

LA36

TERM TEST
CZLAFAO

AH-E100A-MC

COPYRIGHT © 1978

FICHE 1 OF 1

APR 1978

digital

MADE IN USA

The image displays a large grid of 100 small, illegible tables or data sheets arranged in a 10x10 pattern on a dark background. Each individual table within the grid contains text and possibly numerical data, but the resolution is too low to read the specific content. The overall appearance is that of a microfiche or a similar data storage format.

Product code: AC-E099A-MC
Product Name: CZLAFAD LA36 TERM TEST
Date Created: MARCH 1978
Maintainer: DIAGNOSTIC ENGINEERING
Author: RALPH A. SCHAUER

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Copyright © 1978 by Digital Equipment Corporation.

TABLE OF CONTENTS

- 1.0 ABSTRACT
 - 1.1 Functional Description
- 2.0 REQUIREMENTS
 - 2.1 Equipment
 - 2.2 Options
 - 2.3 Related Programs
- 3.0 LOADING & INITIALIZATION
- 4.0 CONTROL & TEST SELECTION
 - 4.1 Switch Register Control
 - 4.2 Console Control
 - 4.3 Commands
 - 4.4 Examples & Hints
- 5.0 SCOPE OF TESTS
 - 5.1 TEST 0
 - 5.2 TEST 1
 - 5.3 TEST 2
 - 5.4 TEST 3
 - 5.5 TEST 4
 - 5.6 TEST 5
- 6.0 ERRORS
- 7.0 PROCEDURES FOR NON STD. DVCS

1.0 ABSTRACT

This diagnostic will functionally test the hardware options on the LA36 terminal.

Up to 48 terminals, including the console device, can be tested at a time. All terminals under test must be interfaced thru a DL11 type interface.

Control of this diagnostic may be thru a switch register, or via interactive console terminal commands.

1.1 Functional Description

The program will first determine if the system has a hardware switch register. If none available a software switch register at loc 176 will be assigned.

The program will then determine what device interfaces are on the system, at the standard DL11-A,B,C,D,E address and the interrupt vector of each interface present. This information will then be stored in a table.

The program will then ask the operator if he/she wants to use console control. If the response is "no" the program will halt and wait for the operator to enter control information thru the switch register.

Note : If no hardware switch register is present control is forced to console control.

If console control is selected a menu of available commands will be printed on the console terminal, and the program will wait for commands via the console.

2.0 REQUIREMENTS

2.1 EQUIPMENT

This diagnostic was written to run on all models of the PDP-11 computer, including the LSI-11. One to forty eight LA36 terminals, connected thru a single line asynchronous interface (DL11-A thru E, DL11-W, DLV11), may be tested. 8K of memory is required.

A hardware switch register is supported, but not required.

2.2 OPTIONS

The following options are tested by this diagnostic:

ASCII/APL character set.
Selective Addressing Option.
Auto-Answerback Option.
Top-of-form Option.
Horizontal & Vertical Tab Option.

2.3 RELATED PROGRAMS

Although some error checking is done, this diagnostic does not functionally test the basic LA36 terminal, nor the terminals interface.

Therefore the basic LA36 terminal diagnostic, MDEC-11-DZLAC-*, and the DL11-*/DLV-11 interface test should be successfully run prior to this diagnostic. Also any other applicable processor/memory tests.

3.0 LOADING & INITIALIZATION

This diagnostic may be loaded using the standard procedures for paper tape, or via XXDP media. This diagnostic will not run in chained mode because manual intervention is required. The absolute loader area is preserved.

This diagnostic self sizes the system as far as the interfaces, and their interrupt vectors. The only operator modifications to be made are:

1. the location 'TIMER' which is a CPU dependent TIME constant.
See listing of DELAYM Routine.

```
TIMER: .WORD 554      :SET FOR 11/35 - 11/40
                   :SET TO 202 IF 11/03
                   :      251 11/05 - 11/10
                   :      314 11/15 - 11/20
                   :      2127 11/45 BIPOLAR - 11.55
                   :      1237 11/45 - MOS - 11/70
                   :      755 11/45
DELAYT: .WORD 0      :DELAY TIME BUFFER
```

2. any of the preset device address to accomodate a non-standard interface address. (See Sect 7.0 for details)
3. location 'WIDTH' if other than 132 COL PAPER is being used.
(See section 4.3). (Common for all terminals)

The initial starting address is 200(B), and all restarts at 1372.

4.0 CONTROL & TEST SELECTION

The diagnostic will ask (via the console) if console control is desired. Answer 'Y' if you want to use interactive commands, otherwise type 'N' for switch register control.

If 'Y' is typed a menu of available commands is printed on the console, and the program will wait for command input. If 'N' is typed the program will print a listing of the line (interface) table, then halt for the operator to set the switches to the desired parameters.

4.1 SWITCH REGISTER CONTROL

When SWITCH REGISTER CONTROL is selected the program will halt. Set the switches to the desired mode, then press continue. The program will check the entry and if a specific test is to be run, or a specific line is to be tested the program will halt again. Enter the desired line/test data in the switches, then press continue. To change parameters the test must be restarted at loc 1372.

ALL of the switch functions are also available under console control mode. (See sect. 4.2.4.3).

SWITCH REGISTER BIT DEFINITIONS
FIRST WORD MODE SELECTION

BIT15	=1 (UP) =0 (DOWN)	HALT ON ERROR CONTINUE AFTER REPORT
BIT14	=1 (UP) =0 (DOWN)	LOOP AFTER ERROR IS DETECTED DON'T LOOP
BIT13	=1 (UP) =0 (DOWN)	INHIBIT ERROR REPORTS PRINT ERROR REPORTS
BIT12	=1 (UP) =0 (DOWN)	PRINT INTERFACE TABLE DON'T PRINT TABLE
BIT11	=1 (UP) =0 (DOWN)	INHIBIT ITERATIONS NORMAL RUN.
BIT6	=1 (UP) =0 (DOWN)	RUN ALL TESTS IN SEQUENCE RUN SELECTED TEST ONLY
BIT5	=1 (UP) =0 (DOWN)	RUN ALL AVAILABLE LINES RUN SINGLE LINE ONLY

SECOND WORD LINE AND TEST NO.

BIT15 - BIT8	SELECTED LINE NUMBER (00-57)
BIT7 - BIT0	SELECTED TEST NUMBER (0-5)

4.2 CONSOLE CONTROL

When console control is selected a menu of available commands is typed on the console terminal. The program will wait for commands to be entered thru the keyboard.

Enter one command per line, followed by a carriage return. To terminate command input and start execution type an ALTmode or ESCape.

To regain control once execution has begun type a CTL-C. The program will respond with 'READY'. You can now enter the desired

commands just as in startup.

4.3 COMMANDS

There are two types of commands available, interactive commands; and static commands.

ALL static commands can only be entered while in "Command Mode", that is during startup after 'READY' is printed on the console, or after the operator has invoked command mode by typing a CTL-C and the program has printed 'READY'.

Interactive commands can be entered at any time, and are essentially the same as the switch register bits 15,14,13.

The available Commands are:

- S (STATIC) Single Line Mode. Test a single device. Line 00 is default. Use add command to select the desired line.
- M (STATIC) Multi Line Mode. Test all lines present and not deselected.
- G (STATIC) Sequence thru all tests, starting with test 0.
- Rn (STATIC) Run test N.
- Dn (STATIC) Deselect or Drop from testing interface line N. (see T command)
- An (STATIC) Add line N. Clear out the error count for line N, and reselect the line for testing. In single line mode sets N as current line to test.
- T (STATIC) Type out a listing of the interface lines present on the system, the vector address at which the device interrupts, and whether or not the line is selected.
- Wn (STATIC) Changes location 'Width' to N. Used to control output of terminal tests. Enter N as an Octal number 32 to 204. (132 decimal)
- CTL-L (interactive) Loop on error. If an error is detected the program will start looping or the test section which caused the error, and continue to loop until a Klear command is issued.
In command mode type an L.

- CTL-H (interactive) Halt on error. Will cause the program to halt after the error message is printed. (assuming error printouts are enabled).
In command mode type an H.
- CTL-K (interactive) Klear - resets both the H and L commands (Don't halt, Don't loop).
In command mode type a C.
- CTL-N (interactive) NO Error reports. Inhibits the printing of normal error messages.
In command mode type an N.
- CTL-P (interactive) Print error reports. ALL report messages will be printed on the console.
In command mode type a P.
- CTL-G (interactive) Will cause the program PCFLAGS to be displayed on the console for trouble shooting purposes. See listing for bit definitions.
- ESC Exit command mode & start execution of the diagnostic test(s).
- CTL-C Returns test to command mode.
All operations in progress stop.

4.4 EXAMPLES & HINTS

Test numbers 0 thru 5 may be selected to run individually on all, or any terminal.

ALL tests can be run sequentially on all or any terminal.

Tests can be run sequentially on a terminal, but there is little chance that any terminal will have more than one or two of these options installed. Sequencing all tests will probably result in numerous errors from trying to test non-existent options.

If a line gets more than 16 errors it will be deselected by the program and a message will be printed on the console. If the line is the only one being tested it will automatically be re-selected.

Example 1. the commands to select test #4 to be run on all lines.

```

READY
RD4(CR) - Run test 4
M(CR)   - Multi line mode
w120(CR) - Set paper width to 120
$       - ESC - echoed as $

```

Example 2. the commands to run all tests on line no. 06, with Halt on error set.

```

READY
Q(CR) - Sequence tests
S(CR) - Single line mode.
AOB(CR) - Add line #06
H(CR) - Halt on error
$ - ESC - echoed as $.

```

EXAMPLE 3. How do I run tests on 10 out of 12 terminals.

First you must be in command mode. If a test is running type CTL-C. The program will respond with "Ready". Now type D nn (cr) to deselect line number nn. Repeat for each additional line to be dropped from the tests. Now select your test other parameters as in Example 1.

EXAMPLE 4. How do I restart a device which has been deselected?

In command mode type Ann (cr) where nn is the line number of the line to be added.

EXAMPLE 5. How do I loop on error.

If the test is running type a CTL-L. When an error is detected the program will start to loop on the test or subtest and continue to loop until a CTL-K or CTL-C command is issued.

If in command mode type L (CR).

EXAMPLE 6. Can I set the 'width' constant different for each terminal?

The constant 'width' is the same for all terminals, although it may be changed any time you are in command mode.

5.0 SCOPE OF TESTS.

This diagnostic will functionally test the following hardware options of the LA36.

1. Secondary character set option APL/ASCII character set selection
2. Selective addressing option
3. Auto-answer back option
4. Top of forms option

5. Horizontal and Vertical tabulation option.

The diagnostic will do cursory testing of the basic interface, and basic LA36 logic. It is assumed that the basic interface, CPU/memory, and LA36 tests have been run successfully.

Due to the nature of the hardware under test most error checking will be by visual inspection of the terminal output by the operator.

Description Of Tests

5.1 Test 0 Secondary Character Set.

This test is an output only test. No terminal feed back is received.

The test prints on each terminal alternating lines of ASCII character set, and APL character set.

Output of characters per line will consist of char codes 40 thru 176 unless the paper width limit is reached first.

Output format:

```
PRIMARY----(ASCII CHARACTER SET)
SECONDARY--(APL CHARACTER SET)
```

This output is controlled by the "width" of the paper. See W command description.

5.2 Test 1 Selective Addressing Option

This test will alternately send out a select code, followed by it's ASCII Equivalent, for all possible select codes (20 thru 177).

This test will also deslect all terminals and try to output a "this should not print" message. "All terminals should be off"

This test will also try to print "this should not print" messages after transmitting 'NUL' select character sequences, and no select codes sequences.

Valid terminal output should be: Select Recognized = /NN(Group Select Code) /NN(Individual Select Code)/

Where NN represents the select codes recognized by this terminal. If the group select code and individual select codes are set to the same thing them only one /NN/ should appear.

More than two /nn/ codes printed indicates a logic failure in the decoder section of the option, or possibly interface to terminal line problems.

Any of the "this should not print" message that appear on the terminal indicate a failure of the selection logic.

5.3 Test 2 Auto Answer Back Option

This test is divided into six subtests:

Subtest -1

The first subtest is actually a sizing routine. The terminal should respond to its unique selection code with an answerback when polled. This test has no way of knowing what the answer back is, nor any way of 'pre-selecting' its unique selection code. Therefore subtest -1 will try all legal selection codes to see if it can cause an answer back to be transmitted. If one is received the program will store the select code associated with the response in the line table for future testing.

Subtest -2

Will see if any answer back has been received, and check its length, the message should not exceed 20 (10) characters. Subtest 2 will print the ASCII message on the terminal, and an octal representation of the characters (to verify non printables are being transmitted correctly, and as a trouble shooting aid if bad data is being sent out from the switches).

Subtest -3

Will read the answer back ten times to verify reliability of the data, and lines.

Subtest -4

Will try to cause transmission of the answer back in response to a broadcast code.

Single Line Mode - Subtests 5, and 6.

Subtest -5

The test will request the operator to press the 'Here-is' key, then check for answerback.

Subtest -6

The test will request the operator to type 'CTL-E', then check for answer back.

The operator must verify that the message echoed back to the terminal is correct, by comparing it to the data switch configuration.

5.4 Test 3 Top Of Forms Option

This test is divided into two subtests, one for multi line mode, the other for single line mode. Operator intervention will be required for the single line test.

1. Multi line mode.

This subtest will assume a standard form of 11 inches being used.

The test will issue a form feed, then print a line of dashes. This FF/dashes is repeated 3 times.

The operator should verify correct operation by checking for a line of dashes at the same place on each page.

2. In single line mode, This test will require the operator to set the forms length switch to the value requested. The test will then do three form feeds at each length setting.

5.5 Test 4 Horizontal Tab Option

This test will adjust it's output to conform to the paper width. Change location "width" to the appropriate value before starting test. (Preset to 132 col.) Note: see W command description.

The test will print a reference line for visual verification. The line will look like this:V.....V.....V.....V.....

Tabs will then be set corresponding to the location of each V. The test will then issue a tab and print an X, tab then X etc until the line is complete. Three lines of X's will be printed. All X's printed should be aligned with the reference line V's.

This will be repeated for various (7) values of tab spacing.

Example of output

```

.....V.....V.....V.....V.....V.....
 X      X      X      X      X
 X      X      X      X      X
 X      X      X      X      X

.....V.....V.....V.....
 X      X      X
 X      X      X
 X      X      X

```

5.6 Test 5 Vertical Tab Option

This test is divided into two subtests, one for multi line mode, the other for single line mode. The single line mode test requires operator intervention.

Multi line mode subtest

Will set tabs at intervals of 1 line, 2 lines, 3 lines etc. up to 11 lines. The test will then issue a vert. tab then print a line of dashes, then repeat until 1 full page has been done. Three pages of output are run for 1 pass of test.

Single line mode testing involves the operator to set up the forms control to 11 inch forms, and then proceeds with the same subtest as for multi line mode.

5.0 ERROR REPORTING

There will be four basic sources of error messages. First the system sizer, second the command decoder, and third the diagnostic tests, and the I/O drivers.

5.1 Diagnostic Tests

All test error messages will be 2 lines of output. A standard format line, shown below, and a descriptive message telling what went wrong.

Std. Fmt.: @ERROR XXY TEST YY LINE ZZ

where XXX = the error number local to the current test.

YY = the current test number.

ZZ = the line under test.

an example of a descriptive message :

.NO ANSWERBACK MESSAGE RECEIVED

As each error is handled a routine will update an error count for the failing line. If 16 errors are accumulated on a line, that line will be "deselected" and the following message will be displayed.

EXCESSIVE ERRORS .. LINE XX DROPPED.

If the line under test is the only line being tested the program will automatically re-select the line, zero the count, and continue testing after typing the following:

LINE RE-SELECTED FOR TEST.

6.2 I O Driver

If the IO Driver finds no available line to test a message will be displayed and then control will return to the "ready" state.

NO LINES AVAILABLE FOR TEST.

#377 Multi line driver error.
Error messages tagged as #377 indicate a failure during an I/O driver operation, such as a failure to interrupt on transmit to a terminal with the interrupt enable set.

#376 Same as #377 except a single line Driver.

6.3 Command Decoder

Console terminal command errors will be handled by a CMDERR module & will output a line of "???" if the input was invalid.

If a line selection command tries to add (re-select) an invalid or non-existent line a "--LINE INVALID" message will be typed.

6.4 System Sizer

If during the sizing operation the sizing routine detects a failure of the interface to interrupt it will be reported.

ex: 'NO INTERRUPT ON TXMIT LINE 27'

7.C PROCEDURES FOR NON-STANDARD DEVICES.

This diagnostic can be modified for use on devices that have non-standard interface addresses by replacing a unused address in the line table with the address of the interface line to be tested.

