETURN TO TEST
4605 020434 022626          XORERR: CMP      (6)+, (6)+ ;CLEAR STACK
4606 020436 012637 000004          MOV      (6)+, @#4   ;RESTORE LOCATION 4
4607 020442 032777 000001 160654      BIT      #1, @SWR    ;RING BELL ON ERROR?
4608 020450 001402          BEQ      1$          ;IF NO, SKIP AHEAD
4609 020452 104012          PRINT          ;SEND BELL
4610 020454 023750          MBELL          ;
4611 020456 000177 177724          1$:  JMP      @RETURN  ;
4612 020462 000000          XORFLG: 000000     ;
4613
4614
4615                                     ;*****
4616                                     ;KEYBOARD CHARACTER TABLE
4617                                     ;*****
4618                                     ;START, TOP UN-SHIFTED
4619 020464          133          CHRTAB: .BYTE  '1
4620 020465          061          .BYTE  '1
4621 020466          062          .BYTE  '2
4622 020467          063          .BYTE  '3
4623 020470          064          .BYTE  '4
4624 020471          065          .BYTE  '5
4625 020472          066          .BYTE  '6
4626 020473          067          .BYTE  '7
4627 020474          070          .BYTE  '8
4628 020475          071          .BYTE  '9
4629 020476          060          .BYTE  '0
4630 020477          055          .BYTE  '-'
4631 020500          137          .BYTE  '+'

```

```

4632 020501 135 .BYTE 'J
4633
4634 ;2ND ROW LOWER CASE
4635
4636 020502 011 .BYTE 11 ;TAB
4637 020503 033 .BYTE 33 ;ALTMODE
4638 020504 161 .BYTE 161 ;Q
4639 020505 167 .BYTE 167 ;W
4640 020506 145 .BYTE 145 ;M
4641 020507 162 .BYTE 162 ;U
4642 020510 164 .BYTE 164 ;Y
4643 020511 171 .BYTE 171 ;U
4644 020512 165 .BYTE 165 ;I
4645 020513 151 .BYTE 151 ;O
4646 020514 157 .BYTE 157 ;P
4647 020515 160 .BYTE 160
4648 020516 100 .BYTE '0
4649 020517 012 .BYTE 12 ;LF'
4650 020520 015 .BYTE 15 ;CR'
4651
4652 ;3RD ROW LOWER CASE
4653
4654 020521 136 .BYTE '↑
4655 020522 141 .BYTE 141 ;S
4656 020523 163 .BYTE 163 ;D
4657 020524 144 .BYTE 144 ;O
4658 020525 146 .BYTE 146 ;U
4659 020526 147 .BYTE 147 ;G
4660 020527 150 .BYTE 150 ;H
4661 020530 152 .BYTE 152 ;I
4662 020531 153 .BYTE 153 ;J
4663 020532 154 .BYTE 154 ;K
4664 020533 073 .BYTE '·
4665 020534 072 .BYTE '·
4666 020535 177 .BYTE 177 ;RUBOUT
4667
4668 ;4TH ROW LOWER CASE
4669
4670 020536 172 .BYTE 172 ;Z
4671 020537 170 .BYTE 170 ;X
4672 020540 143 .BYTE 143 ;C
4673 020541 166 .BYTE 166 ;V
4674 020542 142 .BYTE 142 ;B
4675 020543 156 .BYTE 156 ;N
4676 020544 155 .BYTE 155 ;M
4677 020545 054 .BYTE '·
4678 020546 056 .BYTE '·
4679 020547 057 .BYTE '·
4680 020550 134 .BYTE '·
4681 020551 040 .BYTE 40 ;TERMINATE W/ SPACE
4682
4683 ;START OF 'SHIFTED' PASS
4684
4685 020552 173 UPCASE: .BYTE 173 ;LEFT BRACE
4686 020553 041 .BYTE '!'
4687 020554 042 .BYTE '·

```

4688	020555	043	.BYTE	'#	
4689	020556	044	.BYTE	'\$	
4690	020557	045	.BYTE	'%	
4691	020560	046	.BYTE	'&	
4692	020561	047	.BYTE	'	
4693	020562	050	.BYTE	'('	
4694	020563	051	.BYTE	')	
4695	020564	060	.BYTE	'0	
4696	020565	075	.BYTE	'='	
4697	020566	137	.BYTE	'+'	
4698	020567	175	.BYTE		;RIGHT BRACE
4699					
4700					
4701					;2ND ROW 'SHIFTED'
4702	020570	011	.BYTE	11	
4703	020571	033	.BYTE	33	;ALTMODE
4704	020572	121	.BYTE	'Q	
4705	020573	127	.BYTE	'W	
4706	020574	105	.BYTE	'E	
4707	020575	122	.BYTE	'R	
4708	020576	124	.BYTE	'T	
4709	020577	131	.BYTE	'Y	
4710	020600	125	.BYTE	'U	
4711	020601	111	.BYTE	'I	
4712	020602	117	.BYTE	'O	
4713	020603	120	.BYTE	'P	
4714	020604	140	.BYTE	140	
4715	020605	012	.BYTE	12	;'LF
4716	020606	015	.BYTE	15	;'CR'
4717					
4718					;3RD ROW 'SHIFTED'
4719					
4720	020607	176	.BYTE	176	;'↑'
4721	020610	101	.BYTE	'A	
4722	020611	123	.BYTE	'S	
4723	020612	104	.BYTE	'D	
4724	020613	106	.BYTE	'F	
4725	020614	107	.BYTE	'G	
4726	020615	110	.BYTE	'H	
4727	020616	112	.BYTE	'J	
4728	020617	113	.BYTE	'K	
4729	020620	114	.BYTE	'L	
4730	020621	053	.BYTE	'+'	
4731	020622	052	.BYTE	'*'	
4732	020623	177	.BYTE	177	;RUBOUT
4733					
4734					;4TH ROW 'SHIFTED'
4735					
4736	020624	132	.BYTE	'Z	
4737	020625	130	.BYTE	'X	
4738	020626	103	.BYTE	'C	
4739	020627	126	.BYTE	'V	
4740	020630	102	.BYTE	'B	
4741	020631	116	.BYTE	'N	
4742	020632	115	.BYTE	'M	
4743	020633	074	.BYTE	'<	

```
4744 020634 076 .BYTE '>'
4745 020635 077 .BYTE '? '
4746 020636 174 .BYTE '174'
4747 020637 040 .BYTE '40' ; TERMINATE W/ SPACE
4748
4749 ; START OF 'CONTROL' PASS
4750
4751 020640 033 CONTRL: .BYTE '33' ; 'tl'
4752 020641 061 .BYTE '1'
4753 020642 062 .BYTE '2'
4754 020643 063 .BYTE '3'
4755 020644 064 .BYTE '4'
4756 020645 065 .BYTE '5'
4757 020646 066 .BYTE '6'
4758 020647 067 .BYTE '7'
4759 020650 070 .BYTE '8'
4760 020651 071 .BYTE '9'
4761 020652 060 CNTRIT: .BYTE '0'
4762 020653 015 .BYTE '15'
4763 020654 037 .BYTE '37' ; 't+'
4764 020655 035 .BYTE '35' ; 't' RIGHT BRACKET
4765
4766 ; 2ND ROW 'CONTROL'
4767
4768 020656 011 .BYTE '11'
4769 020657 033 .BYTE '33' ; ALTMODE
4770 020660 021 .BYTE '21' ; 'tQ'
4771 020661 027 .BYTE '27' ; 'tW'
4772 020662 005 .BYTE '05' ; 'tE'
4773 020663 022 .BYTE '22' ; 'tR'
4774 020664 024 .BYTE '24' ; 'tT'
4775 020665 031 .BYTE '31' ; 'tY'
4776 020666 025 .BYTE '25' ; 'tU'
4777 020667 011 .BYTE '11' ; 'tI'
4778 020670 017 .BYTE '17' ; 'tO'
4779 020671 020 .BYTE '20' ; 'tP'
4780 020672 000 .BYTE '00' ; 'tB'
4781 020673 012 .BYTE '12' ; 'LF'
4782 020674 015 .BYTE '15' ; 'CR'
4783
4784 ; 3RD ROW 'CONTROL'
4785
4786 020675 036 .BYTE '36' ; 't+'
4787 020676 001 .BYTE '01' ; 'tA'
4788 020677 023 .BYTE '23' ; 'tS'
4789 020700 004 .BYTE '04' ; 'tD'
4790 020701 006 .BYTE '06' ; 'tF'
4791 020702 007 .BYTE '07' ; 'tG'
4792 020703 010 .BYTE '10' ; 'tH'
4793 020704 012 .BYTE '12' ; 'tJ'
4794 020705 013 .BYTE '13' ; 'tK'
4795 020706 014 .BYTE '14' ; 'tL'
4796 020707 073 .BYTE '73' ;
4797 020710 072 .BYTE '72' ;
4798 020711 177 .BYTE '177' ; 't' RUBOUT
4799
```