The table is preset to the standard DL11-A,B,C,D,E addresses. (775610 - 776170 & 776500 - 776670), and the console address 777560.

No modification need be made because of non-standard interrupt vector addresses. The diagnostic sizes each address for presence on the system, and inserts the interrupt vector data into the table at run time.

NOTE: The table addresses are not in ascending order, rather it has been optimized for relative system size by having the most commonly used addresses at the head of the table. DL11-A,B

and DLil-C.D.E address are merged together.

101		BASIC DEFINITIONS
216	04200	ACT111 HOOKS
507	08500	TEST CONTROL & INITIALIZATION
713	12540	LINE CONTROL & INITIALIZATION
791	14840	SWITCH REGISTER ROUTINES
1189	22020	CONSOLE TERMINAL ROUTINES
1378	00700	ERROR & REPORT ROUTINES
1452	05450	INTERFACE SIZER ROUTINES
1472	06650	EMT HANDLER
1659	11595	I/O DRIVERS
1793	16900	TRAP ROUTINES
1898	00250	CONVERSION ROUTINES
1910	43150	LA36 OPTION TESTS
1910		STORAGE & CONSTANTS

E02

00200

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

*TITLE CZLAFAC LA36 TERM TST
*COPYRIGHT (C) 1977
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM B. R. SCHAUER
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC S/S' AC
*PACKAGE (MAINDEC-11-C20AC-03), JAN 19, 1977.
*
\$TN=1
\$SWR=160000 ::HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR T PLOT

000001
160000

00260
00280
00300
00320
00340
00360
00380
00400
00420
00440
00460
00480
00500
00520
00540
00560
00580
00600
00620
00640
00660
00680
00700
00720
00740
00760
00780
00800
00820
00840
00860
00880
00900
00920
00940
00950
00955
00960
00980
01000
01020
01040
01060
01080

00260
00280
00300
00320
00340
00360
00380
00400
00420
00440
00460
00480
00500
00520
00540
00560
00580
00600
00620
00640
00660
00680
00700
00720
00740
00760
00780
00800
00820
00840
00860
00880
00900
00920
00940
00950
00955
00960
00980
01000
01020
01040
01060
01080

***** OPERATING INSTRUCTIONS *****

1. THIS TEST ASSUMES THAT THE BASIC INTERFACE LOGIC TESTS & BASIC LA36 FUNCTIONS TEST MDEC-11-DZLAC-X HAVE BEEN RUN SUCCESSFULLY.

TIMING FOR ALL TESTS IS DEPENDENT ON CPU TYPE. THE TIMER IS SET FOR AN 11/40...IF THIS IS NOT THE CASE CHANGE LOCATION "TIMER" ACCORDING TO THE TABLE SUPPLIED IN THIS LISTING.

2. THE DIAGNOSTIC WILL START BY ASKING IF THE OPERATOR WANTS TO USE CONSOLE TERMINAL CONTROL, ANSWER Y OR N.

IF Y IS ENTERED, A "MENU" OF AVAILABLE COMMANDS IS PRINTED ON THE TERMINAL, AND THEN THE PROGRAM WAITS FOR INSTRUCTIONS THRU THE KEYBOARD.

IF N IS ENTERED, THE PROGRAM WILL PRINT A LISTING OF INTERFACES BY LINE NUMBER, THEN HALTS. SET THE SWITCHES TO THE DESIRED MODE AND PRESS CONTINUE. THE PROGRAM WILL DECODE THE SWITCH REGISTER, AND IF RUNNING A SELECTED TEST OR A SELECTED LINE, WILL HALT AGAIN. ENTER THE DESIRED TEST NO. IN THE LOW ORDER BYTE, AND/OR THE DESIRED LINE NO. IN THE HIGH ORDER BYTE. PRESS CONTINUE.

IF NO HARDWARE SWITCH REGISTER IS PRESENT ON THE SYSTEM THE PROGRAM WILL USE LOCATION 176 AS A SOFTWARE SWITCH REGISTER. CONTROL WILL DEFAULT TO THE CONSOLE !

3. TO CHANGE MODES TYPE CTL-C. THE CONSOLE WILL RESPOND WITH READY. ENTER YOUR COMMANDS FOLLOWED BY AN ESC. INTERACTIVE COMMANDS SUCH AS CTL-G MAY BE ENTERED DURING COMMAND, OR RUN MODES.

4. IF MULTI-LINE MODE IS SELECTED, A TABLE OF INTERFACE LINES WILL BE PRINTED. ALL LINES PRESENT WILL BE INITIALLY SELECTED. IF YOU ADD OR DROP LINES USE THE LINE NO. SPECIFIED IN THE LINE TABLE LIST.

BASIC DEFINITIONS

```

15          100000
16          002000
17          001000
18          000200
19          000100
20          000040
21          000020
22          000020
23          100377
24          000004
25          177564
26          177566
27          177560
28          177562
29          000060
30
31          :04000
32          :04002
33          :04004
34          :04006

```

```

03080      MERR = BIT15
03120      NEWTS* = BIT10
03140      NEWMOD = BIT9
03160      READY = BIT7
03180      SEQ = BIT6
03200      MULTI = BITS
03220      SWC.L = BIT4
03240      PRINT = BIT4
03260      MERRN = 100377
03280      MFLAGS = 4
03281      TPS = 177564
03282      TPB = 177566
03283      TKS = 177560
03284      TKB = 177562
03285      TKV = 60
03300      ***** EMT CALL EQUATES
03320      TYPE = EMT
03340      PRTTBL = EMT+2
03360      ITRAP = EMT+4
03380      DELAYR = EMT+6
03460

```

```

181      030000      03500
182 000000 000002 03520
183 000002 000000 03540
184 000004 000006 03560
185 000006 000000 03580
186 000010 000012 03600
187 000012 000000 03620
188 000014 007162 03640
189 000016 000200 03660
190 000020 007416 03680
191 000022 000200 03700
192 000024 001372 03720
193 000026 000000 03740
194 000030 006162 03760
195 000032 000000 03780
206      000172      04000
207 000172 000000 04020
208 000174 000000 04040
209 000176 000000 04060
210 000200 000137 04080
211      001102      04100
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

      . = 0
      . + 2
      HALT
MACHR: . + 2
      HALT
      . + 2
      HAL
      INTRAP
      PRI4
      TXTRAP
      PRI4
      RESTAT
      PRI0
      EMTBOS
      PRI0
      . = 172
SWTEST: . WORD 0
SWLINE: . WORD 0
SSWR: . WORD 0
      JMP START

```

```

: TRAP CATCHER
: BREAKPOINT TRAP
: USED DURING SYSTEM SIZER
: IOT TRAP
: USED BY TXMIT I/O DRIVER
: POWER FAIL TRAPS TO RESTART

```

.SBTTL ACT11 HOOKS

```

: *****
: HOOKS REQUIRED BY ACT11
$SVPC=
      . = 46
      START
      . = 52
      . WORD 20000
      . = $SVPC
      . = 1100
      NOP
      : SAVE PC
      : 1) SET LOC. 46 TO ADDRESS OF START
      : 2) SET LOC. 52 TO 20000
      : RESTORE PC

```

```

216
220
221
222
223
224 001102
225 001102 000005
226
227
228
229
230
231
232 001104 012706 001100
233 001110 005037 016152
234 001114 005037 016154
235 001120 005037 016162
236 001124 005037 016160
237 001130 012737 016114 016164
238 001136 012705 010352
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 001206 012700 017051
266 001212 104000
267 001214 012700 017056
268 001220 104000
269 001222 012700 017103
270 001226 104000
271 001230
272 001230 004737 003332
273 001234
274
275 001234 032737 000020 001364

```

```

.SBTTL TEST CONTROL & INITIALIZATION
*****
START:
  RESET
***** TEST MONITOR *****
:
: PROGRAM INITIALIZATION SECTION
:
:   MOV     #STACK,SP
:   CLR     NEXT
:   CLR     INTEST
:   CLR     NXTLIN
:   CLR     ONLIN
:   MOV     #INBUF, PTR
:   MOV     #TOOBLK, RE
:
: SEE IF SYSTEM HAS A SWITCH REGISTER
:   JSR     PC,SWTST
:
: PRINT TEST IDENTIFICATION MESSAGE
:   MOV     #PROGID,RO
:   TYPE
:
: DETERMIN SYSTEM CONFIGURATION +-
: BUILD A TABLE OF INTERFACE LINES.
:   JSR     PC,BUILD
:
: RESTORE TRAP CATCHER FROM 100 TO 1000
:   JSR     PC,CATCH
:   PRITBL
:
: FIND OUT IF OPERATOR WANTS TO USE
: CONSOLE OR SWITCHES FOR CONTROL
:   JSR     PC,GETSRC
:   JSR     PC,CONSON
:   BIT     #SWCTL,PCFLAG
:   BNE     50001$
:
: PRINT A MENUE OF AVAILABLE COMMANDS
:   MOV     #L3,RO
:   TYPE
:   MOV     #HEADR1,RO
:   TYPE
:   MOV     #COMSUM,RO
:   TYPE
50001$:
START3: JSR     PC,CONSON
50002$:
:   BIT     #SWCTL,PCFLAG

```

```

(9) 001242 001403
276 001244 004737 003130
277 001250 000421
278 (3) 001252
279 05420
280 05440
281 05460
282 05480
283 001252 012700 020123
284 001256 104000 05520
285 001260
286 001260 000001
287 001262 032737 000200 001364 05560
(5) 001270 001001
288 001272 000772
289 (3) 001274
290 05620
291 05640
292 05660
293 001274 032737 010000 001364
294 (9) 001302 001404
295 001304 104002
296 001306 042737 010000 001364 05700
297 001314
298 001314
300 05780
301 05800
302 05820
303 05840
304 001314 004737 002110
305 001320 012700 020411
306 001324 004737 007010 05870
307 001330 004737 001446
308 001334 032737 004000 002032 05872
(5) 001342 001406 05874
309 001344 042737 004000 002032
310 001352 004737 001372
311 001356 000726
312 (3) 001360
313 001360 000137 001172 05990
314 001364
315 06020
316 06040
317 06060
318 06080
319 06100
320 06120
321 06140
322 06160
323 001364 000001
324 001366 000000
001370 000000
06180
06200
06220
06240
06260

```

```

; IF IN SWITCH CONTROL GET CONTENTS OF SW REG.
; PC GETSWS
; 50004$:
; IN CONSOLE CONTROL SIGNIFY READY
; AND READ COMMANDS FROM THE CONSOLE.
;
; MOV #RDY, RD
; TYPE
; 50006$:
; WAIT
; BIT #ATTN, PCFLAG
; BNE 50007$
; BR 50006$
; 50007$:
; PRINT THE LINE TABLE IF REQUESTED.
;
; BIT #PRINTT, PCFLAG
; BEQ 50010$
; BIC #PRINTT, PCFLAG
; 50010$:
; 50005$:
; SET UP THE I/O DRIVER AREAS
; SET UP & EXECUTE REQUESTED TESTS.
;
; JSR PC, LINMON
; MOV #ALLON, RD ; ISSUE A SELECT ALL COMMAND
; JSR PC, MTYPE ; IN CASE THERE ARE SELECTIVE
; ; TERMINALS ON LINE.
;
; JSR PC, TSTCTL
; BIT #EOL, CFLAGS
; BEQ 50003$
; BIC #EOL, CFLAGS
; JSR PC, RESTRT
; BR 50002$
; 50003$:
; JMP START2
; 50000$:
; *****
; * NOTE...TYPING CTL-G WHILE IN CONSOLE *
; * CONTROL MODE WILL CAUSE THE *
; * PCFLAG WORD TO BE PRINTED. *
; *****
; PROGRAM CONTROL BLOCK
; PCFLAG: .WORD 1 ; PROGRAM CONTROL FLAGS
; TESTNO: .WORD 0 ; TESTNO
; LINENO: .WORD 0 ; LINENO

```

```

325 06280
326 06281
327 06282
328 06283
329 06284
330 06285
331 06286
332 06287
333 06288
334 06289
335 06290
336 06291
337 06292
338 06293
339 06294
340 06295
341 06296
342 06297
343 06298
344 06299
345 06300
346 06301
347 06305
348 06306
349 06307
350 06319
351 06320
352 06340
353 06360
354 06380
355 06400
356 06420
357 06440
358 06460
359 06480
360 06500
361 06520
362 06540
363 06560
364 06580

```

```

*****
PCFLAG BIT DEFINITIONS *
*****
BIT 15 HALTOE
BIT 14 LOOPOE
BIT 13 INHRPT
BIT 12 PRINTT
BIT 11 DATAIN
BIT 10 NEWST
BIT 9 NEWMOD
BIT 8 LDONE
BIT 7 ATTN
BIT 6 SEQ
BIT 5 MULTI
BIT 4 SWCTL
BIT 3 DROPC
BIT 2 ADDC
BIT 1 FLAG2
BIT 0 FLAG1

*****
RESTART
*****
RESTR: MOV #STACK, SP
CLR NEXT
CLR INTEST
MOV #1, PCFLAG
CLR NXTLIN
CLR ONLIN
MOV #INBUF, PTR
JSR PC, CATCH
NOP
RESET
JMP START3

```

```

HALT ON ERROR (SW-15)
LOOP ON ERROR (SW-14)
INHIBIT REPORTS (SW-13)
PRINT TABLE (SW-12)
DATA IN FROM KBD.
CHANGE IN TEST NO.
CHANGE IN MODE.
END OF LINE TABLE REACHED
ATTENTION !!!!!!!!!!!
SEQUENCE TESTS MODE
MULTI LINE MODE.
CONTROL VIA SWITCHES.
DROP LINE COMMAND
ADD LINE COMMAND
MODE 0 = NO CURRENT I/O TO CONSOLE
1 = IN COMMAND INPUT MODE
2 = I/O TESTING OF CONSOLE
3 = ?

```

;REINITIALIZE EVERYTING

001364

016164

001230

N02

CZLAFAD LA36 TERM TST MACY11 30A(1052) 03-JAN-77 00:01 PAGE 1-9
 CZLAFAP11 03-JAN-78 11:20 TEST CONTROL & INITIALIZATION

SEG 0026

369					06680	:*****	
370					06700	: TSTCTL THIS SECTION CONTROLS TEST SELECTION, TEST	
371					06720	: SEQUENCING, AND INTERFACES TO ERROR AND REPORT	
372					06740	: MODULES AS REQUIRED BY THE TEST MODULES.	
373					06760	:*****	
374					06780		
375	001446	013737	001366	016152	06800	TSTCTL: MOV TESTNO, NEXT	:GET TEST NO.
376	001454	013737	016152	016154	06820	LOOP1: MOV NEXT, INTEST	:GET CURRENT TEST NO.
377	001462	004737	002056		06840	JSR PC, SUTEST	
378	001466	004777	014464		06920	LOOP2: JSR PC, TESTAD	::START TEST
379					06940		
380					06942		
381					06944	: CHECK FOR ERROR FLAG FROM TEST	
382					06946	:	
383	001472	032737	000020	001364		BIT #SWCTL, PCFLAG	
(9)	001500	001414				BEQ 50011\$	
384	001502	017737	014416	016100		MOV #SWR, TEMP	
385	001510	042737	003777	016100		BIC #3777, TEMP	
386	001516	042737	174000	001364		BIC #174000, PCFLAG	
387	001524	053737	016100	001364		BIS TEMP, PCFLAG	
388	001532					50011\$:	
389					06953		
390	001532	032765	100000	000004		BIT #MERR, MFLAGS(R5)	
(9)	001540	001414				BEQ 50012\$	
391	001542	016537	000004	002032		MOV MFLAGS(R5), CFLAGS	
392	001550	016537	000006	002034		MOV POINT(R5), TSCPTR	
393					07002		
394					07004	: CALL ERROR HANDLER ROUTINE	
395					07006	:	
396	001556	004737	005124			JSR PC, ERROR	
397	001562	042765	100377	000004		BIC #MERRN, MFLAGS(R5)	
398	001570	000421				BR 50013\$	
(3)	001572					50012\$:	
399					07062		
400					07064	: SEE IF TEST IS REPORTING DONE CONDITION	
401					07066	:	
402	001572	032765	020000	000004		BIT #TDONE, MFLAGS(R5)	
(9)	001600	001415				BEQ 50014\$	
403					07082		
404					07084	: UPDATE THE PASS COUNT THEN REPORT END OF PASS	
405					07086	:	
406	001602	005265	000002			INC PASCNT(R5)	
407	001606	042765	020000	000004		BIC #TDONE, MFLAGS(R5)	
408	001614	052737	020000	002032		BIS #EOP, CFLAGS	
409	001622	016537	000002	002036		MOV PASCNT(R5), TSCCNT	
410	001630	004737	005436			JSR PC, REPORT	
411	001634					50014\$:	
412	001634					50013\$:	
413					07222		
414					07224	: IF LOOP ON ERROR IS SET AND AN ERROR IS	
415					07226	: DETECTED THE ERROR HANDLER WILL MAKE THE	
416					07228	: RETURN ADDRESS OF THE TEST ODD	
417					07230	:	
418					07232	: CHECK FOR ODD ADDRESS... IN LOOP MODE...	
419					07234	:	
420	001634	032765	000001	000010		BIT #BIT0, RPC(R5)	

421	(9)	001642	001413			BEQ	50015\$	
422						:		IF THE LOOP OF ERROR IS TURNED OFF THEN
423						:		CONTINUE TEST AT THE NEXT SUBTEST.
424						:		
425		001644	032737	040000	001364	BIT	#LOOPOE, PCFLAG	
426	(9)	001652	001006			BNE	50016\$	
427		001654	042765	000001	000010	BIC	#BITO, RPC(R5)	
428		001662	016537	000010	016156	MOV	RPC(R5), TESTAD	
429		001670				50016\$:		
430	(3)	001672	000456			BR	50017\$	
431						50015\$:		
432						:		CHECK TO SEE IF THE ITERATION COUNT IS COMPLETE
433		001672	026565	000002	000000	CMP	PASCNT(R5), CTLCNT(R5)	
434	(9)	001700	003447			BLE	50020\$	
435		001702	052737	040000	002032	BIS	#EOT, CFLAGS	
436		001710	005037	002036		CLR	TSCCNT	
437						07402		
438						07404		REPORT END OF TEST CONDITION
439		001714	004737	005436		07406		
440		001720	016565	000002	000000	JSR	PC, REPORT	
441	(6)	001726	066565	177776	000000	MOV	PASCNT(R5), CTLCNT(R5)	
442						ADD	ICNT(R5), CTLCNT(R5)	
443						07442		
444						07444		IF IN SEQUENCE TESTS MODE SET UP NEXT TEST
445						07446		
446		001734	032737	000100	001364	BIT	#SEQ, PCFLAG	
447	(9)	001742	001423			BEQ	50021\$	
448		001744	013737	016152	016154	MOV	NEXT, INTEST	
449		001752	005237	016152		INC	NEXT	
450						07520		
451						07522		
452						07524		IF NEXT IS A NON EXISTANT TEST SET EOL
453						07526		AND RETURN TO MONITOR FOR NEW COMMANDS
454						07528		
455		001756	023727	016154	000005	CMP	INTEST, #5	
456	(9)	001764	003407			B.E	50022\$	
457		001766	005037	016152		CLR	NEXT	
458		001772	052737	004000	002032	BIS	#EOL, CFLAGS	
459		002000	000207			RTS	PC	
460		002002	000402			BR	50023\$	
461	(3)	002004				50022\$:		
462		002004	004737	002056		JSR	PC, SUTEST	
463		002010				50023\$:		
464	(3)	002012	000402			BR	50024\$	
465						50021\$:		
466						:		SET UP TEST ADDRESS FOR THE SAME TEST AGAIN.
467		002012	004737	002056		JSR	PC, SUTEST	
468		002016				50024\$:		
469	(3)	002016	000403			BR	50025\$	
470		002020				50020\$:		
471						07742		

```

467 07744 : RETURN TO TEST VIA ADDRESS SUPPLIED BY TEST
468 07746 :
469 002020 016537 000010 016156 : MOV RPC(R5),TESTAD
470 002026 :
471 002026 :
472 002026 000137 001466 : JMP LOOP2
473 :
474 :
475 002032 000000 : *****
476 002034 000000 : CFLAGS: .WORD 0 : FLAGS
477 002036 000000 : TSCPTR: .WORD 0 : POINTER
478 : TSCCNT: .WORD 0 : PASCNT
479 :
480 002040 010164 : TSTBL: TESTC : TABLE OF TEST ADDRESSES *****
481 002042 010444 : TEST1
482 002044 011526 : TEST2
483 002046 013700 : TEST3
484 002050 014652 : TEST4
485 002052 015354 : TEST5
486 002054 177777 : -1
487 :
488 :
489 :
490 :
491 :
492 :
493 :
494 :
495 :
496 002056 : *****
497 002056 006337 016154 : SUTEST INITIALIZES THE TEST ADDRESS POINTER
498 002062 012700 002040 : FOR TEST # IN 'INTEST'
499 002066 063700 016154 : *****
500 002072 011037 016156 :
501 002076 006237 016154 :
502 002102 005065 000004 : SUTEST:
503 002106 : ASL INTEST
504 002106 : MOV #TSTBL,R0
505 002106 : ADD INTEST,R0
506 : MOV (R0),TESTAD
507 : ASR INTEST
508 : CLR MFLAGS,R5
509 :
510 :
511 :
512 :
513 :
514 :
515 :
516 :
517 :
518 :
519 :
520 :
521 :
522 :
523 :
524 :
525 :
526 :
527 :
528 :
529 :
530 :
531 :
532 :
533 :
534 :
535 :
536 :
537 :
538 :
539 :
540 :
541 :
542 :
543 :
544 :
545 :
546 :
547 :
548 :
549 :
550 :
551 :
552 :
553 :
554 :
555 :
556 :
557 :
558 :
559 :
560 :
561 :
562 :
563 :
564 :
565 :
566 :
567 :
568 :
569 :
570 :
571 :
572 :
573 :
574 :
575 :
576 :
577 :
578 :
579 :
580 :
581 :
582 :
583 :
584 :
585 :
586 :
587 :
588 :
589 :
590 :
591 :
592 :
593 :
594 :
595 :
596 :
597 :
598 :
599 :
600 :
601 :
602 :
603 :
604 :
605 :
606 :
607 :
608 :
609 :
610 :
611 :
612 :
613 :
614 :
615 :
616 :
617 :
618 :
619 :
620 :
621 :
622 :
623 :
624 :
625 :
626 :
627 :
628 :
629 :
630 :
631 :
632 :
633 :
634 :
635 :
636 :
637 :
638 :
639 :
640 :
641 :
642 :
643 :
644 :
645 :
646 :
647 :
648 :
649 :
650 :
651 :
652 :
653 :
654 :
655 :
656 :
657 :
658 :
659 :
660 :
661 :
662 :
663 :
664 :
665 :
666 :
667 :
668 :
669 :
670 :
671 :
672 :
673 :
674 :
675 :
676 :
677 :
678 :
679 :
680 :
681 :
682 :
683 :
684 :
685 :
686 :
687 :
688 :
689 :
690 :
691 :
692 :
693 :
694 :
695 :
696 :
697 :
698 :
699 :
700 :
701 :
702 :
703 :
704 :
705 :
706 :
707 :
708 :
709 :
710 :
711 :
712 :
713 :
714 :
715 :
716 :
717 :
718 :
719 :
720 :
721 :
722 :
723 :
724 :
725 :
726 :
727 :
728 :
729 :
730 :
731 :
732 :
733 :
734 :
735 :
736 :
737 :
738 :
739 :
740 :
741 :
742 :
743 :
744 :
745 :
746 :
747 :
748 :
749 :
750 :
751 :
752 :
753 :
754 :
755 :
756 :
757 :
758 :
759 :
760 :
761 :
762 :
763 :
764 :
765 :
766 :
767 :
768 :
769 :
770 :
771 :
772 :
773 :
774 :
775 :
776 :
777 :
778 :
779 :
780 :
781 :
782 :
783 :
784 :
785 :
786 :
787 :
788 :
789 :
790 :
791 :
792 :
793 :
794 :
795 :
796 :
797 :
798 :
799 :
800 :
801 :
802 :
803 :
804 :
805 :
806 :
807 :
808 :
809 :
810 :
811 :
812 :
813 :
814 :
815 :
816 :
817 :
818 :
819 :
820 :
821 :
822 :
823 :
824 :
825 :
826 :
827 :
828 :
829 :
830 :
831 :
832 :
833 :
834 :
835 :
836 :
837 :
838 :
839 :
840 :
841 :
842 :
843 :
844 :
845 :
846 :
847 :
848 :
849 :
850 :
851 :
852 :
853 :
854 :
855 :
856 :
857 :
858 :
859 :
860 :
861 :
862 :
863 :
864 :
865 :
866 :
867 :
868 :
869 :
870 :
871 :
872 :
873 :
874 :
875 :
876 :
877 :
878 :
879 :
880 :
881 :
882 :
883 :
884 :
885 :
886 :
887 :
888 :
889 :
890 :
891 :
892 :
893 :
894 :
895 :
896 :
897 :
898 :
899 :
900 :
901 :
902 :
903 :
904 :
905 :
906 :
907 :
908 :
909 :
910 :
911 :
912 :
913 :
914 :
915 :
916 :
917 :
918 :
919 :
920 :
921 :
922 :
923 :
924 :
925 :
926 :
927 :
928 :
929 :
930 :
931 :
932 :
933 :
934 :
935 :
936 :
937 :
938 :
939 :
940 :
941 :
942 :
943 :
944 :
945 :
946 :
947 :
948 :
949 :
950 :
951 :
952 :
953 :
954 :
955 :
956 :
957 :
958 :
959 :
960 :
961 :
962 :
963 :
964 :
965 :
966 :
967 :
968 :
969 :
970 :
971 :
972 :
973 :
974 :
975 :
976 :
977 :
978 :
979 :
980 :
981 :
982 :
983 :
984 :
985 :
986 :
987 :
988 :
989 :
990 :
991 :
992 :
993 :
994 :
995 :
996 :
997 :
998 :
999 :
1000 :

```

```

5007
5008
5009
5010
5011
5012
5013
5014
5015 002110
5016 002110 032737 001000 001364
5017 002116 001427
5018
5019
5020
5021 002120 032737 000040 001364
5022 002126 001407
5023 002130 004737 002332
5024 002134 004737 002432
5025 002140 004737 002556
5026 002144 000410
5027 002146
5028
5029
5030
5031 002146 013737 001370 016160
5032 002154 004737 002432
5033 002160 013737 016160 016162
5034 002166
5035 002166 042737 001000 001364
5036 002174 000402
5037 002176
5038
5039
5040
5041 002176 004737 002204
5042 002202
5043 002202
5044 002202
5045 002202 000207
5046
5047
5048
5049 002204
5050
5051 002204 032737 000040 001364
5052 002212 001426
5053
5054

```

```

08500 .SBTTL LINE CONTROL & INITIALIZATION
08520 :*****
08540 :THIS SECTION CONTROLS THE SELECTION AND SEQUENCING
08560 :OF SINGLE OR MULTIPLE LINES FOR TESTING.
08580 :*****
08600
08620
LINMON: BIT #NEWMOD,PCFLAG
      BEQ 50002$
08680
08700 : INITIALIZE THE DEVICE HANDLER :
08720 : SET UP A POINTER AREA WITH THE :
08740 : DEVICE ADDRESSES & VECTORS ETC. :
08760
      BIT #MULTI,PCFLAG
      BEQ 50003$
      JSR PC,GVL
      JSR PC,MTW
      JSR PC,GNL
      BR 50004$
50003$:
08880 : GET SELECTED LINE NUMBER AND
08900 : PULL THE DATA FROM THE TABLE.
08920
08940
      MOV LINENO,ONLIN
      JSR PC,MTW
      MOV ONLIN,NXTLIN
50004$:
      BIC #NEWMOD,PCFLAG
      BR 50005$
50002$:
09080 : DO LINESEL SECTION FOR EACH DEVICE
09100 : TO BE TESTED.
09120
09140
      JSR PC,LINESEL
50005$:
50000$:
50001$:
      RTS PC
09220
09240
09260 :*****
09280 :LINESEL ROUTINE TO FURNISH THE IODRIVER WITH DVC POINTERS
09300 :*****
09320
LINESEL:
09360 :
09380 : MULTIPLE LINES UNDER TEST ?
09400
      BIT #MULTI,PCFLAG
      BEQ 50002$
09440 :
09460 : SET UP POINTER AREA WITH DATA FOR

```

E03

```

5556 09480 : THE NEXT DEVICE TO BE TESTED
5557 09500 :
5558 002214 013737 016162 016160 : MOV NXTLIN,ONLIN
5559 09540 :
5560 09560 : RESET EVERYTHING IF AT THE END OF OUR DEVICE LIST.
5561 09580 :
5562 002222 023727 016160 177777 : CMP ONLIN,#-1
5563 002230 001012 : BNE 50003$
5564 002232 052737 000400 001364 : BIS #LDONE,PCFLAG
5565 002240 004737 002332 : JSR PC,GVL
5566 002244 004737 002432 : JSR PC,MTW
5567 002250 004737 002556 : JSR PC,GNL
5568 002254 000404 : BR 50004$
5569 002256 :
5570 09720 :
5571 09750 : SET UP POINTER AREA FOR LINE = 'ONLIN'.
5572 09780 :
5573 002256 004737 002432 : JSR PC,MTW
5574 002262 004737 002556 : JSR PC,GNL
5575 002266 :
5576 002266 000420 : 50004$: BR 50005$
5577 002270 :
5578 002270 004737 002432 : 50002$: JSR PC,MTW
5579 002274 032737 000200 016126 : BIT #SEL,DLFLAG
5580 002307 001404 : BLO 50006$
5581 09920 :
5582 09940 : CHECK TO SEE IF ALL DEVICES
5583 09960 : HAVE BEEN TESTED YET. SET LDONE FLAG.
5584 09980 :
5585 002304 052737 000400 001364 : BIS #LDONE,PCFLAG
5586 002312 000406 : BR 50007$
5587 002314 :
5588 10040 :
5589 10060 : MAKE SHURE THAT WHEN TESTING A SINGLE
5590 10080 : DEVICE . IT DOESN'T GET DROPPED
5591 10100 : BECAUSE OF EXCESSIVE ERRORS
5592 10120 :
5593 002314 052737 000200 016126 : BIS #SEL,DLFLAG
5594 002322 012700 020345 : MOV #E20,R0
5595 002326 104000 : TYPE
5596 10180 :
5597 50007$:
5598 50005$:
5599 50000$:
5600 50001$:
5601 : RTS PC
5602 10260 :
5603 10280 : *****
5604 10300 : GVL THIS ROUTINE FINDS A VALID LINE FOR TESTING
5605 10320 : *****
5606 10340 :
5607 10360 : GVL:
5608 : MOV R3, -(SP)
5609 : MOV #LIN00,R3 : GET ADDR OF LINE TABLE
5610 10380 : G1A: TST (R3) : LIN PRESENT?
5611 10400 : BMI G10 : YES BRANCH
5612 10420 : G1B: ADD #10,R3 : POINT TO OTHER WORD
5613 10440 :

```

F03

02LAFAD LAB6 TERM TST MACY11 30A(1052) 03-JAN-77 00:01 PAGE 1-14
 02LAFAP11 03-JAN-78 11:20 LINE CONTROL & INITIALIZATION