```

4800
4801
4802 020712 032
4803 020713 030
4804 020714 003
4805 020715 026
4806 020716 002
4807 020717 016
4808 020720 015
4809 020721 054
4810 020722 056
4811 020723 057
4812 020724 034
4813 020725 040
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825 020726 360
4826 020727 040
4827 020730 361
4828 020731 041
4829 020732 362
4830 020733 042
4831 020734 364
4832 020735 043
4833 020736 370
4834 020737 044
4835 020740 363
4836 020741 045
4837 020742 365
4838 020743 046
4839 020744 367
4840 020745 047
4841 020746 371
4842 020747 050
4843 020750 373
4844 020751 061
4845 020752 375
4846 020753 072
4847 020754 377
4848 020755 107
4849 020756 366
4850 020757 106
4851 020760 372
4852 020761 170
4853 020762 374
4854 020763 123
4855 020764 376

```

;4TH ROW 'CONTROL'

```

.BYTE 32 ;'Z'
.BYTE 30 ;'X'
.BYTE 03 ;'C'
.BYTE 26 ;'V'
.BYTE 02 ;'B'
.BYTE 16 ;'N'
.BYTE 15 ;'M'
.BYTE ' '
.BYTE ' '
.BYTE '/'
.BYTE 34
CHREND: .BYTE 40 ;SPACE, 'DONE'
.EVEN

```

```

*****
; 'VT' SHIFT DATA TABLE USED BY THE 'VT' LOGIC SHIFT TEST
; THIS BUFFER IS SHIFTED IN 2 PASSES (64 TIMES IN ALL) TO TEST THE
; SHIFT REGISTER INPUT & OUTPUT LOGIC. AFTER EACH NPR TRANSFER IS MADE
; THE DATA FROM THIS TABLE IS COMPARED TO THE CONTENTS OF THE MAINTENANCE
; ADDRESS REGISTER. THE SHIFTS ARE COMMENTED IN OCTAL AND REPORTED IN
; OCTAL. THE LOW BYTE OF EACH SHIFT IS A CONTROL FIELD CHARACTER AND
; THE HIGH BYTE IS VALID DATA CHARACTER.
*****

```

```

SHFTBF: .BYTE 360
.BYTE 40 ;SHIFT '1' OR '41'
.BYTE 361 ;SHIFT '2' OR '42'
.BYTE 41 ;SHIFT '3' OR '43'
.BYTE 362 ;SHIFT '4' OR '44'
.BYTE 42 ;SHIFT '5' OR '45'
.BYTE 364 ;SHIFT '6' OR '46'
.BYTE 43 ;SHIFT '7' OR '47'
.BYTE 370 ;SHIFT '10' OR '50'
.BYTE 44 ;SHIFT '11' OR '51'
.BYTE 363 ;SHIFT '12' OF '52'
.BYTE 45 ;SHIFT '13' OR '53'
.BYTE 365 ;SHIFT '14' OR '54'
.BYTE 46 ;SHIFT '15' OR '55'
.BYTE 367 ;SHIFT '16' OR '56'
.BYTE 47 ;SHIFT '17' OR '57'
.BYTE 371
.BYTE 50
.BYTE 373
.BYTE 61
.BYTE 375
.BYTE 72
.BYTE 377
.BYTE 107
.BYTE 366
.BYTE 106
.BYTE 372
.BYTE 170
.BYTE 374
.BYTE 123
.BYTE 376

```


4856	020765	137	.BYTE	137	;SHIFT '20' OR '60'
4857	020766	360	.BYTE	360	
4858	020767	177	.BYTE	177	;SHIFT '21' OR '61'
4859	020770	360	.BYTE	360	
4860	020771	200	.BYTE	200	;SHIFT '22' OR '62'
4861	020772	376	.BYTE	376	
4862	020773	237	.BYTE	237	;SHIFT '23' OR '63'
4863	020774	374	.BYTE	374	
4864	020775	277	.BYTE	277	;SHIFT '24' OR '64'
4865	020776	372	.BYTE	372	
4866	020777	253	.BYTE	253	;SHIFT '25' OR '65'
4867	021000	366	.BYTE	366	
4868	021001	235	.BYTE	235	;SHIFT '26' OR '66'
4869	021002	360	.BYTE	360	
4870	021003	040	.BYTE	40	;SHIFT '27' OR '67'
4871	021004	361	.BYTE	361	
4872	021005	041	.BYTE	41	;SHIFT '30' OR '70'
4873	021006	363	.BYTE	363	
4874	021007	043	.BYTE	43	;SHIFT '31' OR '71'
4875	021010	362	.BYTE	362	
4876	021011	045	.BYTE	45	;SHIFT '32' OR '72'
4877	021012	364	.BYTE	364	
4878	021013	047	.BYTE	47	;SHIFT '33' OR '73'
4879	021014	366	.BYTE	366	
4880	021015	050	.BYTE	50	;SHIFT '34' OR '74'
4881	021016	365	.BYTE	365	
4882	021017	070	.BYTE	70	;SHIFT '35' OR '75'
4883	021020	367	.BYTE	367	
4884	021021	071	.BYTE	71	;SHIFT '36' OR '76'
4885	021022	370	.BYTE	370	
4886	021023	072	.BYTE	72	;SHIFT '37' OR '77'
4887	021024	360	.BYTE	360	
4888	021025	031	.BYTE	31	;SHIFT '40' OR '100'

SHFEND:

```

;*****
;MESSAGES
;*****

```

4893	021026	000			
4894	021027	045	053045	031124	
4895	021034	020060	044504	043501	
4896	021042	047516	052123	041511	
4897	021050	052040	051505	020124	
4898	021056	040515	047111	042504	
4899	021064	026503	030461	042055	
4900	021072	053102	040524	042055	
4901	021100	050055	020102	100	
4902					
4903	021105	042	045506	051440	
4904	021112	044527	041524	020110	
4905	021120	020046	044514	044107	
4906	021126	020124	042524	052123	
4907	021134	026042	051040	040505	
4908	021142	054504	043040	051117	
4909	021150	044440	050116	052125	
4910	021156	040045			
4911	021160	040445	042522	042040	

TITLE: .ASCII '%VT20 DIAGNOSTIC TEST MAINDEC-11-DBVTA-D-PB 3'

MES1: .ASCII '"FK SWITCH & LIGHT TEST", READY FOR INPUT%3'

MES1A: .ASCII ;%ARE DEFAULT ADDRESSES & VECTORS OK (Y OR N) ? 3;

4912	021166	043105	052501	052114	
4913	021174	040440	042104	042522	
4914	021202	051523	051505	023040	
4915	021210	053040	041505	047524	
4916	021216	051522	047440	020113	
4917	021224	054450	047440	020122	
4918	021232	024516	037440	040040	
4919					
4920	021240	021045	045506	046040	MES2: .ASCII '%FK LOGIC TEST'%@'
4921	021246	043517	041511	052040	
4922	021254	051505	021124	022445	
4923	021262	100			
4924	021263	045	042524	052123	MES3: .ASCII ;%TEST NO. (0-22)%@;
4925	021270	047040	027117	024040	
4926	021276	026460	031062	020051	
4927	021304	027056	027056	040045	
4928					
4929	021312	043042	020113	044103	MES4: .ASCII ;"FK CHARACTER TEST",INPUT 'LOWER CASE' CHARACTERS.%@;
4930	021320	051101	041501	042524	
4931	021326	020122	042524	052123	
4932	021334	026042	047111	052520	
4933	021342	020124	046047	053517	
4934	021350	051105	041440	051501	
4935	021356	023505	041440	040510	
4936	021364	040522	052103	051105	
4937	021372	027123	040045		
4938					
4939	021376	052045	041125	020105	MES5: .ASCII ;%TUBE '0', '1', '2', '3', OR '4'? @;
4940	021404	030047	026047	023440	
4941	021412	023461	026040	031047	
4942	021420	026047	031447	026047	
4943	021426	047440	020122	032047	
4944	021434	037447	040040		
4945					
4946	021440	043045	051117	042511	MFCP: .ASCII /%FORIEGN CHARACTER SET ('Y' OR 'N' ?)@/
4947	021446	047107	041440	040510	
4948	021454	040522	052103	051105	
4949	021462	051440	052105	024040	
4950	021470	054447	020047	051117	
4951	021476	023440	023516	037440	
4952	021504	040051			
4953	021506	044445	050116	052125	MES6: .ASCII ;%INPUT CHAR., TEST 1.@;
4954	021514	041440	040510	027122	
4955	021522	020054	042524	052123	
4956	021530	030440	040056		
4957	021534	044445	050116	052125	MES6A: .ASCII ;%INPUT CHAR., TEST 2.@;
4958	021542	041440	040510	027122	
4959	021550	020054	042524	052123	
4960	021556	031040	040056		
4961	021562	044445	050116	052125	MES6B: .ASCII ;%INPUT CHAR., TEST 3.@;
4962	021570	041440	040510	027122	
4963	021576	020054	042524	052123	
4964	021604	031440	040056		
4965					
4966	021610	021045	042524	052123	MES7: .ASCII ;%"TEST COMPLETE" @;
4967	021616	041440	046517	046120	

4968	021624	052105	021105	020040	
4969	021632	040040			
4970					
4971					
4972	021634	042524	052123	040440	MES8: .ASCII ;TEST ADDR.? a;
4973	021642	042104	027122	020077	
4974	021650	100			
4975					
4976	021651	042	052126	030062	MES9: .ASCII ;"VT20 SYSTEM EXERCISER TEST". ;
4977	021656	051440	051531	042524	
4978	021664	020115	054105	051105	
4979	021672	044503	042523	020122	
4980	021700	042524	052123	027042	
4981	021706	040			
4982	021707	012			.BYTE 12
4983	021710	012			.BYTE 12
4984	021711	116	052117	035105	.ASCII ;NOTE: THE HOST DIAGNOSTIC, EITHER FOR THE PDP-8 OR 11 MUST BE;
4985	021716	052040	042510	044040	
4986	021724	051517	020124	044504	
4987	021732	043501	047516	052123	
4988	021740	041511	020054	044505	
4989	021746	044124	051105	043040	
4990	021754	051117	052040	042510	
4991	021762	050040	050104	034055	
4992	021770	047440	020122	030461	
4993	021776	046440	051525	020124	
4994	022004	042502			
4995	022006	012			.BYTE 12
4996	022007	114	040517	042504	.ASCII ;LOADED INTO THE HOST COMPUTER BEFORE RUNNING THIS TEST.;
4997	022014	020104	047111	047524	
4998	022022	052040	042510	044040	
4999	022030	051517	020124	047503	
5000	022036	050115	052125	051105	
5001	022044	041040	043105	051117	
5002	022052	020105	052522	047116	
5003	022060	047111	020107	044124	
5004	022066	051511	052040	051505	
5005	022074	027124			
5006	022076	031			.BYTE 31
5007					
5008	022077	042	052126	047440	MES10: .ASCII ;"VT ODD ADDRESS TEST".a;
5009	022104	042104	040440	042104	
5010	022112	042522	051523	052040	
5011	022120	051505	021124	040056	
5012					
5013					
5014	022126	044445	046114	043505	MESSA: .ASCII ;%ILLEGAL TRAP TO a;
5015	022134	046101	052040	040522	
5016	022142	020120	047524	040040	
5017					
5018	022150	043040	047522	020115	MES6C: .ASCII ; FROM a;
5019	022156	100			
5020	022157	103	040510	051522	MES80: .ASCII ;CHARS.: 0000 RECV'D: 0000 XFERS.: 0000 ;
5021	022164	035056	030040	030060	
5022	022172	020060	051040	041505	
5023	022200	023526	035104	030040	

5024	022206	030060	020060	054040	
5025	022214	042506	051522	035056	
5026	022222	030040	030060	020060	
5027	022230	040			
5028	022231	105	051122	027123	.ASCII ;ERRS.: 0000;
5029	022236	020072	030060	030060	
5030	022244	012			.BYTE 12
5031	022245	012			.BYTE 12
5032	022246	031			.BYTE 31
5033					
5034		022250			.EVEN
5035	022250	000			.BYTE
5036	022251	124	044510	020123	MES11: .ASCII ;THIS MESSAGE IS BEING LOADED FROM AN ODD STARTING ;
5037	022256	042515	051523	043501	
5038	022264	020105	051511	041040	
5039	022272	044505	043516	046040	
5040	022300	040517	042504	020104	
5041	022306	051106	046517	040440	
5042	022314	020116	042117	020104	
5043	022322	052123	051101	044524	
5044	022330	043516	040		