PAGE 003

604	002350	020027	016770	10460		CMP	R0, #TABEND	:END OF TABLE ?
605	002354	001371		10480		BNE	G1A	:NO BRANCH
606	002356	012700	020306	10500	G1C:	MOV	#E19,R0	:NOTIFY OPERATOR - NO LINES
607	002362	104000		10520		TYPE		
608	002364	000137	001372	10540		JMP	RESTR	
609	002370	105713		10560	G1D:	TSTB	(R3)	:LINE SELECTED?
610	002372	100364		10580		BPL	G1B	:NO TRY ANOTHER LINE
611	002374	062703	000006	10600		ADD	#6,R3	:POINT TO OTHER WORD
612	002400	011337	016160	10620		MOV	(R3),ONLIN	:GET DATA FROM TABLE
613	002404	000337	016160	10640		SWAB	ONLIN	
614	002410	105037	016161	10660		CLRB	ONLIN+1	
615	002414	005037	000174	10680		CLR	SWLINE	
616	002420	113737	016160	10700	000174	MOVB	ONLIN,SWLINE	
617	002426	012603				MOV	(SP)+,R3	
618	002430	000207				RTS	PC	:EXIT
619				10740				
620				10760		*****		
621				10780		: MTW THIS ROUTINE TRANSFERS TABLE DATA TO THE WORK AREA		
622				10800		*****		
623				10820				
624	002432			10840	MTW:			
625	002432	010346				MOV	R3, -(SP)	
626	002434	013703	016160	10860		MOV	ONLIN,R3	:GET LINE NO.
627	002440	006303		10880		ASL	R3	
628	002442	006303		10900		ASL	R3	
629	002444	006303		10920		ASL	R3	:XB FOR OFFSET
630	002446	062703	016170	10940		ADD	#LIN00,R3	:ADD IN BASE ADDR
631	002452	012337	016126	10960	MTW1:	MOV	(R3)+,DLFLAG	:GET FLAG WORD
632	002456	012337	016130	10980		MOV	(R3)+,DLADR	:GET ADDRESS
633	002462	012337	016132	11000		MOV	(R3)+,DLVEC	:GET VECTOR
634	002466	011337	016134	11020		MOV	(R3),DL0TH	:GET "OTHER WORD"
635	002472	013737	016130	11040		MOV	DLADR,DVCRXB	
636	002500	062737	000002	016136	016136	ADD	#2,DVCRXB	
637	002506	013737	016136	016140	11080	MOV	DVCRXB,DVCTXS	
638	002514	062737	000002	016140	11100	ADD	#2,DVCTXS	
639	002522	013737	016140	016142	11120	MOV	DVCTXS,DVCTXB	
640	002530	013737	016132	016144	11140	MOV	DLVEC,DXVEC	
641	002536	062737	000004	016144	11160	ADD	#4,DXVEC	
642	002544	062737	000002	016142	11180	ADD	#2,DVCTXB	
643	002552	012603				MOV	(SP)+,R3	
644	002554	000207				RTS	PC	
645				11220		*****		
646				11240		: GNL THIS ROUTINE FINDS THE NEXT VALID LINE TO TEST		
647				11260		*****		
648	002556			11280				
649	002556	010346		11300	GNL:			
650	002556	013703	016160	11320		MOV	R3, -(SP)	
651	002564	005203				MOV	ONLIN,R3	:GET CURRENT LINE
652	002566	006303		11360		INC	R3	:CURRENT +1
653	002570	006303		11380		ASL	R3	
654	002572	006303		11400		ASL	R3	
655	002574	062703	016170	11420		ASL	R3	:XB FOR OFFSET
656	002600	005713		11440		ADD	#LIN00,R3	:ADD IN BASE ADDR OF TABLE
657	002602	100403		11460	GN1:	TST	(R3)	:LINE PRESENT?
658	002604	062703	000010	11480		BMI	GN3	:YES - BRANCH
659	002610	000773		11500	GN2:	ADD	#10,R3	:POINT TO NEXT LINE ENTRY
660				11520		BR	GN1	:CHECK NEXT


```

658 002612 105713 11540
659 002614 100373 11560
660 002616 021327 177777 11580
661 002622 001412 11600
662 002624 062703 000006 11620
663 002630 011337 016162 11640
664 002634 000337 016162 11660
665 002640 105037 016163 11680
666 002644 11700
667 (4) 002644 012603 11720
668 002646 000207 11740
669 002650 012737 177777 016162 11760
670 002656 000772 11780
671 11800
672 11820
673 11840
674 11860
675 11880
676 002660
677 11902
678 11904
679 11906
680 002660 006337 004776
681 002664 006337 004776
682 002670 006337 004776
683 002674 062737 016170 004776 11930
684 11960
685 11980
686 12000
687 002702 032777 100000 002066 12020
688 002710 001003
689 002712 004737 005100
690 002716 000430
691 002720
692 12100
693 12120
694 002720 032737 000004 005002 12140
695 002726 001415 12160
696 002730 052777 000200 002040
697 002736 042777 000037 002032
698 002744 062737 000007 004776
699 002752 117737 002020 001370
700 002760 000407
701 002762
702 12300
703 12320
704 002762 032737 000010 005002 12340
705 002770 001403
706 002772 042777 000200 001776
707 003000
    
```

```

GN3: TSTB (R3) :LINE SELECTED?
      BPL GN2 :NO TRY ANOTHER
      CMP (R3),#-1 :END OF TABLE?
      BEQ GN5 :YES - BRANCH
      ADD #6,R3 :GET "OTHER WORD"
      MOV (R3),NXTLIN
      SWAB NXTLIN : = NEXT AVAILABLE LINE
      CLRB NXTLIN+1
GN4: MOV (SP)+,R3
      RTS PC :EXIT
GN5: MOV #1,NXTLIN :SET NXTLIN TO -1 - NO SELECT
      BR GN4
*****
:UPDATE ROUTINE TO UPDATE INTERFACE TABLE FROM COMMANDS
*****
UPDATE:
: SHIFT THE CONVERTED LINE NO. FOR AN
: OFFSET TO THE LINE TABLE.
:
: ASL DATA
: ASL DATA
: ASL DATA
: ADD IN THE BASE ADDRESS OF THE TABLE.
: ADD #LIN00,DATA
:
: IF THE LINE SELECTED DOESN'T EXIST -
: SEND AN ERROR MESSAGE.
:
: BIT #DLP,DATA
: BNE 50002$
: JSR PC,SELERR
: BR 50003$
50002$:
: ADDING A LINE SETS IT'S "SELECTED" FLAG
: AND CLEARS OUT THE ERROR COUNT FOR THAT LINE
:
: BIT #ADDC,TEMPF
: BEQ 50004$
: BIS #SEL,DATA
: BIC #ABO!,#17,DATA
: ADD #7,DATA
: MOVB DATA,LINENO
: BR 50005$
50004$:
: DROPPING A LINE JUST RESETS IT'S "SELECTED" FLAG
:
: BIT #DROPC,TEMPF
: BEQ 50006$
: BIC #SEL,DATA
50006$:
    
```

H03

02LAFAC LA36 TERM TST MAC11 30A 1052 03-JAN-77 00:01 PAGE 1-16
02LAFAP11 03-JAN-78 11:20 LINE CONTROL & INITIALIZATION

EE2 CC33

1106	003000		
1107	003000		
1108	003000	005037	004776
1109	003004		
1110	003004	000207	
1111			

50005\$:		
50003\$:		
5J000\$:	CLR	DATA
50001\$:	RTS	PC

12480
12500

```

12540
12560
12580
12600
12620
12640
12660
12680
12700
12720
12740
12760
12780
12800
12820
12840
12860
12880
12900
12920
13160
13180
13200
13220
13240
13260
13280
13320
13340
13360
13380
13420
13440
13460
13480
13520

```

```

.SBTTL SWITCH REGISTER ROUTINES ;
:*****
:SWRST TESTS FOR HARDWARE SWITCH REGISTER
:*****
SWRST: MOV #4S,MACHER ;SU NXM TRAP TO 4S
      MOV #PR17,MACHER+2
      TST @SWR ;ACCESS SWITCH REG.
      NOP
      MOV #177570,SWR ;RETAIN HARDWARE POINTER
      BR 6S
4S: MOV #SSWR,SWR ;SU FOR SOFTWARE SWITCH REG.
     ISP2 ;CLEAN THE STACK
6S: MOV #6,MACHER ;RESET TRAP CATCHER
     CLR MACHER+2
     RTS PC ;EXIT

:*****
: CTLGX THIS ROUTINE PRINTS THE PROGRAM CONTROL FLAGS ON THE CONSOLE.
:*****
CTLGX: MOV PCFLAG,-(SP
      MOV #6,-(SP)
      MOV #SW+11,-(SP)
      JSR PC,02ASC
      BICB #6,SW+11
      MOV #SW,RO
      TYPE MOV #L1,RO
50000$:
50001$: RTS PC

:*****
:GETSWS THIS ROUTINE READS THE SWITCH REGISTER AND
: CONVERTS THE DATA TO THE APPROPRIATE CONTROL
: FLAGS OR POINTERS.
:*****
GETSWS:
: STOP HERE FOR OPERATOR TO ENTER CONTROL SWITCHES
:
: HALT
: MOV @SWR,TEMP
:
: IF SWITCHES INDICATE A SINGLE LINE OR A SINGLE TEST
: TO BE DONE STOP SO OPERATOR CAN ENTER LINE TEST DATA
:
: BIT #SEQ,TEMP
: BEQ 50002$
: BIT #MULTI,TEMP
: BNE 50003$
50002$: HALT

```

763	003162	017737	012736	016102		MOV	#SWR, TEMP+2
764	003170	005037	001366			CLR	TESTNO
765	003174	113737	016102	001366		MOVB	TEMP+2, TESTNO
766	003202	005037	001370			CLR	LINENO
767	003206	113737	016103	001370		MOVB	TEMP+3, LINENC
768	003214				50003\$:		
769	003214	032737	000100	016100		BIT	#SEQ, TEMP
770	003222	001406				BEQ	50004\$
771	003224	052737	000100	001364		BIS	#SEQ, PCFLAG
772	003232	005037	001366			CLR	TESTNO
773	003236	000403				BR	50005\$
774	003240				50004\$:		
775	003240	042737	000100	001364		BIC	#SEQ, PCFLAG
776	003246				50005\$:		
777	003246	032737	000040	016100	13780	BIT	#MULTI, TEMP
778	003254	001406				BEQ	50006\$
779	003256	052737	000040	001364		BIS	#MULTI, PCFLAG
780	003264	005037	001370			CLR	LINENO
781	003270	000403				BR	50007\$
782	003272				50006\$:		
783	003272	042737	000040	001364		BIC	#MULTI, PCFLAG,
784	003300				50007\$:		
785	003300	052737	003200	001364	13940	BIS	#ATTN! #NEWMOD! #NEWTST, PCFLAG
786	003306	042737	174037	001364	13980	BIC	#174037, PCFLAG
787	003314	042737	003777	016100		BIC	#3777, TEMP
788	003322	053737	016100	001364		BIS	TEMP, PCFLAG
789	003330				50000\$:		
790	003330				50001\$:		
791	003330	000207				RTS	PC
792					14100		

K03

791				14840	
792				14860	
793				14880	
794				14900	
795				14920	
796	003332	012737	003466	000060	14940
797	003340	012737	000101	177560	14960
798	003346	000207			14980
799					15000
800					15020
801					15040
802					15060
803					15080
804					15100
805					15120
806					15140
807	003350				
808	003350	005077	012610		
809	003354	012700	020370		
810	003360	104000		15220	
811	003362	012737	000001	177560	
812	003370				
813	003370	032737	000200	177560	
(9)	003376	001410			
814	003400	113777	177562	012556	
815	003406	004737	005044		
816	003412	012700	017730		
817	003416	104000		15360	
818	003420				
819	003420	005777	012540		
(5)	003424	001001			
820	003426	000760			
(3)	003430				
821	003430	042777	000200	012526	
822	003436	027727	012522	000116	
(9)	003444	001007			
(6)	003446	023727	016124	000176	
(9)	003454	001403			
823	003456	052737	000020	001364	
824	003464				
825	003464				
(2)	003464				
(2)	003464	000207			

```

.SBTTL CONSOLE TERMINAL ROUTINES
:*****
:CONSGN-- ROUTINE TO INITIALIZE CONSOLE VECTOR AREA
:*****
CONSON: MOV    #READKB, @TKV    ; INTERRUPT TO "READKB"
        MOV    #101, @TKS
        RTS    PC
:*****
: GETSRC
THIS ROUTINE ASKS THE OPERATOR IF HE/SHE
WANTS TO USE CONSOLE CONTROL. THEN SETS
A CONTROL FLAG ACCORDINGLY.
:*****
GETSRC:
        CLR    @PTR
        MOV    @CTLM, RO
        TYPE
        MOV    #1, @TKS
50002$: BIT    #READY, @TKS
        BEQ    50004$
        MOVB  @TKB, @PTR
        JSR   PC ECHO
        MOV    #L1, RO
        TYPE
50004$: TST    @PTR
        BNE   50003$
        BR    50002$
50003$: BICB  #200, @PTR
        CMP   @PTR, #'N
        BNE   50005$
        CMP   SWR, #SSWR
        BEQ   50005$
        BIS   #SWCTL, PCFLAG
50005$:
50000$:
50001$: RTS    PC

```

827					15540	*****
828					15560	: READKB THIS MODULE IS AN INTERRUPT HANDLER
829					15580	: FOR THE CONSOLE TERMINAL.
830					15600	*****
831					15620	
832	003466					READKB:
833	003466	010046				MOV RO,-(SP)
834					15650	:
835					15655	: GET CHAR FROM KEYBOARD BUFFER REG.
836					15660	: CLEAR PARITY BIT IF SET.
837					15665	:
838	003470	013737	177562	003776		MOV #TKB,RDSAV
839	003476	142737	000200	003776		BICB #200,RDSAV
840					15680	:
841					15685	: CHECK FOR DEVICE ERROR
842					15690	:
843	003504	032737	100000	003776		BIT #MERR,RDSAV
(9)	003512	001405				BEQ 50002\$
844	003514	004737	005062			JSR PC,CMDERR
845	003520	005037	177560			CLR #TKS
846	003524	000516				BR 50003\$
(3)	003526					50002\$:
847					15715	:
848					15720	: IF CMD CHAR WAS A CTL-G DO THE CTLGX ROUTINE.
849					15725	: PRINT OUT PCFLAGS ON CONSOLE.
850					15730	:
851	003526	123727	003776	000007		CMPB RDSAV,#CTLG
(9)	003534	001006				BNE 50004\$
852	003536	004737	003064			JSR PC,CTLGX
853	003542	012700	017730			MOV #LI,RO
854	003546	104000			15750	TYPE
855	003550	000504				BR 50005\$
(3)	003552					50004\$:
856					15756	:
857					15757	: IF IN I/O MODE PUT DATA IN I/O BUFFER
858					15758	:
859	003552	032737	000002	001364		BIT #FLAG2,PCFLAG
(9)	003560	001410				BEQ 50006\$
860	003562	113711	003776			MOVB RDSAV,(R1)
861	003566	052737	004000	001364		BIS #DATAIN,PCFLAG
862	003574	005037	007564			CLR DELAYT
863	003600	000470				BR 50007\$
(3)	003602					50006\$:
864					15776	:
865					15777	: IF IN COMMAND MODE PUT DATA IN INBUF
866					15778	: AND CALL INTERPRITER
867					15779	:
868	003602	032737	000001	001364		BIT #FLAG1,PCFLAG
(9)	003610	001406				BEQ 50010\$
869	003612	113777	003776	012344		MOVB RDSAV,#PTR
870	003620	004737	004000			JSR PC,CSI
871	003624	000456				BR 50011\$
(3)	003626					50010\$:
872					15796	:
873					15797	: CLEAR AND GO TO READY STATE.
874					15798	:

M03

875	003626	123727	003776	000003		CMPB	RDSAV,#CTLG
(9)	003634	001003				BNE	50012\$
876	003636	004737	001372			JSR	PC,RESTR
877	003642	000447				BR	50013\$
(3)	003644				50012\$:		
878					15811	:	
879					15812	:	CHECK FOR PRINT REPORTS COMMAND
880					15813	:	
881	003644	123727	003776	000020		CMPB	RDSAV,#CTLP
(9)	003652	001004				BNE	50014\$
882	003654	042737	020000	001364		BIC	#INHRPT,PCFLAG
883	003662	000437				BR	50015\$
(3)	003664				50014\$:		
884					15826	:	
885					15827	:	CHECK FOR NO REPORTS COMMAND
886					15828	:	
887	003664	123727	003776	000016		CMPB	RDSAV,#CTLN
(9)	003672	001004				BNE	50016\$
888	003674	052737	020000	001364		BIS	#INHRPT,PCFLAG
889	003702	000427				BR	50017\$
(3)	003704				50016\$:		
890					15841	:	
891					15842	:	CHECK FOR HALT ON ERROR COMMAND
892					15843	:	
893	003704	123727	003776	000010		CMPB	RDSAV,#CTLH
(9)	003712	001004				BNE	50020\$
894	003714	052737	100000	001364		BIS	#HALTOE,PCFLAG
895	003722	000417				BR	50021\$
(3)	003724				50020\$:		
896					15856	:	
897					15857	:	CHECK FOR LOOP ON ERROR COMMAND
898					15858	:	
899	003724	123727	003776	000014		CMPB	RDSAV,#CTLL
(9)	003732	001004				BNE	50022\$
900	003734	052737	040000	001364		BIS	#LOOPOE,PCFLAG
901	003742	000407				BR	50023\$
(3)	003744				50022\$:		
902					15871	:	
903					15872	:	CHECK FOR CLEAR COMMAND
904					15873	:	
905	003744	123727	003776	000013		CMPB	RDSAV,#CTLK
(9)	003752	001003				BNE	50024\$
906	003754	042737	140000	001364		BIC	#HALTOE!#LOOPOE,PCFLAG
907	003762				50024\$:		
908	003762				50023\$:		
909	003762				50021\$:		
910	003762				50017\$:		
911	003762				50015\$:		
912	003762				50013\$:		
913	003762				50011\$:		
914	003762				50007\$:		
915	003762				50005\$:		
916	003762				50003\$:		
917					15935	:	
918					15940	:	TURN CONSOLE BACK ON & EXIT.
919					15945	:	