5045	022333	101	042104	042522	.ASCII ;ADDRESS. HOW DOES IT LOOK??;
5046	022340	051523	020056	047510	
5047	022346	020127	042040	042517	
5048	022354	020123	052111	046040	
5049	022362	047517	037513	077	
5050	022367	031			.BYTE 31
5051	022370	052045	052123	047040	MES13: .ASCII ;%TST NO PASS MA CSR DATA@;
5052	022376	020117	050040	051501	
5053	022404	020123	020040	046440	
5054	022412	020101	020040	041440	
5055	022420	051123	020040	020040	
5056	022426	040504	040524	100	
5057	022433	042	045506	051040	MES14: .ASCII ;"FK REPEATIBILITY TEST", READY FOR INPUT.%@;
5058	022440	050105	040505	044524	
5059	022446	044502	044514	054524	
5060	022454	052040	051505	021124	
5061	022462	020054	042522	042101	
5062	022470	020131	047506	020122	
5063	022476	047111	052520	027124	
5064	022504	040045			
5065	022506	041445	040510	027122	MES15: .ASCII ;%CHAR. CODE(S) @;
5066	022514	041440	042117	024105	
5067	022522	024523	040040		
5068					
5069	022526	022445	051524	020124	MES16: .ASCII ;%TST NO PASS MA CSR MAR@;
5070	022534	047516	020040	040520	
5071	022542	051523	020040	020040	
5072	022550	040515	020040	020040	
5073	022556	051503	020122	020040	
5074	022564	046440	051101	100	
5075					
5076	022571	045	052045	052123	MES17: .ASCII ;%TST NO PASS MA SHIFT IN/OUT EXPT'D RECV'D@;
5077	022576	047040	020117	050040	
5078	022604	051501	020123	020040	
5079	022612	046440	020101	020040	

5080	022620	044123	043111	020124		
5081	022626	044440	027516	052517		
5082	022634	020124	042440	050130		
5083	022642	023524	020104	042522		
5084	022650	053103	042047	100		
5085	022655	045	042502	040503	MES18:	.ASCII ;%BECAUSE I LOVE YOU!!%;
5086	022662	051525	020105	020111		
5087	022670	047514	042526	054440		
5088	022676	052517	020441	040045		
5089	022704	044445	020106	047516	MES19:	.ASCII ;%IF NOT WITH THE ONE YOU LOVE,;
5090	022712	020124	044527	044124		
5091	022720	052040	042510	047440		
5092	022726	042516	054440	052517		
5093	022734	046040	053117	026105		
5094	022742	046040	053117	020105		.ASCII ; LOVE THE ONE YOUR WITH.%;
5095	022750	044124	020105	047117		
5096	022756	020105	047531	051125		
5097	022764	053440	052111	027110		
5098	022772	040045				
5099						
5100	022774	041536	027045	100	CNTRLC:	.ASCII '↑C%.';
5101	023001	136	022501	100	CNTRLA:	.ASCII '↑A%.';
5102						
5103	023005	045	100		CRLF:	.ASCII '%.';
5104						
5105						
5106	023007	077	020040	100	QMARK:	.ASCII '? .';
5107						
5108						
5109	023013	045	053042	031124	MES20:	.ASCII ;%"VT20 LOGIC TEST"%;
5110	023020	020060	047514	044507		
5111	023026	020103	042524	052123		
5112	023034	022442	100			
5113						
5114	023037	045	051045	041505	MES21:	.ASCII '%:RECOVERED FROM POWER FAILURE .';
5115	023044	053117	051105	042105		
5116	023052	043040	047522	020115		
5117	023060	047520	042527	020122		
5118	023066	040506	046111	051125		
5119	023074	020105	100			
5120						
5121	023077	111	050116	052125	MES22:	.ASCII ;INPUT THE 'UPPER CASE' CHARACTERS.%;
5122	023104	052040	042510	023440		
5123	023112	050125	042520	020122		
5124	023120	040503	042523	020047		
5125	023126	044103	051101	041501		
5126	023134	042524	051522	022456		
5127	023142	100				
5128						
5129	023143	111	050116	052125	MES23:	.ASCII ;INPUT THE 'CONTROL' CHARACTERS.%;
5130	023150	052040	042510	023440		
5131	023156	047503	052116	047522		
5132	023164	023514	041440	040510		
5133	023172	040522	052103	051105		
5134	023200	027123	040045			
5135	023204	053042	020124	044103	MES24:	.ASCII ;%"VT CHARACTER DISPLAY TEST"%;

5136	023212	051101	041501	042524		
5137	023220	020122	044504	050123		
5138	023226	040514	020131	042524		
5139	023234	052123	040042			
5140	023240	053042	020124	044506	MES25: .ASCII	;"VT FIELD CONTROL TEST"Q;
5141	023246	046105	020104	047503		
5142	023254	052116	047522	020114		
5143	023262	042524	052123	040042		
5144	023270	053042	020124	051503	MES26: .ASCII	;"VT CSR PRESET TEST"Q;
5145	023276	020122	051120	051505		
5146	023304	052105	052040	051505		
5147	023312	021124	100			
5148	023315	042	052126	046040	MES27: .ASCII	;"VT LINE FEED TEST"Q;
5149	023322	047111	020105	042506		
5150	023330	042105	052040	051505		
5151	023336	021124	100			
5152	023341	042	052126	042440	MES30: .ASCII	;"VT END OF SCREEN TEST"Q;
5153	023346	042116	047440	020106		
5154	023354	041523	042522	047105		
5155	023362	052040	051505	021124		
5156	023370	100				
5157	023371	042	052126	041040	MES31: .ASCII	;"VT BLANK CONTROL TEST"Q;
5158	023376	040514	045516	041440		
5159	023404	047117	051124	046117		
5160	023412	052040	051505	021124		
5161	023420	100				
5162	023421	042	052126	040440	MES33: .ASCII	;"VT ALIGNMENT TEST"Q;
5163	023426	044514	047107	042515		
5164	023434	052116	052040	051505		
5165	023442	021124	100			
5166	023445	042	052126	043040	MES34: .ASCII	;"VT FOCUS TEST"Q;
5167	023452	041517	051525	052040		
5168	023460	051505	021124	100		
5169	023465	042	052126	053440	MES35: .ASCII	;"VT WORST CASE TEST"Q;
5170	023472	051117	052123	041440		
5171	023500	051501	020105	042524		
5172	023506	052123	040042			
5173	023512	053042	020124	052503	MES36: .ASCII	;"VT CURSOR MOVEMENT TEST"Q;
5174	023520	051522	051117	046440		
5175	023526	053117	046505	047105		
5176	023534	020124	042524	052123		
5177	023542	040042				
5178						
5179	023544	020040	047111	020040	MES37: .ASCII	; IN Q;
5180	023552	020040	100			
5181						
5182	023555	040	052517	020124	MES38: .ASCII	; OUT Q;
5183	023562	020040	040040			
5184						
5185	023566	041042	046117	020104	MES39: .ASCII	;"BOLD PRESET TEST"Q;
5186	023574	051120	051505	052105		
5187	023602	052040	051505	021124		
5188	023610	100				
5189						
5190	023611	042	046102	047111	MES40: .ASCII	;"BLINK PRESET TEST"Q;
5191	023616	020113	051120	051505		