920	003762	012737	000101	177560		MOV	#101,2#TKS
921	003770	012600				MOV	(SP)+,RO
922	003772	000002		15960		RTI	
923	003774				50000\$:		
(3)	003774				50001\$:		
(2)	003774	000207				RTS	PC
924				16580			
925	003776	000000		16600	RDSAV:	.WORD	0


```

927      16640      ;*****
928      16660      ; CSI  COMMAND STRING INTERPPETER
929      16680      ;*****
930      16700
931      004000      CSI:
932      16740      ;
933      16760      ; IF CMD CHMR IS AN ESCAPE  ECHO A '$'
934      16780      ; AND SET MODE BACK TO 0.
935      16800      ;
936      004000      123727      003776      000033      CMPB      RDSAV,#ESC
          (9) 004006      001030      BNE      50002$
937      004010      112777      000044      012146      MOVB     #'$,PTR
938      004016      004737      005044      JSR      PC,ECHO
939      004022      012700      017051      MOV      #L3,RO
940      004026      104000      TYPE
941      004030      042737      000001      005002      BIC      #FLAG1,TEMPF
942      17100      ;
943      17120      ; MOVE NEW CONTROL FLAGS TO THE PCFLAG WORD.
944      17140      ; RESET THE BUFFER POINTER.
945      17160      ;
946      004036      013737      005002      001364      MOV      TEMPF,PCFLAG
947      004044      013737      005004      001366      MOV      TEMPT,TESTNO
948      004052      012737      016114      016164      MOV      #INBUF,PTR
949      17235      ; CLEAR ATTENTION FLAGS FROM TEMPF
950      004060      042737      017603      005002      BIC      #17603,TEMPF
951      17260
952      004066      000457      BR      50003$
          (3) 004070
953      17300      50002$:
954      17320      ; IF CMD CHAR WAS A DELETE  RESET THE BUFFER
955      17340      ; POINTER AND ECHO A CR/LF.
956      17360      ;
957      004070      123737      003776      016166      CMPB     RDSAV,DEL
          (9) 004076      001007      BNE     50004$
958      004100      012737      016114      016164      MOV      #INBUF,PTR
959      004106      012700      017730      MOV      #L1,RO
960      004112      104000      TYPE
961      004114      000444      BR      50005$
          (3) 004116
962      17480      50004$:
963      17500      ; IF CMD CHAR WAS A RETURN ECHJ A CR LF
964      17520      ; AND CALL THE DECODER.
965      17540      ;
966      004116      123727      003776      000015      CMPB     RDSAV,#CR
          (9) 004124      001021      BNE     50006$
967      004126      012700      017730      MOV      #L1,RO
968      004132      104000      TYPE
969      004134      004737      004230      JSR      PC,DECODE
970      004140      123727      016114      000121      CMPB     INBUF,#'Q
          (9) 004146      001007      BNE     50007$
971      004150      005037      001056      CLR      TESTNO
972      004154      005037      005034      CLR      TEMPT
973      004160      012737      000001      016152      MOV      #1,NEXT
974      004166
975      004166      000417      50007$: BR      50010$
          (3) 004170      50006$:

```

```

976 17800
977 17820
978 17840
979 17860
980 004170 123727 003776 000077
(9) 004176 001007
981 004200 012700 017103
982 004204 104000 17920
983 004206 012700 020123
984 004212 104000 17960
985 004214 000404
(3) 004216
986 18000
987 18020
988 18040
989 004216 004737 005044
990 004222 005237 016164
991 004226
992 004226
993 004226
994 004226
995 004226
(3) 004226
(2) 004226 000207
996 18220
997 18240

```

```

: IF CMD CHAR WAS A '?' RETYPE THE COMMAND
: SUMMARY & GO TO READY CONDITION.
:
: CMPB RDSAV,#'?'
: BNE 50011$
: MOV #COMSUM,R0
: TYPE
: MOV #RDY,R0
: TYPE
: BR 50012$
50011$:
: ECHO THE INPUT CHARACTER.
:
: JSR PC,ECHO
: INC PTA
50012$:
50010$:
50005$:
50003$:
50000$:
50001$:
: RTS PC

```

```

999
:000
1001
1002
1003
1004 004230
1005
1006 004230 010046
1007 004230 010146
1008 004234 010246
1009 004236 012702 001364
1010 004242 012700 004656
1011 004246 012701 016114
1012 004252
1013
1014
1015
1016 004252 121110
      004254 001145
1017
1018
1019
1020 004256 116037 000001 004774
1021 004264 056037 000002 005002
1022 004272 046037 000004 005002
1023 004300 005037 004776
1024 004304 005037 005000
1025
1026
1027
1028 004310 032737 000340 004774
      004316 001520
1029 004320 126127 000001 000015
      004326 001006
1030
1031
1032
1033 004330 004737 005062
1034
1035
1036
1037 004334 012737 016114 016164
1038 004342 000505
      004344
1039
1040
1041
1042 004344 012746 004776
1043 004350 116137 000001 004776
1044 004356 126127 000002 000015
      004364 001003
1045 004366 012746 000001
1046 004372 000417
      004374
1047 004374 116137 000002 004776
1048 004402 126127 000003 000015

```

```

18280 :***:*****
18300 :DECODE THIS SECTION DECODES THE COMMAND STRING FROM THE
18320 :      CONSOLE, AND SETS THE APPROPRIATE CONTROL FLAGS.
18340 :*****:
18360
DECODE:
18400
      MOV      RO, -(SP)
      MOV      R1, -(SP)
      MOV      R2, -(SP)
      MOV      #CTLBLK, R2
      MOV      #CECTBL, RO
      MOV      #INBUF, R1
50002$:
18560 :
18580 : COMPARE CHAR IN TO FIRST BYTE OF TABLE
18600 :
      CMPB     (R1), (RO)
      BNE     50004$
18640 :
18660 : IF SAME GET FLAGS FROM THE TABLE TO TEMPF
18680 :
      MOVB     1(R0), DECSAV
      BIS     2(R0), TEMPF
      BIC     4(R0), TEMPF
      CLR     DATA
      CLR     DATA2
18800 :
18820 : SEE IF THIS COMMAND REQUIRES ADDITIONAL DATA
18840 :
      BIT     #NREQ, DECSAV
      BEQ     50005$
      CMPB     1(R1), #CR
      BNE     50006$
18900 :
18920 : DATA REQUIRED BUT NOT PRESENT...ERROR
18940 :
      JSR     PC, CMDERR
18980 :
19000 : IF A OR D COMMAND USE DATA FOR LINE NO.
19020 :
      MOV     #INBUF, PTR
      BR     50007$
50006$:
19080 :
19100 : CONVERT THE CHARS TO OCTAL...DATA
19120 :
      MOV     #DATA, -(SP)
      MOVB     1(R1), DATA
      CMPB     2(R1), #CR
      BNE     50010$
      MOV     #1, -(SP)
      BR     50011$
50010$:
50010$:
      MOVB     2(R1), DATA+1
      CMPB     3(R1), #CR

```

```

(9) 004410 001003
1049 004412 012746 000002
1050 004416 000405
(3) 004420
1051 004420 116137 000003 005000
1052 004426 012746 000003
1053 004432
1054 004432
1055 004432 012746 004776
1056 004436 004737 007676
1057
1058 1944C
1059 19460
1060 004442 121027 000122 19480
(9) 004446 001020
1061 19520
1062 19540
1063 19560
1064 19580
1065 004450 005737 004776
(8) 004454 002404
(6) 004456 023727 004776 000005
(9) 004464 003403
(6) 004466
1066 004466 004737 005100
1067 1964C
1068 19660
1069 19680
1070 004472 000403
(3) 004474
1071 004474 013737 004776 005004
1072 004502
1073 004502 052737 100000 005000
1074 004510
1075 19800
1076 19820
1077 19840
1078 004510 121027 000127
(9) 004514 001005
1079 19880
1080 19900
1081 19920
1082 004516 004737 005006
1083 004522 052737 100000 005000
1084 004530
1085 20000
1086 20020
1087 20040
1088 004530 121027 000101
(8) 004534 001403
(6) 004536 121027 000104
(9) 004542 001005
(6) 004544
1089 20080
1090 20100
1091 20120

```

```

BNE 50012$
MOV #2, -(SP)
BR 50013$
50012$:
MOV 3(R1), DATA2
MOV #3, -(SP)
50013$:
50011$:
MOV #DATA, -(SP)
JSR PC, A2BIN
:
: IF R COMMAND USE DATA AS A TEST NO.
:
CMPB (R0), #R
BNE 50014$
:
: CHECK THE LIMITS FOR VALID TEST NO.
:
***** #5 BELOW IS HIGHEST TEST NO THIS DIAGNOSTIC *****
TST DATA
BLT 50015$
CMP DATA, #5
BLE 50016$
50015$:
JSR PC, SELERR
:
: OUT OF RANGE ERROR.
:
SR 50017$
50016$:
MOV DATA, TEMPT
50017$:
BIS #MERR, DATA2
50014$:
:
: IF W COMMAND USE DATA AS WIDTH
:
CMPB (R0), #W
BNE 50020$
:
: GO CHECK FOR VALID LIMITS ON WIDTH ENTRY.
:
JSR PC, CHKW
BIS #MERR, DATA2
50020$:
:
: IF ADDING OR DROPPING A LINE CALL UPDATE ROUTINE
:
CMPB (R0), #A
BEQ 50021$
CMPB (R0), #D
BNE 50022$
50021$:
:
: TAKE LINE NO. AND UPDATE INTERFACE TABLE
:

```

```

1092 004544 004737 002660
1093 004550 052737 100000 005000
1094 004556
1095 004556
1096 004556 000403
(3) 004560
1097 004560 052737 100000 005000
1098 004566
1099 004566 000415
(3) 004570
1100 004570 062700 000006
1101
1102
1103
1104
1105 004574 020027 004774
(9) 004600 001010
1106 004602 004737 005062
1107 004606 012737 016114 016164
1108 004614 052737 100000 005000
1109 004622
1110 004622
1111
1112
1113
1114
1115 004622 032737 100000 005000
(5) 004630 001001
1116 004632 000607
(3) 004634
1117 004634 005037 005000
1118
1119
1120
1121 004640 012737 016114 016164
1122 004646 012602
1123 004650 012601
1124 004652 012600
1125 004654
(3) 004654
(2) 004654 000207
1126
1127
1128 004656 123 000
1129 004660 001200 000040
1130 004664 115 000
1131 004666 001240 000000
1132 004672 121 000
1133 004674 000300 000000
1134 004700 122 200
1135 004702 000200 000100
1136 004706 104 100
1137 004710 002010 000004
1138 004714 101 100
1139 004716 002004 000010
1140 004722 124 000

```

```

JSR PC,UPDATE
BIS #MERR,DATA2
50022$:
50007$:
BR 50023$
50005$:
BIS #MERR,DATA2
50023$:
BR 50024$
50004$:
ADD #6,R0
:
: IF THE CHAR IN DOESN'T COMPARE TO ANY
: TABLE ENTRY THE COMMAND IS INVALID
:
CMP RO,#DTEND
BNE 50025$
JSR PC,CMDERR
MOV #INBUF_PTR
BIS #MERR,DATA2
50025$:
50024$:
:
: KEEP LOOKING AT CHAR UNTIL IT'S
: DECODED, OR END OF TABLE (ERROR)
:
BIT #MERR,DATA2
BNE 50003$
BR 50002$
50003$:
CLR DATA2
:
: RESET THE INPUT BUFFER POINTER
:
MOV #INBUF_PTR
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
50000$:
50001$:
RTS PC
:
: * * * * *
DECTBL: .BYTE 'S,0 ;DECODE TABLE
.WORD ATTN!NEWMOD,MULTI ;FIRST - CHAR TO BE DECODED
.BYTE 'M,0 ;SECOND - CONTROL BITS
.WORD ATTN!MULTI!NEWMOD,0 ;THIRD - SET MASK
.BYTE 'Q,0 ;FOURTH - CLEAR MASK
.WORD ATTN!SEQ,0
.BYTE 'R,200
.WORD ATTN,SEQ
.BYTE 'D,100
.WORD DR0PC!NEWST,ADD0
.BYTE 'A,100
.WORD ADD0!NEWST,DR0PC
.BYTE 'T,C ;CONTROL BITS:

```

```

20320
20340
20360
20380
20520
20540
20560
20580
20660
20680
20700
20820
20840
20860
20880
20900
20920
20940
20960
20980
21000
21020
21040
21060
21080
21100

```

```

1141 004724 010000 000000 21120
1142 004730 114 000 21140
1143 004732 040000 000000 21160
1144 004736 110 000 21180
1145 004740 100000 000000 21200
1146 004744 116 000 21220
1147 004746 020000 000000 21240
1148 004752 120 000 21260
1149 004754 000000 020000 21280
1150 004760 103 000 21300
1151 004762 000000 140000 21320
1152 004766 127 04C 300 21340
      004771 000 00C 300
1153 004774 21360
1154 004774 000000 21380
1155 004776 000000 21400
1156 005000 000000 21420
1157 005002 000000 21440
1158 005004 000000 21460
1159
1160
1161
1162
1163
1164
1165
1166 005006 21480
1167 005006 000000 21500
1168 005006 000000 21520
1169 005006 000000 21540
1170 005006 023727 004776 000032 21560
      (8) 005014 002404 21580
1171 005016 023727 004776 000204 21600
      (9) 005024 003403
1172 005026 004737 005100 21621
      (6) 005026 000403 21622
1173 005034 013737 004776 016146 21623
      (3) 005034
1174 005042
1175 005042
      (3) 005042
      (2) 005042 005207
1176
1177
1178

```

```

.WORD PRINTT,C
.BYTE 'L,O
.WORD LOOPOE,C
.BYTE 'H,O ;5 = GET WIDTH
.WORD HALTOE,C ;6 = GET LINE N
.BYTE 'N,O ;7 = GEN TEST NO.
.WORD INHRPT
.BYTE 'P,O
.WORD O,INHRPT
.BYTE 'C,O
.WORD O,HALTOE!LOOPOE
.BYTE 'W,40,0,0,0,0

DTEND:
DECSAV: .WORD 0
DATA: .WORD 0
DATA2: .WORD 0
TEMPF: .WORD 0 ;TEMPORARY PCFLAG WORD
TEMPT: .WORD 0 ;TEMPORARY TEST NO.

:*****
:CHKW THIS ROUTINE VALIDATES A "W" COMMAND
:*****

CHKW:
: RANGE OF 26 THRU 132 CHARACTERS IS VALID
:
CMP DATA,#32
BLT 50002$
CMP DATA,#132.
BLE 50003$
50002$: JSR PC_SELERR
BR 50004$
50003$: MOV DATA,WIDTH
50004$:
50000$:
50001$: RTS PC

```

H04

CLAFAD LA36 TERM TST MACY11 30A 1052 03-JAN-77 00:01 PAGE 1-29
CLAFAP11 03-JAN-78 11:20 CONSOLE TERMINAL ROUTINES

SES 0046

1179					21820
1180					21840
1181					21860
1182					21880
1183	005044	105737	177564		21900
1184	005050	100375			21920
1185	005052	117737	011106	177566	21940
1186	005060	000207			21960
1187					21980

```

:*****
:ECHO CONSOLE KEYBOARD ECHO ROUTINE: PTR HAS ADDR OF CHAR
:*****
ECHO: TSTB @#TPS
      BPL ECHO
      MOV @PTR,@#TF3
      RTS PC

```

```

1189          22020          .SBTTL  ERROR & REPORT ROUTINES
1190          22040          :*****
1191          22060          :CMDERR  ROUTINE TO HANDLE INVALID COMMANDS
1192          22080          :*****
1193          22100
1194 005062          CMDERR:
1195 005062 012700 0200.7          TYPE  MOV      #ER1,RO
1196 005066 104000          22160          TYPE  MOV      #RDY,RO
1197 005070 012700 020123          TYPE
1198 005074 104000          22200          50000$:
1199 005076          50001$:
1200 (3) 005076          RTS      PC
1201 (2) 005076 000207          :*****
1202          22240          :SELERR  ROUTINE TO HANDLE SELECTION ERRORS
1203          22260          :*****
1204          22280          :*****
1205          22300          SELERR:
1206 005100          TYPE  MOV      #ER2,RO
1207 005104 012700 020032          TYPE  MOV      #RDY,RO
1208 005106 104000          22360          TYPE  MOV      #INBUF,PTR
1209 005112 012700 020123          22400          50000$:
1210 005114 012737 016114 016164          50001$:
1211 (3) 005122          RTS      PC
1212 (2) 005122 000207          22460