```

5192 023624 052105 052040 051505
5193 023632 021124 100
5194
5195 023635 042 046102 047101 MES41: .ASCII ;"BLANK PRESET TEST"Q;
5196 023642 020113 051120 051505
5197 023650 052105 052040 051505
5198 023656 021124 100
5199
5200 023661 042 047125 042504 MES42: .ASCII ;"UNDERLINE PRESET TEST"Q;
5201 023666 046122 047111 020105
5202 023674 051120 051505 052105
5203 023702 052040 051505 021124
5204 023710 100
5205 023711 045 052522 047116 MES43: .ASCII ;%RUNNING TUBE QQ;
5206 023716 047111 020107 052524
5207 023724 042502 030040 100
5208
5209 023731 045 052522 047116 MES44: .ASCII ;%RUNNING TUBE Q;
5210 023736 047111 020107 052524
5211 023744 042502 040040
5212
5213 023750 040007 MBELL: .ASCII <?>/Q/
5214 023752 050045 047522 051107 MXORH: .ASCII /%PROGRAM RUN IS ABORTEDQ/
5215 023760 046501 051040 047125
5216 023766 044440 020123 041101
5217 023774 051117 042524 040104
5218 024002 047045 046525 042502 MED1: .ASCII /%NUMBER OF TUBES? Q/
5219 024010 020122 043117 052040
5220 024016 041125 051505 020077
5221 024024 100
5222 024026 .EVEN
5223 024026 053045 020124 100 MUTP: .ASCII /%VT Q/
5224 024033 045 045506 040 MFKP: .ASCII /%FK /
5225 024037 040 042526 052103 MAUFK: .ASCII / VECTOR AND ADDRESS FOR TUBE Q/
5226 024044 051117 040440 042116
5227 024052 040440 042104 042522
5228 024060 051523 043040 051117
5229 024066 052040 041125 020105
5230 024074 100
5231 024075 045 042040 030514 MDLVA: .ASCII /% DL11 VECTOR FOR TUBE Q/
5232 024102 020061 042526 052103
5233 024110 051117 043040 051117
5234 024116 052040 041125 020105
5235 024124 100
5236 024126 .EVEN
5237 ;ADDRESS AND CONSTANTS TABLE
5238
5239 024126 000000 PIMP: 0
5240 024130 001726 AVECTR: INITB ;CNTR 'A' VECTOR ADDRESS
5241 024132 001000 STACK: 1000 ;'SP' INITIALIZATION ADDRESS
5242 024134 000000 UNITFG: 0 ;SOFTWARE SW., SET IF BOTH UNITS AVAIL.
5243 024136 000000 UNITNO: 0 ;CONTAINS NO. OF UNITS AVAILABLE
5244 024140 000000 MEMSIZ: 0 ;CONTAINS HIGHEST 'K' OF MEM. ADDR.
5245 024142 000000 TSTNUM: 0 ;CONTAINS CURRENT LOGIC TEST NO.
5246 024144 000000 FIELDSW: 0 ;FIELD CONTROL SOFTWARE SW.
5247 024146 000000 CHRCNT: 0 ;'VT' BUFFER CHARACTER COUNTER

```


5304	024344	000000	INTRNO: 0	;SET ON DL11 TRANS. INITIALIZATION
5305	024346	000000	INTRN1: 0	
5306	024350	000000	INTRN2: 0	
5307	024352	000000	INTRN3: 0	
5308	024354	000000	RESTR0: 0	;CONTAINS 1ST FREE LOCATION AFTER
5309	024356	000000	RESTR1: 0	;THE USER DATA
5310	024360	000000	RESTR2: 0	
5311	024362	000000	RESTR3: 0	
5312	024364	000000	XFER0: 0	
5313	024366	000000	XFER1: 0	
5314	024370	000000	XFER2: 0	
5315	024372	000000	XFER3: 0	
5316	024374	000000	BUSY0: 0	;SET IF DOING CONTINUOUS TRANSFERS
5317	024376	000000	BUSY1: 0	
5318	024400	000000	BUSY2: 0	
5319	024402	000000	BUSY3: 0	
5320	024404	000000	BUFOR0: 0	
5321	024406	000000	BUF1R0: 0	
5322	024410	000000	BUF2R0: 0	
5323	024412	000000	BUF3R0: 0	
5324	024414	000000	BUF0R1: 0	
5325	024416	000000	BUF1R1: 0	
5326	024420	000000	BUF2R1: 0	
5327	024422	000000	BUF3R1: 0	
5328	024424	000000	START: 0	
5329	024426	000000	DISPSW: 0	
5330	024430	000000	SELECT: 0	
5331	024432	000000	SHFTSW: 0	
5332	024434	000000	SHFPRT: 0	
5333	024436	000000	SAVOLD: 0	; TEMPORARY STORAGE
5334	024440	000000	TEMP1: 0	; TEMPORARY STORAGE
5335	024442	000000	TEMP2: 0	; TEMPORARY STORAGE
5336	024444	000000	MESPRT: 0	; SOFTWARE PRINT INHIBIT SW.
5337	024446	000000	MESPT1: 0	
5338	024450	000000	SAVEPC: 0	; TEMPORARY STORAGE OF 'PC'
5339	024452	000000	SAVPSW: 0	; TEMPORARY STORAGE OF 'PSW'
5340	024454	000000	SAV2PC: 0	
5341	024456	000000	SAV2SW: 0	
5342	024460	000000	KSTOR1: 0	; PERMANENT STORAGE
5343	024462	000000	KSTOR2: 0	; PERMANENT STORAGE
5344	024464	000000	KSTOR3: 0	; PERMANENT STORAGE
5345	024466	000000	KSTOR4: 0	; PERMANENT STORAGE
5346	024470	000000	KSTOR5: 0	
5347	024472	000000	KSTOR6: 0	
5348				
5349				
5350				
5351				
5352				
5353	024474	031		
5354	024475	031		
5355				
5356				
5357		000001		

VTBUFF: .BYTE EOS
.BYTE EOS
.END

; HERE STARTS A '1200' WORD 'VT' BUFFER USED FOR ALL
; VT DISPLAY TESTS. DATA TO BE DISPLAYED IS ASSEMBLED
; IN EACH INDIVIDUAL TEST AND STORED HERE.

DELAY = 104004	1250#	2269	2363	2380	2405	2430	2433	2505					
DELAYL = 104020	1262#	2340	3090	3091	4064	4075	4384						
DISPLA = 104000	1246#	2264	2286	2313	2316	2320	2324	2328	2351	2396	2422	2446	2462
	2477	2495	2522	2544	2784	2803	2822	2842	2857	2913	2953	2961	3008
	4107												
DISPMS 011610	3112	3116	3140	3157	3234#	3237							
DISPSW 024426	4017	4383*	4385*	5329#									
DL11A 010726	3123	3127#											
DL11B 010750	3127	3130#											
DL11C 011076	3144	3147#											
DL11D 011230	3161	3164#											
DONESW 024162	1652*	3818*	4063*	4070	5253#								
EMTOK 001220	1279	1281#											
EMTSRV 001200	1230	1275#											
EMTTAB 001240	1283	1289#											
ENDCUR 006132	2507	2512#											
ENDEOS 005654	2407	2411#											
ENDTST = 104006	1252#	2192	2273	2332	2383	2411	2436	2512	2844	2894	2974		
EOL = 000012	1449#												
EOP = 000014	1450#	3478	3540	3588	3629								
EOS = 000031	1451#	3236	3420	3451	3492	3506	3512	3625	3643	3694	3699	4048	5353
	5354												
EOSBUF = 104005	1251#	2266	2288	2315	2353	3092	4373						
ERFLG0 024334	3188	3416*	3462*	3636*	3655	3661*	5300#						
ERFLG1 024336	3200	5301#											
ERFLG2 024340	3214	5302#											
ERFLG3 024342	3228	5303#											
ERRMES 020166	4475*	4495*	4501*	4544#									
ERROR = 104400	1245#	1768	1779	1790	1802	1814	1820	1831	1844	1858	1870	1881	1897
	1910	1922	1937	1953	1965	1985	2004	2028	2031	2045	2061	2077	2091
	2105	2120	2135	2140	2156	2181	2248	2557	2569	2581	2592	2604	2618
	2633	2645	2657	2676	2692	2708	2726	2730	2739	2743	2754	2765	2769
	2795	2814	2833										
ERTRAP 016456	1225	4203#											
EXITFO 007544	2875	2884	2889#										
EXITF1 007564	2873	2893#											
EXITSF 005160	2233	2252#											
EXREPT 010334	3021	3042#											
EXTCHR 010036	2928	2958	2960	2965#									
EXTPRT 020252	4538	4564#											
EXTTY 014552	3802#												
EXTVTR 020112	4507	4526#											
FCSET 001620	1457#	1500*	1552*	3941									
FCSET1 001622	1458#	3929	3932*	3940*	3965*	3974*							
FIELDS 024144	2916#	2932	2939	2941*	2977	2979*	3011*	3031	3033*	3036	3038*	5246#	
FKCHAR 007576	1715	2912#	2955	2963	2975								
FKCSR 001474	1394#	2536	2870	2891*	2893*	2927	2968*	3020	3042*	4262*	4269*	4433	4477
	4540*												
FKCSR4=%000004	1207#	2536*	2555	2564*	2565	2570*	2577*	2578*	2579	2582*	2588*	2590	2602
	2605#	2613*	2616	2619*	2626*	2627	2629*	2630*	2631	2639*	2640	2643	2671*
	2687#	2703*	2736*	2737	2740*	2741	2750*	2752	2755*	2762*	2763	2767	2770*
	2771#	2783*	2793	2796*	2812	2815*	2821*	2831	2834*	2835*			
FKDATA 001476	1395#	2751	2766	2872	2929	3022	4434						
FKERR 017446	1232	2539	4423#										
FKFUN 007416	1713	2856#	2895										
FKINT 001500	1396#	1512*	1523*	1571*	1588*	1675	1676*	1677	1679*	2667*	2683*	2699*	4259*

FKLDA	001472	4270*							
FKLDB	001470	1393#	2877	2889*					
FKLOGI	006236	1392#	2883	2890*	4189				
FKLVL	001502	1711	2529	2544#	2847				
FKREPT	010126	1397#	2668*	2684*	2700*	4193	4261*	4270	4271*