```



```

1213          22500  :*****
1214          22520  :ERRORS ERROR LOGGER AND TYPEOUT ROUTINE
1215          22540  :
1216          22560  :*****
1217          22580  :
1218 005124          ERROR:
1219 005124 005037 005434          CLR      ERRSAV
1220 005130 032737 020000 001364  BIT      #INHRPT,PCFLAG
      005136 001044          BNE      50002$
1221          22642  :
1222          22644  :CONVERT TEST NO. FOR OUTPUT
1223          22646  :
1224 005140 013746 016154          MOV      INTEST,-(SP)
1225 005144 012746 000002          MOV      #2,-(SP)
1226 005150 012746 020001          MOV      #ERO+16,-(SP)
1227 005154 004737 007566          JSR      PC,02ASC
1228          22722  :
1229          22724  :CONVERT ERROR NO. FOR OUTPUT
1230          22726  :
1231 005160 113737 002032 005434  MOVB     CFLAGS,ERRSAV
1232 005166 013746 005434          MOV      ERRSAV,-(SP)
1233 005172 012746 000003          MOV      #3,-(SP)
1234 005176 012746 017770          MOV      #ERO+7,-(SP)
1235 005202 004737 007566          JSR      PC,02ASC
1236          22822  :
1237          22824  :CONVERT LINE NO. FOR OUTPUT
1238          22826  :
1239 005206 013746 016160          MOV      ONLIN,-(SP)
1240 005212 012746 000002          MOV      #2,-(SP)
1241 005216 012746 020011          MOV      #ERO+24,-(SP)
1242 005222 004737 007566          JSR      PC,02ASC
1243 005226 012760 017761          MOV      #ERO,RO
1244 005232 104000          TYPE
1245          22940  :
1246          22960  :
1247          22980  :CLEAR THE ERROR FLAG
1248          23000  :
1249 005234 042737 100377 002032  BIC      #MERRN,CFLAGS
1250          23040  :
1251          23060  :GET THE POINTER SUPPLIED BY THE PROGRAM
1252          23080  :AND PRINT THE ERROR DESCRIPTION MSG.
1253          23100  :
1254 005242 013700 002034          MOV      TSCPTR,RO
1255 005246 104000          TYPE
1256 005250          50002$:
1257 005250 005037 005434          CLR      ERRSAV
1258          23200  :
1259          23220  :UPDATE THE ERROR COUNT FOR THE FAILING LINE
1260          23240  :
1261 005254 013737 016160 005434  MOV      ONLIN,ERRSAV
      005262 006337 005434          ASL     ERRSAV
      005266 006337 005434          ASL     ERRSAV
      005272 006337 005434          ASL     ERRSAV
1262 005276 062737 016170 005434  ADD     #LINOO,ERRSAV
1263 005304 005277 000124          INC     #ERRSAV
1264          23340  :
1265          23360  :IF LOOP ON ERROR IS SET , MAKE THE
    
```

```

1266      23380      : RETURN ADDRESS OF THE TEST ODD.
1267      23400      : THE TEST CONTROLLER WILL USE THE OLD
1268      23420      : RPC TO RE-DO THE SUBTEST.
1269      23440      :
1270      005310    032737 040000 001364      BIT      #LOOPOE,PCFLAG
      (9) 005316    001403      BEQ      50003$
1271      005320    052765 000001 000010      BIS      #BITO,RPC(R5)
1272      005326      50003$:
1273      23520      :
1274      23540      : SEE IF LINE ABORT FLAG IS SET
1275      23560      :
1276      005326    032777 000020 000100      BIT      #ABO,@ERRSAV
      (9) 005334    001431      BEQ      50004$
1277      23600      :
1278      23620      : IF ABORT IS SET DESELECT THE LINE
1279      23640      : UNLESS IT'S THE ONLY ONE BEING TESTED
1280      23660      :
1281      005336    032737 000040 001364      BIT      #MULTI,PCFLAG
      (9) 005344    001417      BEQ      50005$
1282      005346    042777 000377 000060      BIC      #SEL,#177,@ERRSAV
1283      005354    013746 016160      MOV      ONLIN,-(SP)
1284      005360    012746 000002      MOV      #2,-(SP)
1285      005364    012746 020270      MOV      #DR1,-(SP)
1286      005370    004737 007566      JSR      PC,02ASC
1287      23800      :
1288      23820      : NOTIFY OPERATOR THAT LINE WAS DROPPED
1289      23840      :
1290      005374    012700 020242      MOV      #DRO,RO
1291      005400    104000      TYPE
1292      23880      :
1293      23900      : IF TESTING ONLY ONE LINE DONOT ALLOW IT TO BE DESELECTED
1294      23920      :
1295      005402    000406      BR      50006$
      (3) 005404
1296      005404    052777 000200 000022      BIS      #SEL,@ERRSAV
1297      005412    042777 000020 000014      BIC      #ABO,@ERRSAV
1298      005420      50006$:
1299      005420      50004$:
1300      24040      :
1301      24060      : HALT HERE IF HALT ON ERROR IS SET
1302      24080      :
1303      005420    032737 100000 001364      BIT      #HALTOE,PCFLAG
      (9) 005426    001401      BEQ      50007$
1304      005430    000000      HALT
1305      24120      50007$:
1306      005432      50000$:
      (3) 005432
      (2) 005432    000207      50001$:
1307      24180      RTS      PC
1308      005434    000000      ERRSAV: .WORD 0
1309      24200
1310      24220
1311      24240
1312      24260
1313      24280
1314      24300
      005436      24320
: *****
: REPORT THIS ROUTINE HANDLES END OF TEST AND
: END OF PASS REPORTS.
: *****
REPORT.

```

```

1315      24345 ; CHECK FOR END OF TEST CONDITION
1316      24350 ;
1317 005436 032737 040000 002032      BIT      #EOT,CFLAGS
(9) 005444 001423                      BEQ      50002$
1318      24365 ;
1319      24370 ; CONVERT TEST NO. FOR OUTPUT
1320      24375 ;
1321 005446 013746 016154      MOV      INTEST,-(SP)
1322 005452 012746 000002      MOV      #2,-(SP)
1323 005456 012746 020210      MOV      #EOTM+18.,-(SP)
1324 005462 004737 007566      JSR      PC,02ASC
1325      24445 ;
1326      24450 ; SEND END OF TEST MESSAGE
1327 005466 042737 040000 002032      BIC      #EOT,CFLAGS
1328      24455 ;
1329 005474 012700 020166      MOV      #EOTM,RO
1330 005474 004737 007010      JSR      PC,MTYPE
1331 005474 012700 017051      MOV      #L3,RO
1332 005510 004737 007010      JSR      PC,MTYPE
1333      50002$:
1334      24505 ;
1335      24510 ; CHECK FOR END OF PASS CONDITION
1336      24515 ;
1337 005514 032737 020000 002032      BIT      #EOP,CFLAGS
(9) 005522 001425                      BEQ      50003$
1338 005524 013746 016154      MOV      INTEST,-(SP)
1339      24545 ;
1340      24550 ; CONVERT TEST NO. FOR OUTPUT
1341      24555 ;
1342 005530 012746 000002      MOV      #2,-(SP)
1343 005534 012746 020161      MOV      #EOPM+19.,-(SP)
1344 005540 004737 007566      JSR      PC,02ASC
1345      24605 ;
1346      24610 ; CONVERT PASS NUMBER FOR OUTPUT
1347      24615 ;
1348 005544 013746 002036      MOV      TSCCNT,-(SP)
1349 005550 012746 020145      MOV      #EOPM+7,-(SP)
1350 005554 004737 010006      JSR      PC,BIN20A
1351      24685 ;
1352      24690 ; SEND END OF PASS MESSAGE.
1353      24695 ;
1354 005560 012700 020136      MOV      #EOPM,RO
1355 005564 004737 007010      JSR      PC,MTYPE
1356 005570 042737 020000 002032      BIC      #EOP,CFLAGS
1357 005576
1358 005576      50003$:
(3) 005576      50000$:
(2) 005576 000207      50001$:
1359      RTS      PC
1360      24800
1361      24820
1362      24840

```

```

1363 00050 :*****
1364 00070 : SETIO ROUTINE TO SET I/O MODE
1365 00090 :*****
1366 00110
1367 00130
1368 005600 SETIO:
1369 005600 50002$:
1370 005600 032737 000001 001364 BIT #FLAG1,PCFLAG
(9) 005606 001003 BNE 50003$
1371 005610 052737 000003 001364 BIS #FLAG1,#FLAG2,PCFLAG
1372 005616 50003$:
1373 005616 032737 000001 001364 BIT #FLAG1,PCFLAG
(7) 005624 001765 BEQ 50002$
(4) 005626 032737 000002 001364 BIT #FLAG2,PCFLAG
(7) 005634 001761 BEQ 50002$
1374 005636 50000$:
(3) 005636 50001$:
(2) 005636 000207 RTS PC
1375 00320
1376 00340

```

```

1378 00700 .SBTTL INTERFACE SIZER ROUTINES
1379 00900 ;*****
1380 00950 ;BUILD SUBROUTINE TO BUILD THE DEVICE TABLE USED
1381 01000 ; IN MULTI LINE MODE.
1382 01050 ;*****
1383 005640 012737 000003 000066 01100 BUILD: MOV #BPT,66 ;SET UP CONSOLE TRAP
1384 005646 012737 006120 000004 01150 MOV #4$,MACHER ;SET UP NXM TRAP
1385 005654 012701 016170 01200 MOV #LIN00,R1
1386 005660 012137 016126 01220 1$: MOV (R1)+,DLFLAG
1387 005664 012137 016130 01240 MOV (R1)+,DLADR
1388 005670 012137 016132 01260 MOV (R1)+,DLVEC
1389 005674 012137 016134 01280 MOV (R1)+,DLOTH
1390 005700 013737 016130 016140 01400 MOV DLADR,DVCTXS
1391 005706 062737 000004 016140 01430 ADD #4,DVCTXS
1392 005714 013737 016140 016142 01440 MOV DVCTXS,DVCTXB
1393 005722 062737 000002 016142 01450 ADD #2,DVCTXB
1394 005730 113737 016135 016160 01500 MOVB DLOTH+1,ONLIN
1395 005736 005777 010166 01750 TST @DLADR
1396 005742 052737 100000 016126 01850 BIS #DLP,DLFLAG ;TRY TO ACCESS DVC.
1397 005750 012737 000300 007564 01900 MOV #300,DELAYT ;SET DVC PRESENT FLAG
1398 005756 112777 000076 010156 01925 MOVB #'>,@DVCTXB ;SET UP FOR DELAY
1399 005764 052777 000100 010146 01950 BIS #100,@DVCTXS ;TXMIT A ">" CHARACTER
1400 005772 104006 02050 DELAYR ;SET DVC TX INTR ENABLE
1401 005774 005737 016132 02100 TST DLVEC ;WAIT FOR INTERRUPT
1402 006000 001433 02125 BEQ 2$ ;IF ZERO NO INTERRUPT OCCURED
1403 02150 ;NO INTERRUPT - BRANCH
1404 02200 ;OTHERWISE DLVEC=ADDR THAT
1405 006002 052737 000200 016126 02300 BIS #SEL,DLFLAG ;DVC INTERRUPTED TO VIA INTRAP
1406 006010 013741 016134 02350 3$: MOV DLOTH,-(R1) ;SET SELECTED FLAG
1407 006014 013741 016132 02400 MOV DLVEC,-(R1)
1408 006020 013741 016130 02450 MOV DLADR,-(R1) ;PUT NEW INFORMATION
1409 006024 013741 016126 02500 MOV DLFLAG,-(R1) ;INTO LINE TABLE
1410 02550
1411 006030 062701 000010 02600 ADD #10,R1 ;JUMP POINTER TO NEXT LINE
1412 006034 020127 016770 02650 5$: CMP R1,#TABEND ;ALL DONE ?
1413 006040 001307 02700 BNE 1$ ;NO - DO NEXT LINE
1414 006042 162701 000010 02750 6$: SUB #10,R1 ;CKECH LAST ENTRY
1415 006046 005711 02800 TST (R1) ;FOR LINE PRESENT
1416 006050 100403 02850 BMI 7$
1417 006052 012711 177777 02900 MOV #-1,(R1) ;IF NOT SET IT TO END
1418 006056 000771 02950 BR 6$ ;OF TABLE
1419 006060 012737 000006 000004 03100 7$: MOV #6,MACHER ;RESET TRAP CATCHER
1420 006066 000207 03150 RTS PC
1421 006070 052737 000020 016126 03200 2$: BIS #AB0,DLFLAG ;SET ABORT FLAG
1422 006076 042737 000200 016126 03250 BIC #SEL,DLFLAG ;MAKE SURE LINE IS DESELECTED
1423 006104 004737 005124 JSR PC,ERROR
1424 006110 012700 020055 MOV #EA7,RO ;SU ERROR MSG
1425 006114 104000 03350 TYPE ;TYPE MSG ON CONSOLE
1426 006116 000734 03400 BR 3$ ;FIX TABLE ENTRIES
1427 03500
1428 006120 062706 000004 03550 4$: ADD #4,SP ;ERASE INTR FROM STACK
1429 006124 000743 03600 BR 5$ ;GET NEXT LINE ENTRY
1430 03650
1431 03700 ;*****
1432 03750 ; CATCH REPLACES TRAP CATCHER FROM 100 TO 1000 .
1433 03800 ;*****

```

1434	006126	012700	001000	03850	CATCH: MOV	#1000,RO	: START AT 1000
1435	006132	005040		03900	1\$: CLR	-(RO)	: PUT HALT IN PC+2
1436	006134	010037	016100	03950	MOV	RO,TEMP	
1437	006140	013740	016100	04000	MOV	TEMP -(RO)	: PUT PC+2 IN PC
1438	006144	020027	000100	04050	CMP	RO,#100	: FIN?
1439	006150	002370		04100	BGE	1\$: NO - DO MORE
1440	006152	012737	003466 000060	04200	MOV	#READKB,2#TKV	: SU CONSOLE
1441	006160	000207		04250	RTS	PC	
1442				04300			
1445				04500			

```

1451
1452
1453
1454
1455
1456
1457
1458 006162
      (4) 006162 011637 016112
1459 006166 162737 000002 016112 05800
1460 006174 017737 007712 016110 05850
1461 006202 042737 104400 016110 05950
1462 006210 062737 006230 016110 06000
1463 006216 017737 007666 016112 06050
1464 006224 000177 007662 06100
1465
1466
1467 005230 006240
1468 006232 007212
1469 006234 007162
1470 006236 007530

```

```

05400
05450 .SBTTL EMT HANDLER
05500
05550
05600
05650
05700
05750
: * * * * *
: THIS SECTION CONTAINS THE HANDLER AND MOST ROUTINES ACCESSED
: BY TRAPS THROUGH LOCATION 30.
: * * * * *

EMTBOS:
      MOV      (SP),TEMP+12
      SUB      #2,TEMP+12 ;GET REAL PC
      MOV      @TEMP+12,TEMP+10 ;GET EMT INSTRUCTION
      BIC      #104400,TEMP+10 ;MASK INSTR BITS
      ADD      @EMTABL,TEMP+10 ;ADD TABLE ADDR
      MOV      @TEMP+10,TEMP+12
      JMP      @TEMP+12

.EVEN
EMTABL: ETYPE ;NSOLE TYPE ROUTINE
        PRTLTB ;LINE TABLE PRINTER
        INTRAP ;DL INTERRUPT CATCHER
        DELAYM ;DELAY ROUTINE

```

```

1473 06650
1474 06700
1475 06750
1476 06800
1477 06850
1478 06900
1479 006240 105710 06950
1480 006242 001406 07000
1481 006244 105737 177564 07050
1482 006250 100375 07100
1483 006252 112037 177566 07150
1484 006256 000770 07200
1485 006260 105737 17756 07250
1486 006264 100375 07275
1487 006266 000002 07287
1488 07300
1489 07350
1490 07400
1491 07450
1492 07500
1493 07550
1494 006270 010237 006646 07600
1495 006274 010446 07650
1496 006276 012702 020434 07750
1497 006302 005037 020670 07800
1498 006306 012704 020574 07850
1499 006312 013722 016132 07900
1500 006316 013737 016144 016150 07950
1501 006324 062737 000002 016150 08000
1502 006332 013777 016150 007604 08050
1503 006340 012777 000004 007602 08100
1504 006346 012737 000310 007564 08150
1505 006354 113777 006646 007560 08200
1506 006362 012777 000100 007550 08250
1507 006370 005237 020670 08300
1508 006374 004737 002110
1509 006400 032737 000400 001364 08400
1510 006406 001741 08450
1511 006410 042737 000400 001364 08500
1512 006416 010237 006650 08550
1513 006422 104006 08600
1514 006424 005737 020670 08650
1515 006430 001004 08700
1516 006432 08750
1517 006432 012604
1518 006434 013702 006646 08800
1519 006440 000207 08850
1520 006442 010437 006652 08900
1521 006446 012704 020574 08950
1522 006452 012702 020434 09000
1523 006456 021224 09050
1524 006460 001404 09100
1525 006462 020437 006652 09150
1526 006466 001403 09200
1527 006470 000772 09250