FKRPTA	010144	1717	2995#	3010					
FKRPTB	010160	2996	3001#						
FKT1	006246	3000	3005#	3007					
FKT10	006504	2540	2552#	4598					
FKT11	006532	2638#							
FKT12	006566	2651#							
FKT13	006640	2666#							
FKT14	006714	2682#							
FKT15	006770	2698#							
FKT16	007040	2715#							
FKT17	007072	2727	2735#						
FKT2	006262	2749#							
FKT20	007120	2556	2563#						
FKT21	007166	2761#							
FKT22	007254	2779#							
FKT23	007320	2802#							
FKT24	007372	2820#							
FKT3	006314	2781	2841#						
FKT4	006342	2576#							
FKT5	006362	2587#							
FKT6	006412	2598#							
FKT7	006446	2611#							
FKOCSR	001334	2625#							
FKODAT	001336	1324#	3174*	3426	3439	3513*			
FKOINT	001340	1325#	3443						
FKOLDA	001332	1326#	3106*						
FKOLDB	001330	1323#							
FKOLVL	001342	1322#	1510	1563	4178				
FK1CSR	001350	1327#	3107*						
FK1DAT	001352	1333#	3175*						
FK1INT	001354	1334#							
FK1LDA	001346	1335#	3108*						
FK1LDB	001344	1332#							
FK1LVL	001356	1331#	4181						
FK2CSR	001364	1336#	3109*						
FK2DAT	001366	1342#	3165*						
FK2INT	001370	1343#							
FK2LDA	001362	1344#	3153*						
FK2LDB	001360	1341#							
FK2LVL	001372	1340#	4184						
FK3CSR	001400	1345#	3154*						
FK3DAT	001402	1351#	3148*						
FK3INT	001404	1352#							
FK3LDA	001376	1353#	3136*						
FK3LDB	001374	1350#							
FK3LVL	001406	1349#	4187						
FLGERR	012412	1354#	3137*						
FREPT0	010234	3416#	3441	3527	3595	3603	3608	3611	
FREPT1	010260	3015	3020#						
FREPT2	010330	3024	3028#						
		3032	3035	3037	3040#				

RECVDD	014004	3656	3660	3663#															
RECVEE	014010	3658	3664#																
RECVIT	013362	3289	3294	3300	3306	3577#													
RECVON	013444	3581	3587	3591#															
RECVQ	011766	3098	3288#																
RECVQA	013472	3592	3594	3599#															
RECVQB	013540	3600	3605	3610	3613#														
RECVQC	013676	3633	3642#	3670															
RECVQD	013706	3589	3630	3645#															
RECVQE	014016	3618	3627	3650	3665#														
RECV1	012000	3100	3293#																
RECV2	012012	3149	3299#																
RECV3	012024	3132	3305#																
RELOAD	012556	3447	3450#	3484	3653														
REPORT	005126	2235	2243#																
REPTER	010264	3027	3030#																
REPTOK	010312	3026	3029	3036#															
RESCNT	005472	2355#	2362																
RESTR0	024354	3453#	3474	3476	3493#	3654	5308#												
RESTR1	024356	5309#																	
RESTR2	024360	5310#																	
RESTR3	024362	5311#																	
RETURN	020406	1742*	2263*	2285*	2312*	2350*	2395*	2421*	2445*	2461*	2476*	2494*	2521*	2540*					
RINT0	001540	2782*	2856*	2912*	3007*	3073*	3728*	4109	4588	4593*	4598#	4611							
RINT1	001544	1416#	1531	1596	3098*														
RINT2	001550	1418#	3100*																
RINT3	001554	1420#	3149*																
RLVLO	001542	1422#	3132*																
RLVL1	001546	1417#	3099*																
RLVL2	001546	1419#	3101*																
RLVL3	001552	1421#	3150*																
RUNEM	016000	1423#	3133*																
RUNEXT	016006	4067	4069	4075#															
RUNSWH	024150	4077#																	
RO	=%000000	1650*	1730*	4066	5248#														
		1196#	1509*	1517*	1521*	1528*	1530*	1539*	1564*	1575*	1578*	1592*	1595*	1610*					
		2089*	2118*	2152*	2153*	2167*	2175	2176	2206*	2215*	2223	2228*	2230	2243					
		2249	2301	2355*	2357*	2372*	2373	2375*	2376	2377*	2402*	2403*	2408*	2409*					
		2427*	2431*	2432*	2501*	2502	2503*	2508*	2509	2510*	2565*	2566*	2567	2599*					
		2600*	2612*	2614*	2652*	2655*	2669*	2672*	2685*	2688*	2701*	2704*	2707*	2791*					
		2806*	2810*	2825*	2829*	2859*	2879	2880*	2882*	2889	2942*	2948*	2980*	3034*					
		3039*	3041*	3094*	3095*	3096	3110*	3111	3114	3117*	3118	3119	3178*	3181*					
		3184*	3187*	3190*	3193*	3196*	3199*	3204*	3207*	3210*	3213*	3218*	3221*	3224*					
		3227*	3312	3317*	3327	3332*	3342	3347*	3357	3362*	3371	3377*	3382	3388*					
		3393	3399*	3404	3410*	3417*	3418*	3419*	3420*	3450*	3451*	3476*	3478*	3492*					
		3493	3506*	3511*	3512*	3580	3586*	3623*	3624*	3625*	3638*	3639*	3640*	3642*					
		3643*	3654*	3693*	3694*	3698*	3699*	3913*	3922*	3931*	3934*	4035*	4046*	4048*					
		4051*	4053	4084	4102*	4156*	4157*	4160*	4161*	4162	4178*	4181*	4184*	4187*					
		4191	4243*	4246*	4300*	4302*	4304*	4306*	4308*	4389*	4391*	4428*	4452	4454*					
		4496																	
R1	=%000001	1197#	1489*	1490	1492*	1493	1669	1670*	1673*	1678*	1765*	1766*	1776*	1777*					
		1786*	1787*	1788	1816*	1817*	1818	2166*	2178*	2220*	2221*	2222*	2234*	2236					
		2238*	2240*	2241	2354*	2360*	2361	2401*	2406*	2428*	2434*	2500*	2506*	2716*					
		2719*	2724	2728	2788*	2789*	2807*	2808*	2826*	2827*	2860*	2885	2886*	2888*					
		2890	2915*	2936	2937	2951	2959	2965	2969	3022*	3025	3028	3041	3234*					
		3235	3236	3313	3318*	3328	3333*	3343	3348*	3358	3363*	3372	3376*	3383					

SCOPEC	020272	1290	4576#											
SCOPEF	020404	1747*	3727*	4429	4455	4551	4583	4585*	4589*	4597#				
SCOPEG	020350	4582	4584	4589#										
SCOPI	020312	4581#	4604											
SELECT	024430	2074*	2086*	2102*	2115*	2216	5330#							
SERFK0	011626	3106	3241#											
SERFK1	011642	3108	3247#											
SERFK2	011656	3153	3253#											
SERFK3	011672	3136	3259#											
SERVAA	012664	3445	3468#											
SERVAB	012706	3474#	3491											
SERVBB	012770	3477	3479	3492#	3663									
SERVBC	013030	3499#	3535											
SERVCC	013064	3502	3508#	3555	3565	3573								
SERVCE	011326	3178#	3203	3217	3230									
SERVDX	013116	3509	3515#											
SERVFK	012444	3242	3248	3254	3260	3426#								
SERVOA	012530	3440	3443#											
SERVOB	012730	3469	3481#											
SERVOC	013036	3490	3501#											
SERVOD	013106	3435	3449	3467	3471	3473	3488	3500	3504	3507	3513#	3517		
SERVOD	012514	3427	3439#											
SETBFO	012036	3241	3264	3288	3311#									
SETBF1	012102	3247	3270	3293	3326#									
SETBF2	012152	3253	3276	3299	3341#									
SETBF3	012222	3259	3282	3305	3356#									
SETED	002700	1516	1527	1573	1590	1669#								
SETFKA	016442	4191#	4194											
SETLNE=	104007	1253#	2300											
SETUPF	006174	2536#	2546	3719	3723									
SETUPV	003026	1736#	1755	3721										
SETUTO	016360	4170	4177#											
SETUT1	016376	4172	4181#											
SETUT2	016410	4174	4184#											
SETUT3	016422	4176	4187#											
SHFBAD	005074	2226	2234#											
SHFCUR	006106	2505#	2511											
SHFEND	021024	2056	4887#											
SHFEOL	005536	2373#	2379											
SHFLST	005032	2223#												
SHFMES	020026	2073*	2085*	2101*	2114*	4513#								
SHFPRT	024434	1746*	5332#											
SHFTBF	020726	2022	2024	2039	2041	2167	2168	2202	2203	2205	4825#			
SHFTSW	024432	1745*	2129*	2207*	4459	4489	4508	5331#						
SHF32	005046	2227*	2250											
SHF64	005060	2224	2230#											
SHIFT	005004	2076	2090	2104	2119	2215#	2229	2242	2251					
SP	=%000006	1202#	1275*	1276*	1277*	1278	1281*	1282*	1283*	1284*	1285	1482*	1639*	2136*
		2157*	2182*	2252*	3311	3312*	3313*	3314*	3315*	3316*	3326	3327*	3328*	3329*
		3330*	3331*	3341	3342*	3343*	3344*	3345*	3346*	3356	3357*	3358*	3359*	3360*