```

```

.SBTTL I O DRIVERS
*****
: ETYPE CONSOLE OUTPUT ROUTINE. ENTER WITH ADDRESS OF
: DATA IN R0. NULL TERMINATES OUTPUT.
*****
ETYPE: TSTB R0 :CHECK FOR NULL
BEQ 3$ :EXIT ROUTINE
1$: TSTB 0$TPS :CHECK FOR TRANSMIT REAC
BPL 1$ :WAIT
MOVB (R0)+,0$TPB :TRANSMIT CHARACTER
BR ETYPE :GET NEXT CHAR
3$: TSTB 0$TPS :WAIT TILL ALL DONE
BPL 3$
RTI :EXIT...

*****
:MECHO TERMINAL OUTPUT ROUTINE - SINGLE CHAR
: CHAR IN R2
: INTERRUPT DRIVEN ALL LINES
*****
MECHO: MOV R2,MSAVE
MOV R4 -(SP)
MOV #STACK2,R2 :INITIALIZE STACK2
CLR ENDS :ZERO COUNT
1$: MOV #STACK3,R4 :INITIALIZE STACK3
MOV DLVEC,(R2)+ :GET THE BASE VECTOR ADDR
MOV TXVEC,SAVE :SAVE THE VECTOR
ADD #2,SAVE :PUT ADDR+2 INTO ADDR
MOV #IOT,0$SAVE :PUT TRAP INTO ADDR+2
MOV #200,DELAYT :WAIT FOR 200 MS.
MOVB MSAVE,0$DVCTXB :PUT CHAR IN BUF REG
MOV #100,0$DVCTXS :ENABLE TX INTERRUPT
MOV ENDS :ADD 1 TO INTR PENDING COUNT
INC ENDS
JSR PC,LINMON
BIT #LDONE,PCFLAG :END OF DVC LIST ?
BEQ 1$ :NO DO THIS LINE
BIC #LDONE,PCFLAG
MOV R2,MSAVE+2 :SAVE STACK2 POINTER
DELAYR
TST ENDS
BNE 3$ :ALL PENDING INTERRUPTS SHOULD
:BE COUNTED DOWN BY TXTRAP.
12$: MOV (SP)+,R4
MOV MSAVE,R2
2$: RTS PC :EXIT...
3$: MOV R4,MSAVE+4 :SAVE STACK 3 LIMIT
MOV #STACK3,R4 :RESET STACK3 POINTER
MOV #STACK2,R2 :RESET STACK2 POINTER
4$: CMP (R2),(R4)+ :VECTOR MATCH ?
BEQ 5$ :YES - BRANCH
CMP R4,MSAVE+4 :STACK END ?
BEQ 6$ :YES - BRANCH
BR 4$ :COMPARE NEXT VECT.

```



```

1527 006472 005012 09300 55: CLR (R2) ;ERASE VECT OUT
1528 006474 005044 09350 CLR -(R4) ;ERASE VECT IN
1529 006476 062702 000002 09400 65: ADD #2,R2 ;MOVE STACK POINTER
1530 006502 020237 006650 09450 CMP R2,MSAVE+2 ;END OF OUT STACK ?
1531 006506 001403 09500 BEQ 75 ;YES - GO GET ODD VECTOR
1532 006510 012704 020574 09550 MOV #STACK3,R4 ;RESET STACK3 POINTER
1533 006514 000760 09600 BR 45 ;KEEP SORTING
1534 006516 012702 020434 09650 75: MOV #STACK2,R2 ;RESET STACK2 POINTER
1535 006522 005712 09700 85: TST (R2) ;CHECK FOR NON ZERO
1536 006524 001003 09750 BNE 95
1537 006526 062702 000002 09800 ADD #2,R2
1538 006532 000773 09850 BR 85
1539 006534 012737 016174 006652 09900 95: MOV #LIN00+4,MSAVE+4 ;GET VECT FROM TABLE
1540 006542 027712 000104 09950 105: CMP #MSAVE+4,(R2) ;MATCH ?
1541 006546 001404 10000 BEQ 115 ;YES THIS LINE IS N.G.
1542 006550 062737 000010 006652 10050 ADD #10,MSAVE+4 ;MOVE POINTER TO NEXT
1543 006556 000771 10100 BR 105
1544 006560 062737 000002 006652 10150 115: ADD #2,MSAVE+4 ;GET LINE NUMBER
1545 006566 017737 000060 016160 10200 MOV #MSAVE+4,ONLIN
1546 006574 105037 016160 10250 CLRB ONLIN
1547 006600 000337 016160 10300 SWAB ONLIN ;ERASE JUNK BITS
1548 006604 004737 002432 10350 JSR PC,MTW ;MOVE TABLE TO WORK AREAS
1549 006610 012737 020055 002034 10400 MOV #ER7,TSCPTR ;POINT TO ERROR MESSAGE
1550 006616 112737 000377 002032 10450 MOVB #377,CFLAGS ;ERROR NO.
1551 006624 052737 100000 002032 10500 BIS #MERR,CFLAGS ;SET ERROR FLAG
1552 10551
1553 10600 ;*****
1554 10650 ;ERROR 377 *
1555 006632 004737 005124 10750 JSR PC,ERROR ;ERASE ERROR DATA
1556 006636 042737 100377 002032 10800 BIC #MERR,CFLAGS ;CLEAN HOUSE & EXIT
1557 006644 000672 10850 BR 125
1558 006646 000000 000000 10850 MSAVE: .WORD 0,0,0
1559 10855
1560 10856
1561 10857
1562 10860 ;*****
1563 10861 ;SECHO SINGLE LINE ECHO ROUTINE
1564 10862 ; ENTER WITH CHAR IN R2
1565 10863 ; TRANSMITS TO DVC VIA I/O DRIVER WORK AREA
1566 10864 ;*****
1567 10865
1568 10866
1569 006654 SECHO:
1570 006654 013737 016144 016150 MOV TXVEC,SAVE
1571 006662 062737 000002 016150 ADD #2,SAVE
1572 006670 012777 007510 007246 MOV #STRAP,ATXVEC
1573 006676 012777 000200 007244 MOV #PRI4,MSAVE
1574 006704 012737 000144 007564 MOV #100,DELAYT
1575 006712 110277 007224 MOV R2,ADVCTXB
1576 006716 012777 000100 007214 MOV #100,ADVCTXS
1577 006724 005237 020670 INC ENDS
1578 006730 104006 10877 DELAYR
1579 006732 005737 020670 TST ENDS
1580 006736 001413 000025 BEQ 500025
1581 006740 012737 020055 002034 MOV #ER7,TSCPTR
1581 006746 052737 100376 002032 BIS #376!MERR,CFLAGS

```

F05

```

1582 006754 004737 005124
1583 006760 042737 100377 002032
1584 006766
1585 006766 013777 016150 007150
1586 006774 005077 007150
1587 007000 013737 016160 016150
1588 007006
(3) 007006
(2) 007006 000207
1589
1590
1591
1592
1593
1594
1595
1596 007010 112002
1597 007012 001403
1598 007014 004737 006270
1599 007020 000773
1600 007022 000207
1601
1602
1603
1604
1605
1606
1607
1608
1609 007024
1610 007024 042737 004000 001364
1611 007032 010237 007564
1612 007036 012777 000101 007064
1613 007044 104006
1614
1615 007046 032737 004000 001364
(9) 007054 001003
1616 007056 052765 100000 000004
1617 007064
1618
1619 007064 105737 016160
(9) 007070 001004
1620 007072 042737 000003 001364
1621 007100 000402
(3) 007102
1622 007102 005077 007022
1623 007106
1624 (3) 007106
(2) 007106 000207
1625
1626
1627
1628
1629
1630

```

```

JSR PC_ERROR
BIC #MERRN,CFLAGS
50002$: MOV SAVE,@TXVEC
CLR @SAVE
MOV ONLIN,SAVE
50000$:
50001$: RTS PC

10900
10950
11000 :*****
11050 :MTYPE TERMINAL OUTPJT ROUTINE - LINE TABLE VERSION
11100 : ENTER WITH ADDR OF MSG IN R0
11150
11200 :*****
11250 MTYPE: MOVB (R0)+,R2 ;GET CHAR TO PRINT
11300 BEQ 1$ ;EXIT IF NULL CHAR
JSR PC_MECHO
BR MTYPE ;GET NEXT CHAR
1$: RTS PC ;EXIT...

11400
11450
11460
11462
11464
11466 :*****
11468 : READIO THIS ROUTINE MONITORS AN I/O READ OPERATION
11470 :*****
11472
11474
READIO:
BIC #DATAIN,PCFLAG
MOV R2,DELAYT
MOV #101,@DLADR
DELAYR
: IF NO CHAR RECVD WITHIN (R2) MS SET ERROR FLAG
BIT #DATAIN,PCFLAG
BNE 50002$
BIS #MERR,MFLAGS(R5)
50002$:
: IF ON LINE=0 CLEAR I/O MODE FLAGS
TSTB ONLIN
BNE 50003$
BIC #FLAG1!#FLAG2,PCFLAG
BR 50004$
50003$: CLR @DLADR
50004$:
50000$:
50001$: RTS PC

11500
11502
11504 :*****
11506 : TYPES TERMINAL OUTPUT ROUTINE SINGLE LINE
11508 :*****
11510

```

```

1631
1632
1633 007110 112002
1634 007112 001403
1635 007114 004737 006654
1636 007120 000773
1637 007122 000207
1638
1639
1640
1641
1642
1643
1644
1645
1646 007124
1647 007124 105737 016160
1648 007130 001003
1649 007132 004737 005600
1650 007136 000410
1651 007140
1652 007140 010277 006766
1653 007144 013702 016132
1654 007150 062702 000002
1655 007154 012712 000200
1656 007160
1657 007160
1658 007160
1659 007160 000207
1660

```

```

11512
11514
11516
11518
11520
11522
11524
11526
11528
11530
11532
11533
11534
11536
11538
11554
11556

```

```

TYPES:  MOVB  (R0)+,R2      ;GET CHAR TO PRINT
        BEQ   1$           ;EXIT IF NULL
        JSR   PC,SECHO     ;SEND THE MESSAGE
        BR    TYPES
1$:      RTS    PC         ;EXIT

:*****
: READS  THIS ROUTINE SETS UP DVC RECV VECTOR AREAS
:        IF THE CURRENT LINE IS NOT LINE-CC
:*****

READS:  TSTB  ONLIN
        BNE  50002$
        JSR  PC,SETIO
        BR   50003$

50002$: MOV   R2,DLVEC
        MOV  DLVEC,R2
        ADD  #2,R2
        MOV  #PRI4,(R2)

50003$:
50000$:
50001$:
        RTS  PC

```

```

1659 11595
1660 11600
1661 11650
1662 11700
1663 11750
1664 11800
1665 007162 005077 006752 11875
1666 007166 012637 016132
1667 007172 062706 000002 11900
1668 007176 162737 000010 11950
1669 007204 005037 007564 12000
1670 007210 000002 12050
1671 12100
1672 12150
1673 12200
1674 12250
1675 12300
1676 12350
1677 12400
1678 007212 12450
(2) 007212 013746 016100
1679 007216 013746 016102
1680 007222 012702 016170
1681 007226 012700 017676
1682 007232 104000
1683 007234 005712
1684 007236 100406
1685 007240 062702 000010
1686 007244 021227 177777
1687 007250 001452
1688 007252 000770
1689 007254 012237 016100
1690 007260 012246
1691 007262 012746 000004
1692 007266 012746 017743
1693 007272 004737 007566
1694 007276 012246 13300
1695 007300 012746 000003 13400
1696 007304 012746 017752
1697 007310 004737 007566 13500
1698 007314 012237 016102 13550
1699 007320 000337 016102 13600
1700 007324 013746 016102
1701 007330 012746 000002
1702 007334 012746 017733
1703 007340 004737 007566 13800
1704 007344 012700 017733 13850
1705 007350 104000 13900
1706 007352 105737 016100 13950
1707 007356 001403 14000
1708 007360 012700 020117 14050
1709 007364 000402 14100
1710 007366 012700 020105 14150
1711 007372 104000 14200
1712 007374 000723 14250
1713 007376 14300

```

```

.SBTTL TRAP ROUTINES
:*****
:INTRAP: USED BY TABLE BUILD TO GET ADDRESS THAT A LINE
: INTERRUPTS TO AN STORE IT IN - DLVEC.
: TRANSMIT INTERRUPT USED, DLV HAS NO MAINT MODE.
:*****
INTRAP: CLR DDVCTXS ;DISABLE THE INTERRUPTS.
MOV (SP)+,DLVEC
ADD #2,SP ;SP+2 ADJUST STACK POINTER
SUB #10,DLVEC ;ADJUST TO RCVR INTR ADDR
CLR DELAYT ;RESET TIMER
RTI ;GO BACK TO BUILD ROUTINE

:*****
:PRTLTB THIS ROUTINE TYPES THE LINE TABLE ON THE CONSOLE
: DEVICE. DROPPED FLAGS ARE DECODED AND THE
: APPROPRIATE INFORMATION IS PRINTED FOR EACH LINE.
:*****
PRTLTB:
MOV TEMP, -(SP)
MOV TEMP+2, -(SP)
MOV #LIN00,R2 ; POINTER TO ; START OF TABLE
MOV #HEADR2,R0
TYPE ; PRINT HEADER
1$: TST (R2) ; LINE PRESENT?
BMI 2$ ; YES - BRANCH
ADD #10,R2 ; MOVE POINTER TO NEXT ENTR
6$: CMP (R2), #-1 ; END OF TABLE?
BEQ 10$
BR 1$ ; YES - BRANCH
2$: MOV (R2)+,TEMP ; SAVE FLAG WORD
MOV (R2)+, -(SP)
MOV #4, -(SP)
MOV #DLAD, -(SP) ; CONVERT ADDRESS TO ASCII
JSR PC, O2ASC
MOV (R2)+, -(SP)
MOV #3, -(SP)
MOV #DLV, -(SP) ; CONVERT LINE NO.
JSR PC, O2ASC
MOV (R2)+, TEMP+2
SWAB TEMP+2
MOV TEMP+2, -(SP)
MOV #2, -(SP)
MOV #LIN, -(SP)
JSR PC, O2ASC ; TYPE FORMATTED LINE
MOV #LIN,R0
TYPE
3$: TSTB TEMP ; SELECTED?
BEQ 4$ ; NO - BRANCH
MOV #S1,R0 ; SEND STAR
BR 5$
4$: MOV #DR,R0 ; SEND DROPPED MSG
TYPE
5$: BR 6$
10$:

```

1712	007376	012637	016102	MOV	(SP)+,TEMP+2
1714	007402	012637	016100	MOV	(SP)+,TEMP
1715	007406	012700	017051	MOV	#L3,R0
1716	007412	104000		TYPE	
1717	007414	000002		RTI	
1718			14400		
			14450		
			14500		
			14550		

```

1720      14600  ;*****
1721      14650  ;TXTRAP THIS ROUTINE CATCHES THE INTERRUPTS FROM
1722      14700  ;          DL11'S IN USE BY THE MECO ROUTINE .
1723      14750  ;*****
1727      14950
1728 007416 162716 000010 TXTRAP: SUB #10,(SP) ;SUB 10 FROM UPDATED PC ON STACK
1729 007422 011614 15000 MOV (SP),(R4) ;PUT BASE VECTOR INTO STACK3
1730 007424 012746 016174 MOV #LIND0+4,-(SP) ;GET POINTER TO LINE TABLE VECTORS
1731 007430 027614 000000 1$: CMP 00(SP),(R4) ;COMPARE TABLE TO STACK3
1732 007434 001403 15080 BEQ 2$ ;SAME - BRANCH
1733 007436 062716 000010 15100 ADD #10,(SP) ;POINT TO NEXT TABLE ENTRY
1734 007442 000772 15120 BR 1$ ;KEEP LOOKING FOR A MATCH
1735 007444 162716 000002 2$: SUB #2,(SP) ;ADDR OF DLADR NOW ON STACK
1736 007450 017637 000000 016100 15140 MOV 00(SP),TEMP ;GET DLADR FROM TABLE
1737 007456 062737 000004 016100 15160 ADD #4,TEMP ;POINT TO DVCTXS REGISTER
1738 007464 005077 006410 15180 CLR #TEMP ;DISABLE INTERRUPT S
1739 007470 062706 000006 15200 ADD #6,SP ;SET STACK POINTER TO DRIVER PC
1740 007474 005337 020670 15220 DEC ENDS ;DECRIMENT INTERRUPT PENDING COUNT
1741 007500 003002 15240 BGT 3$
1742 007502 005037 007564 15260 CLR DELAYT ;ABORT TIMEOUT IF ALL ACCOUNTED FOR
1743 007506 000002 15280 3$: RTI ;RETURN TO I/O DRIVER
1744 15300
1745 15500
1746 15501
1747 15502
1748 15503
1749 15504
1750 15505 ;*****
1751 15506 ; STRAP SINGLE LINE TRANSMIT INTERRUPT CATCHER
1752 15507 ; USED IN CONJUNCTION WITH SECHO ROUTINE.
1753 15508 ;*****
1754 15509
1755 007510
1756 007510 005077 006424 STRAP: CLR #DVCTXS
1757 007514 005337 020670 DEC ENDS
1758 007520 005037 007564 CLR DELAYT
1759 007524 000002 15515 RTI
1760 007526
1761 (3) 007526 50000$:
1762 (2) 007526 50001$:
1763 000207 RTS PC
1764 15517
1765 15518
1766 15519
1767 15550
1768 15750 ;*****
1769 15800 ;DELAYM, DELAYS FOR X MILLI SECONDS, X STORED IN - DELAYT
1770 15850 ;*****
1771 15900
1772 15950
1773 007530 005737 007564 DELAYM: TST DELAYT
1774 007534 001411 15955 BEQ 3$
1775 007536 010346 15960 MOV R3,-(SP) ;
(2) .MEXIT ;
1776 007540 013703 007562 1$: MOV TIMER,R3 ;1MS LOOP TIME
1777 007544 005303 16100 2$: DEC R3
1778 007546 001376 16150 BNE 2$