		3361*	3373	3374	3375	3376	3377	3384	3385	3386	3387	3388	3395	3396
		3397	3398	3399	3406	3407	3408	3409	3410	3418	3421*	3855	3856*	3881
		3882*	3923*	3927*	4077*	4084*	4085*	4086*	4087*	4088*	4089*	4090*	4091	4095*
		4096	4097	4098	4099	4100	4101	4102	4115	4116	4117	4118	4119*	4120*
		4121*	4122*	4123*	4124*	4125*	4126*	4127*	4133	4134	4135	4136	4137	4138
		4139	4140	4141	4142*	4143*	4144*	4145*	4154	4155*	4206	4208	4259	4260*

FKTX	1243#	2563	2576	2587	2598	2611	2625	2638	2651	2666	2682	2698	2715	2735	2749
	2761	2779	2802	2820	2841										
TA	1243#	1774	1784	1797	1809	1826	1839	1852	1864	1875	1890	1903	1915	1929	1944
	1959	1972	1979	1991	1997	2013	2021	2038	2055	2070	2099	2127	2147	2165	2189
TS	1242#	2563	2576	2587	2598	2611	2625	2638	2651	2666	2682	2698	2715	2735	2749
	2761	2779	2802	2820	2841										
VTX	1243#	1774	1784	1797	1809	1826	1839	1852	1864	1875	1890	1903	1915	1929	1944
	1959	1972	1979	1991	1997	2013	2021	2038	2055	2070	2099	2127	2147	2165	2189

ADD	1283	1672	1676	1679	2252	3421	3579	3724	3726	3856	3882	3923	3927	4077	4155
	4236	4260	4279	4349	4357	4377	4592								
ASL	1281	1662	4361	4362	4363	4521									
ASR	4343	4344	4345												
BE2	1560	1562	1572	1589	1614	1627	1751	1767	1778	1789	1801	1813	1819	1830	1843
	1857	1869	1880	1896	1909	1921	1936	1952	1964	1984	2003	2030	2044	2233	2242
	2244	2246	2250	2407	2507	2556	2568	2580	2591	2603	2628	2632	2641	2644	2742
	2873	2878	2884	2933	2940	2944	2946	2970	2996	3026	3029	3037	3087	3169	3433
	3447	3471	3475	3482	3484	3486	3504	3533	3589	3602	3607	3616	3618	3630	3633
	3650	3652	3656	3660	3710	3750	3791	3797	3804	3826	3838	3865	3925	3930	3946
	3948	3950	3952	3954	3956	3959	3960	3962	3964	3967	3969	3971	4170	4172	4174
	4176	4316	4331	4333	4347	4355	4388	4490	4509	4517	4535	4565	4608		
BGE	3689														
BGT	1664	3535	3581	3741	4359	4563									
BHI	3720														
BHIS	2729	3509													
BIC	1282	1504	1551	1661	1766	1777	1787	1817	2221	2240	2566	2629	2880	2881	2886
	2887	3005	3006	3531	3560	3583	3614	3679	3764	4229	4290	4356	4500	4518	4520
	4540														
BICB	3691	3890													
BIS	2234	2238	2564	2577	2588	2613	2626	2630	2639	2736	2740	2750	2762	2771	2783
	2835	2882	2888	2891	2893	2947	2968	2976	3030	3042	3513	3563	3891	4262	4281
	4425	4522	4594												
BISB	3692	3831	4245												
BIT	2026	2029	2043	2046	2059	2236	2245	2579	2602	2616	2627	2631	2640	2741	2780
	2930	3561	3593	3601	3606	3669	3839	4019	4021	4068	4506	4516	4537	4577	4581
	4607														
BITB	3892														
BLE	3944														
BLOS	2725														
BLT	3203	3217	3536												
BMI	1506	2139	2738	2764	2794	2832	3427	3658	3766	3768	3772				
BNE	1279	1497	1518	1529	1540	1545	1576	1593	1611	1622	1642	1654	1674	1895	1939
	2007	2027	2047	2060	2154	2174	2179	2224	2226	2231	2237	2303	2359	2362	2374
	2379	2382	2435	2601	2615	2617	2656	2673	2689	2705	2720	2781	2790	2792	2809
	2811	2828	2830	2875	2931	2938	2950	2952	2958	2960	2966	2978	3024	3032	3079
	3097	3131	3237	3429	3431	3445	3469	3473	3488	3490	3502	3522	3526	3541	3562
	3585	3592	3594	3620	3622	3635	3670	3697	3780	3783	3789	3799	3806	3820	3840
	3860	3897	3912	3920	3937	3942	3984	4018	4020	4022	4024	4026	4037	4050	4067
	4069	4073	4106	4159	4194	4212	4239	4242	4352	4381	4407	4424	4448	4460	4492
	4507	4538	4542	4578	4580	4582									
BPL	2753	2768	2813	2871	2928	2935	3021	3440	3600	3738	3752	3762	3774	3858	3870
	3893	4235	4401	4409	4414	4427	4436	4450	4584						
BR	1479	1491	1543	1657	1668	2078	2106	2155	2180	2193	2194	2229	2235	2239	2251
	2268	2272	2274	2294	2295	2298	2333	2334	2339	2384	2410	2412	2437	2454	2469
	2484	2511	2513	2528	2529	2658	2674	2690	2706	2727	2845	2865	2895	2923	2975
	2981	3000	3017	3027	3035	3084	3166	3230	3290	3295	3301	3307	3443	3467	3477
	3479	3491	3500	3507	3517	3555	3565	3573	3587	3605	3610	3627	3641	3671	3684
	3715	3722	3776	3795	3810	3867	3873	3885	3933	3939	4013	4180	4183	4186	4337
	4364	4390	4393	4463	4499	4556	4604								
CLR	1480	1620	1623	1640	1643	1650	1651	1652	1738	1739	1745	1746	1747	1791	1803
	1815	1832	1833	1845	1846	1855	1882	1883	1918	1966	1986	2005	2032	2050	2062
	2129	2172	2206	2267	2290	2336	2354	2524	2537	2578	2670	2716	2770	2774	2786
	2788	2805	2807	2824	2826	2834	2859	2860	2861	2876	2916	2917	2918	2919	2941
	3011	3012	3038	3095	3176	3319	3320	3452	3453	3454	3455	3456	3457	3458	3459
	3460	3461	3462	3463	3464	3465	3466	3515	3516	3538	3542	3543	3646	3647	3648

CLRB	3661	3662	3727	3742	3759	3760	3818	3822	3932	3940	3974	4014	4052	4104	4152
CMP	4177	4228	4263	4271	4282	4292	4318	4350	4372	4385	4399	4404	4410	4494	4589
CMPB	1561	1621	1641	1663	1788	1800	1812	1829	1842	1879	1895	1908	1935	1963	1983
COM	2002	2223	2225	2230	2232	2241	2243	2249	2361	2567	2724	2728	2872	2874	2943
COMB	2951	2959	2965	2969	3078	3086	3096	3130	3168	3202	3216	3508	3532	3580	3584
DEC	3615	3719	3771	3936	3943	4171	4173	4175	4193	4330	4332	4354	4583	4586	4605
EMT	2373	2937	3025	3028	3236	3444	3468	3481	3483	3485	3489	3501	3540	3588	3621
HALT	3629	3632	3765	3767	3779	3782	3788	3796	3798	3803	3859	3864	3945	3947	3949
INC	3951	3953	3955	3957	3959	3961	3963	3966	3968	3970	4169	4380	4387		
INC B	3040	3470	3965												
JMP	1517	1528	1539	1575	1592	1610	1673	2178	2358	2378	2381	2406	2434	2506	3505
JSR	3534	3688	3696	3740	3792	3896	4049	4158	4346	4358	4406	4562			
MOV	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260
	1261	1262	1263	1264	1265	1266									
	1280	1498	1501	1511	1514	1522	1525	1533	1541	1655	3001	3003	3716	4093	4213
	4437	4465	4568												
	1624	1644	1730	2089	2118	2153	2173	2215	2302	2360	2451	2467	2482	2526	2600
	2614	2655	2672	2688	2704	2719	2789	2791	2808	2810	2827	2829	2869	2956	2964
	2967	2979	3033	3081	3083	3088	3089	3416	3448	3510	3523	3544	3559	3626	3631
	3636	3637	3645	3675	3676	3677	3770	3823	3918	3919	3924	3935	3973	3983	4023
	4025	4036	4063	4105	4233	4238	4383	4430	4456	4493	4536	4545	4552	4585	
	1574	1591	1609	4400	4408										
	1239	1240	1241	1285	1507	1665	1731	2195	2846	2847	3243	3249	3255	3261	3266
	3272	3278	3284	3322	3337	3352	3367	3435	3436	3437	3653	3663	3729	3781	3787
	3807	3975	4109	4222	4340	4480	4527	4588	4611						
	1516	1527	1573	1590	1625	1645	1647	1648	1755	2075	2076	2087	2090	2103	2104
	2116	2119	2131	2137	2149	2159	2170	2183	2227	2270	2292	2297	2319	2323	2327
	2331	2337	2546	2653	2660	2677	2693	2709	2717	2723	2862	2920	2972	3013	3014
	3112	3116	3140	3157	3241	3242	3247	3248	3253	3254	3259	3260	3264	3265	3270
	3271	3276	3277	3282	3283	3288	3289	3293	3294	3299	3300	3305	3306	3434	3441
	3527	3595	3603	3608	3611	3721	3723	3824	3853	3879	3981	4015	4065	4071	4076
	4237	4328	4371	4403	4405	4479	4505	4511	4515	4524					
	1275	1277	1284	1478	1481	1482	1488	1489	1493	1494	1495	1500	1503	1509	1510
	1512	1515	1520	1521	1523	1526	1530	1531	1532	1534	1535	1536	1537	1538	1542
	1552	1556	1563	1564	1571	1577	1578	1588	1595	1596	1597	1605	1606	1607	1608
	1618	1619	1639	1646	1649	1656	1669	1670	1671	1675	1677	1678	1736	1737	1740
	1741	1742	1743	1744	1763	1765	1776	1785	1786	1798	1810	1816	1827	1828	1840
	1841	1853	1854	1865	1867	1876	1877	1878	1891	1892	1893	1904	1905	1906	1916
	1917	1930	1931	1932	1933	1945	1946	1947	1950	1960	1962	1980	1982	1998	1999
	2001	2022	2024	2025	2039	2041	2042	2056	2058	2072	2073	2074	2085	2086	2101
	2102	2114	2115	2128	2130	2136	2148	2151	2152	2157	2166	2167	2168	2175	2182
	2202	2203	2205	2207	2216	2218	2220	2228	2263	2285	2289	2291	2301	2312	2318
	2322	2326	2330	2350	2356	2370	2371	2372	2395	2401	2402	2404	2408	2421	2428
	2429	2445	2461	2476	2494	2500	2501	2502	2503	2504	2508	2509	2510	2521	2525
	2536	2538	2539	2540	2541	2565	2570	2582	2599	2605	2612	2619	2652	2667	2668
	2669	2671	2683	2684	2685	2686	2687	2699	2700	2701	2702	2703	2755	2782	2787
	2796	2806	2815	2821	2825	2856	2877	2879	2883	2885	2889	2890	2912	2915	2955
	2963	3007	3010	3073	3075	3076	3077	3080	3082	3093	3094	3098	3099	3100	3101
	3102	3103	3104	3105	3106	3107	3108	3109	3110	3111	3113	3114	3115	3117	3118
	3119	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3132	3133	3134	3135
	3136	3137	3138	3139	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
	3152	3153	3154	3155	3156	3158	3159	3160	3161	3162	3163	3164	3165	3170	3171
	3172	3173	3174	3175	3178	3179	3181	3182	3184	3185	3187	3188	3190	3191	3193
	3194	3196	3197	3199	3200	3204	3205	3207	3208	3210	3211	3213	3214	3218	3219

	3221	3222	3224	3225	3227	3228	3234	3311	3312	3313	3314	3315	3316	3317	3318
	3321	3326	3327	3328	3329	3330	3331	3332	3333	3334	3335	3336	3341	3342	3343
	3344	3345	3346	3347	3348	3349	3350	3351	3356	3357	3358	3359	3360	3361	3362
	3363	3364	3365	3366	3371	3372	3373	3374	3375	3376	3377	3382	3383	3384	3385
	3386	3387	3388	3393	3394	3395	3396	3397	3398	3399	3404	3405	3406	3407	3408
	3409	3410	3450	3476	3493	3494	3495	3496	3498	3530	3545	3553	3554	3564	3577
	3578	3582	3586	3613	3654	3664	3681	3682	3683	3695	3714	3717	3725	3728	3739
	3758	3763	3769	3793	3855	3868	3871	3881	3883	3884	3895	3913	3914	3915	3916
	3917	3921	3938	3982	4012	4016	4033	4034	4046	4047	4051	4053	4054	4070	4084
	4085	4086	4087	4088	4089	4090	4091	4092	4094	4095	4096	4097	4098	4099	4100
	4101	4102	4115	4116	4117	4118	4119	4120	4121	4122	4123	4124	4125	4126	4127
	4133	4134	4135	4136	4137	4138	4139	4140	4141	4142	4143	4144	4145	4153	4154
	4156	4157	4161	4162	4178	4179	4181	4182	4184	4185	4187	4188	4189	4190	4191
	4192	4206	4208	4230	4231	4232	4259	4261	4268	4269	4270	4278	4280	4289	4291
	4317	4327	4329	4348	4374	4375	4379	4402	4428	4429	4431	4433	4434	4452	4453
	4454	4455	4457	4461	4462	4464	4475	4476	4477	4479	4495	4496	4497	4498	4501
	4502	4503	4504	4510	4514	4519	4523	4551	4557	4558	4587	4590	4591	4593	4600
	4601	4603	4606												
MOV B	1565	1579	1598	2355	2375	2377	2403	2409	2427	2431	2432	2929	2942	2948	2980
	3002	3004	3022	3034	3039	3041	3235	3417	3418	3419	3420	3443	3451	3478	3492
	3506	3511	3512	3537	3547	3570	3572	3623	3624	3625	3638	3639	3640	3642	3643
	3678	3690	3693	3694	3698	3699	3775	3830	3866	3889	3894	3922	3931	3934	4035
	4048	4160	4243	4246	4300	4302	4304	4306	4308	4389	4391				
NOP	1483	1484	1485	1486	1487	1499	1502	1513	1524	1761	1762	2048	2092	2121	2271
	2299	2338	2552	2553	4205										
RESET	1637	1764	1775	1866	1948	1973	1992	2014	2071	2100	2141	2158	2554	2589	2642
	3074	4103	4203												
ROL	1884	1938	2006	3685	3686	3687	3886	3887	3888						
RTI	2892	2971	3043	3378	3389	3400	3411	3743	3754	3802	3834	3842	3863	3899	3985
	4027	4039	4056	4074	4078	4128	4146	4163	4240	4309	4334	4386	4411	4438	4467
	4595														
RTS	1680	1748	2208	2247	2253	2304	2341	2542	3238	3422	3514	3524	3529	3539	3546
	3548	3590	3597	3644	3665	3700	3821	3843	3926	3928	3972	4195	4244	4247	4264
	4272	4283	4293	4319	4353	4360	4415	4570							
SUB	1276	1492	4209	4210	4234	4376	4432	4458	4539						
SWAB	2222														
TRAP	1245														
TST	1278	1490	1496	1505	1544	1559	1613	1626	1653	1750	1818	1856	1868	1920	1951
	2088	2117	2138	2176	2217	2555	2590	2643	2751	2763	2766	2767	2831	2932	2934
	2939	2945	2949	2957	2977	2995	3023	3031	3036	3428	3430	3432	3439	3446	3472
	3474	3487	3503	3521	3525	3591	3599	3619	3634	3649	3651	3655	3657	3659	3709
	3749	3790	3794	3819	3825	3837	3911	3929	3941	4017	4066	4072	4211	4241	4315
	4342	4351	4423	4426	4435	4447	4449	4459	4489	4491	4508	4534	4541	4564	4579
	4602														
TSTB	2376	2737	2752	2793	2812	2870	2927	2936	3020	3426	3497	3617	3737	3751	3761
	3773	3805	3857	3869	3872	4392	4413								
WAIT	2864	2922	3016												
.ABS	1169														
.ASCII	4894	4903	4911	4920	4924	4929	4939	4946	4953	4957	4961	4966	4972	4976	4984
	4996	5008	5014	5018	5020	5028	5036	5045	5051	5057	5065	5069	5076	5085	5089
	5094	5100	5101	5103	5106	5109	5114	5121	5129	5135	5140	5144	5148	5152	5157
	5162	5166	5169	5173	5179	5182	5185	5190	5195	5200	5205	5209	5213	5214	5218
	5223	5224	5225	5231											
.BYTE	2399	2400	2425	2426	2449	2450	2465	2466	2480	2481	2498	2499	3844	3989	3990
	3991	3992	3993	3994	3995	3996	3997	3998	3999	4000	4001	4002	4003	4004	4619
	4620	4621	4622	4623	4624	4625	4626	4627	4628	4629	4630	4631	4632	4636	4637

	4638	4639	4640	4641	4642	4643	4644	4645	4646	4647	4648	4649	4650	4654	4655
	4656	4657	4658	4659	4660	4661	4662	4663	4664	4665	4666	4670	4671	4672	4673
	4674	4675	4676	4677	4678	4679	4680	4681	4685	4686	4687	4688	4689	4690	4691
	4692	4693	4694	4695	4696	4697	4698	4702	4703	4704	4705	4706	4707	4708	4709
	4710	4711	4712	4713	4714	4715	4716	4720	4721	4722	4723	4724	4725	4726	4727
	4728	4729	4730	4731	4732	4736	4737	4738	4739	4740	4741	4742	4743	4744	4745
	4746	4747	4751	4752	4753	4754	4755	4756	4757	4758	4759	4760	4761	4762	4763
	4764	4768	4769	4770	4771	4772	4773	4774	4775	4776	4777	4778	4779	4780	4781
	4782	4786	4787	4788	4789	4790	4791	4792	4793	4794	4795	4796	4797	4798	4802
	4803	4804	4805	4806	4807	4808	4809	4810	4811	4812	4813	4825	4826	4827	4828
	4829	4830	4831	4832	4833	4834	4835	4836	4837	4838	4839	4840	4841	4842	4843
	4844	4845	4846	4847	4848	4849	4850	4851	4852	4853	4854	4855	4856	4857	4858
	4859	4860	4861	4862	4863	4864	4865	4866	4867	4868	4869	4870	4871	4872	4873
	4874	4875	4876	4877	4878	4879	4880	4881	4882	4883	4884	4885	4886	4887	4888
	4893	4982	4983	4995	5006	5030	5031	5032	5035	5050	5353	5354			
.ENABL	1169														
.END	5357														
.EVEN	4005	4814	5034	5222	5236										
.LIST	1169	1242	1775	1785	1798	1810	1827	1840	1853	1865	1876	1891	1904	1916	1930
	1945	1960	1973	1980	1992	1998	2014	2022	2039	2056	2071	2100	2128	2148	2166
	2190	2564	2577	2588	2599	2612	2626	2639	2652	2667	2683	2699	2716	2736	2750
	2762	2780	2803	2821	2842										
.MACRO	1242	1243													
.NLIST	1169	1242	1775	1785	1798	1810	1827	1840	1853	1865	1876	1891	1904	1916	1930
	1945	1960	1973	1980	1992	1998	2014	2022	2039	2056	2071	2100	2128	2148	2166
	2190	2564	2577	2588	2599	2612	2626	2639	2652	2667	2683	2699	2716	2736	2750
	2762	2780	2803	2821	2842										
.REM	1														
.REPT	1220														
.TITLE	1170														
.WORD	3549	3666	4253												

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DBVTAD.SEQ/NL:TOC/SOL/CRF/PAGNUM=DSKZ:DBVTAD.P11
RUN-TIME: 5 10 2 SECONDS
RUN-TIME RATIO: 198/18=10.9
CORE USED: 14K (27 PAGES)