```

K05

CZLAFAD LA36 TERM TST MACY11 30A(1052) 03-JAN-77 00:01 PAGE 2-11
CZLAFAP11 03-JAN-78 11:20 TRAP ROUTINES

SEQ 0062

1779	007550	005337	007564	16200		DEC	DELAYT
1780	007554	003371		16250		BGT	1\$
1781	007556	012603				MOV	(SP)+R3
(2)							.MEXIT
1782	007560	000002		16350	3\$:	RTI	
1783				16400			
1784	007562	000554		16450	TIMER:	.WORD	554
1785				16500			
1786				16550			
1787				16600			
1788				16650			
1789				16700			
1790				16750			
1791	007564	000000		16800	DELAYT:	.WORD	0

```

:SET FOR 11/35 - 11/40
:SET TO 202 IF 11/03
:      251   11/05 - 11/10
:      314   11/15 - 11/20
:      2127  11/45 BIPOLAR
:      1237  11/45 - 11/70
:      755   11/45 MOS
:DELAY TIME BUF R

```

.SBTL CONVERSION ROUTINES

1793				16900
1794				16950
1795				17000
1796				17050
1797				17100
1798				17150
1799				17200
1800				17250
1801	007566	016637	000006	016150 17300
1802	007574	013746	016150	17350
1803	007600	066666	000006	000004 17400
1804	007606	005366	000004	17450
1805	007612	042716	177770	17500
1806	007616	052716	000060	17550
1807	007622	111676	000004	17600
1808	007626	005366	000004	17650
1809	007632	005366	000006	17700
1810	007636	001411		17750
1811	007640	006266	000010	17800
1812	007644	006266	000010	17850
1813	007650	006266	000010	17900
1814	007654	016616	000010	17950
1815	007660	000754		18000
1816	007662	016666	000002	000010 18050
1817	007670	062706	000010	18100
1818	007674	000207		18150
1819				18200
1820				18250
1821				18300
1822				18350
1823				18400
1824				18450
1825				18500
1829				18700
1830	007676			
1831	007676	010046		
1832	007700	005037	010004	
1833	007704	016600	000010	
1834	007710			
1835	007710	142710	000370	
1836	007714	005366	000006	
1837	007720	152037	010004	
1838	007724	005766	000006	
(5)	007730	001407		
1839	007732	006337	010004	
(7)	007736	006337	010004	
(7)	007742	006337	010004	
1840	007746	000760		
(3)	007750			
1841	007750	016600	000004	
1842	007754	013710	010004	
1843	007760	011637	010004	
1844	007764	016600	000002	
1845	007770	062706	000010	
1846	007774	010016		
1847	007776	013700	010004	

```

:O2ASCI OCTAL TO ASCII CONVERSION ROUTINE - ENTER WITH
:NUMBER TO BE CONVERTED ON THE STACK, FOLLOWED
:BY THE NUMBER OF DIGITS TO CONVERT, FOLLOWED
:BY THE STORAGE ADDRESS FOR THE ASCII STRING.
:*****
O2ASC: MOV 6(SP),SAVE ;GET WORK COPY OF NUMBER
MOV SAVE, -(SP)
ADD 6(SP),4(SP) ;ADD COUNT TO POINTER
DEC 4(SP) ;DEC FOR END ADDR
2$: BIC #177770,(SP) ;MASK OUT ALL BUT 3 BITS
BIS #60,(SP) ;MAKE CHAR ASCII
MOV 4(SP),24(SP) ;PUT ASCII CHAR IN BUFFER
DEC 4(SP) ;INC POINTER
DEC 6(SP) ;DEC DIGIT COUNT
BEQ 1$ ;BRANCH IF DONE
ASR 10(SP)
ASR 10(SP) ;GET NEXT DIGIT
ASR 10(SP)
MOV 10(SP),(SP)
BR 2$ ;DO NEXT CHAR FOR CONVERSION
1$: MOV 2(SP),10(SP) ;PUT RETURN PC AT TOP OF JUNK
ADD #10,SP ;POINT TO RETURN PC
RTS PC ;EXIT...

```

```

:*****
:A2BIN CONVERTS INPUT ASCII TO BINARY NUMBER
:ENTER WITH ADDR OF ASCII STRING ON STACK
:FOLLOWED BY # DIGITS TO CONVERT
:FOLLOWED BY ADDR OF WORD FOR ANSWER.
:*****

```

```

A2BIN: MOV R0, -(SP)
CLR A2SAV
MOV 10(SP),R0
50002$: BICB #370,(R0)
DEC 6(SP)
BISB (R0)+,A2SAV
TST 6(SP)
BEQ 50003$
ASL A2SAV
ASL A2SAV
ASL A2SAV
BR 50002$
50003$: MOV 4(SP),R0
MOV A2SAV,(R0)
MOV (SP),A2SAV
MOV 2(SP),R0
ADD #10,SP
MOV R0,(SP)
MOV A2SAV,R0

```



```

1848 010002
(3) 010002
(2) 010002 000207
1849
1850 010004 000000
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862 010006 012700 010146
1863 010012 112737 000005 010162
1864 010020 005037 010160
1865 010024 021066 000004
1866 010030 003005
1867 010032 161066 000004
1868 010036 105237 010160
1869 010042 000770
1870 010044 152737 000060 010160
1871 010052 105737 010161
1872 010056 001012
1873 010060 123727 010160 000060
1874 010066 001004
1875 010070 112737 000177 010160
1876 010076 000402
1877 010100 105137 010161
1878 010104 113776 010160 000002
1879 010112 005266 000002
1880 010116 062700 000002
1881 010122 105037 010160
1882 010126 105337 010162
1883 010132 001334
1884 010134 011666 000004
1885 010140 062706 000004
1886 010144 000207
1887
1888 010146 023420 001750 000144
010154 000012 000601
1889 010160 000
1890 010161 000
1891 010162 000 000
1892
1893

```

```

50000$:
50001$:
RTS PC
A2SAV: .WORD 0 ; STORAGE AREA

:*****
:BIN2DA BINARY TO DECIMAL ASCII CONVERSION ROUTINE
: ENTER WITH NUMBER TO CONVERT ON THE STACK,
: FOLLOWED BY THE ADDRESS OF THE ASCII BUFFER.
: 5 DIGITS WILL BE CONVERTED
:*****

BIN2DA: MOV #TABDA,RO ; INITIALIZE TABLE POINTER
MOV #5,DIGITS
CLR CNTDA
1$: CMP (RO),4(SP)
BGT 2$
SUB (RO),4(SP)
INCB CNTDA
BR 1$
2$: BISH #60,CNTDA
TSTB FLAGDA
BNE 4$
CMPB CNTDA,#'0
BNE 3$
MOVB #177,CNTDA
BR 4$
3$: COMB FLAGDA
4$: MOVB CNTDA,2(SP)
INC 2(SP)
ADD #2,RO
CLRB CNTDA
DECB DIGITS
BNE 1$
MOV (SP),4(SP)
ADD #4,SP
RTS PC

TABDA: .WORD 10000.,1000.,100.,10.,1

CNTDA: .BYTE 0
FLAGDA: .BYTE 0
DIGITS: .BYTE 0,0

```

N05

1898				00250
1899				00300
1900				00350
1901				00400
1902				00450
1903				00500
1904				00550
1905	010164	012705	010352	00600
1906	010170	012700	010366	00650
1907	010174	004737	007010	
1908	010200	012700	010424	00750
1909	010204	004737	007010	
1910	010210	004737	010310	00850
1911	010214	012700	017730	00900
1912	010220	004737	007010	
1913	010224	012700	010434	01000
1914	010230	004737	007010	
1915	010234	012702	000016	01100
1916	010240	004737	006270	
1917	010244	004737	010310	01200
1918	010250	012700	017730	01250
1919	010254	004737	007010	
1920	010260	012702	000017	01350
1921	010264	004737	006270	
1922				01450
1923	010270	052765	020000	000004 01500
1924	010276	012702	000012	01550
1925	010302	004737	006270	
1926	010306	000207		01650
1927				01700
1928				01750
1929				01800
1930	010310	013701	016146	01850
1931	010314	012702	000040	01900
1932	010320	162701	000007	01950
1933	010324			02000
(3)	010324	004737	006270	
1934	010330	005202		02050
1935	010332	020237	010364	02100
1936	010336	001403		02150
1937	010340	005301		02200
1938	010342	001401		02250
1939	010344	000767		02300
1940	010346	000207		02350
1941	010350	000006		02400
1942	010352	000000		02450
1943	010354	000000		02500
1944	010356	000000		02550
1945	010360	000000		02600
1946	010362	010200		02650
1947	010364	000177		02700
1948				02750
1949	010366	042524	052123	030040 02800
1950	010424	051501	044503	026511 02850
1951	010434	050101	026514	026455 02900
1952				02950

```

.SBTTL LA36 OPTION TESTS
*****
:TESTC SECONDARY CHARACTER SET OPTION
:      NO MANUAL INTERVENTION REQUIRED
*****
TESTO:  MOV      #TOOBLK,R5          ;SET UP POINTER TO MODULE BLOCK
        MOV      #TO,RO             ;SU TEST ID
        JSR      PC,MTYPE
TO1:    MOV      #PRI,RO            ;SU PRIMARY MSG
        JSR      PC,MTYPE
        JSR      PC,CHARS          ;SEND ALL CHARACTERS
        MOV      #L1,RO
        JSR      PC,MTYPE
        MOV      #SEC,RO           ;SU SECONDARY MSL.
        JSR      PC,MTYPE
        MOV      #EO,R2            ;SEND SO - SELECT APL SET
        JSR      PC,MECHO
        JSR      PC,CHARS          ;SEND ALL CHARS AGAIN
        MOV      #L,RO
        JSR      PC,MTYPE
        MOV      #S1,R2            ;SEND SI-SELECT ASCII
        JSR      PC,MECHO
        BIS      #TCONE,MFLAGS(R5) ;SET DONE AND ATTENTION FLAGS
        MOV      #1?,R2            ;SU FOR LF
        JSR      PC,MECHO
        RTS      PC
*****
: SUBROUTINE TO FILL OUTPUT LINE WITH ALL CHARACTERS
CHARS:  MOV      WIDTH,R1           ;SAVE WIDTH
        MOV      #40,R2            ;SAVE START CHAR
        SUB      #7,R1              ;ADJUST WIDTH FOR PRI/SEC MSG
2$:     JSR      PC,MECHO
        INC      R2                 ;NEXT CHAR
        CMP      R2,RUB            ;LAST CHAR?
        BEQ     3$                 ;YES - EXIT
        DEC     R1                 ;END OF PAPER?
        BEQ     3$                 ;YES - EXIT
        BR      2$                 ;SEND NEXT
3$:     RTS      PC
        .WORD   6                   ;ITERATION COUNT
TOOBLK: .WORD   0                   ;CTLCNT
        .WORD   0                   ;PASS COUNT
        .WORD   0                   ;STATUS FLAGS
        .WORD   0                   ;POINTER
        .WORD   TO1                 ;RETURN PC
RUB:    .WORD   177
.NLIST BEX
TO:     .ASCIZ  *TEST 0 APL/ASCII CHAR SET*(15)<(12)<(12)
PRI:    .ASCIZ  /ASCII--/
SEC:    .ASCIZ  /APL----/
        .EVEN

```

1954				03050
1955				03100
1956				03150
1957				03200
1958				03250
1959				03300
1960				03350
1961				03400
1962				03450
1963				03500
1964				03550
1965				03600
1966	010444			
1967	010444	012705	010722	03700
1968	010450	012700	010742	03750
1969	010454	004737	007010	
1970	010460			03850
1971				03900
1972				03950
1973				04000
1974				04050
1975				04100
1976	010460	012765	010566	000010
1977	010466	013701	010734	
1978	010472	012737	011176	010740
1979	010500	012737	000001	010736
(5)	010506	000402		
(4)	010510			
(7)	010510	005237	010736	
(5)	010514			
(5)	010514	023727	010736	000010
(7)	010522	003014		
1980	010524	012700	017730	
1981	010530	004737	007010	
1982	010534	017700	000200	
1983	010540	004737	007010	
1984	010544	062737	000002	010740
1985	010552	000756		
(3)	010554			
1986				04650
1987				04700
1988	010554			04750
(4)	010554	012765	010460	000010
1989	010562	013701	010734	
1990	010566			04850
1991	010566			
1992	010566	020127	000200	
(5)	010572	001420		
1993	010574	012700	020421	
1994	010600	110160	000003	
1995	010604	004737	007010	
1996	010610	112702	000002	
1997	010614	004737	006270	
1998				05250
1999	010620	004737	010662	
2000	010624	004737	007010	

```

.LIST BEX
*****
TEST1 SELECTIVE ADDRESSING OPTION
      OPERATOR MUST COMPARE TYPEOUT AND SWITCHES ON THE M7737
      TO VERIFY CORRECT OPERATION.
      IF A GROUP OR UNIT SELECT CODE OF LESS THAN 20(8)
      IS USED MODIFY LOCATION GSEL ACCORDINGLY.
*****

TEST1:
      MOV     #T01BLK,R5      ;SET UP POINTER TO MODULE BLOCK
      MOV     #T1,R0
      JSR     PC,MTYPE
T11:    ;DESELECT ALL TERMINALS, THEN TRY TO
      ;PRINT ERROR MESSAGES...SHOULD NOT PRINT
      ;TRANSMIT A BAD SELECT SEQUENCE, THEN TRY TO
      ;PRINT ERROR MESSAGES...SHOULD NOT PRINT
      ;SELECT ALL TERMINALS, PRINT GP MESSAGE.

      MOV     #T13,RPC(R5)
      MOV     GSEL,R1
      MOV     #TABL1,T1TEMP+2
      MOV     #1,T1TEMP
      BR      50002$

50003$: INC     T1TEMP
50002$:
      CMP     T1TEMP,#8.
      BGT     50004$
      MOV     #L1,R0
      JSR     PC,MTYPE
      MOV     #T1TEMP+2,R0
      JSR     PC,MTYPE
      ADD     #2,T1TEMP+2
      BR      50003$

50004$: ; TRANSMIT SELECT CODES TO ALL TERMINALS
      ; FOLLOWED BY ASCII EQUIV OF CODE.
T12:    MOV     #T11,RPC(R5)
      MOV     GSEL,R1
T13:    ;OUTPUT ALL CODES AND ACCII EQUIVELANTS
50005$:
      CMP     R1,#200
      BEQ     50006$
      MOV     #SCODE,R0
      MOV     R1,3(R0)
      JSR     PC,MTYPE
      MOV     #STX,R2
      JSR     PC,MECHO
      ; NOW CONVERT SELECT CODE TO ASCII FOR OUTPUT
      JSR     PC,CON
      JSR     PC,MTYPE

```

```

2001 010630 005201
2002 010632 000755
(3) 010634
2003 05500
2004 010634 05550
(4) 010634 012765 010460 000010
2005 010642 052765 020000 000004
2006 010650 012700 020411
2007 010654 004737 007010
2008 010660
(3) 010660
(2) 010660 000207
2009 05750
2010 05800
2011 010662
2012 010662 005037 010736
2013 010666 110137 010736
2014 010672 013746 010736
2015 010676 012746 000003
2016 010702 012746 011520
2017 010706 004737 007566
2018 010712 012700 011520
2019 010716
(3) 010716
(2) 010716 000207
2020 06300
2021 010720 000002 06350
2022 010722 000000 06400
2023 010724 000000 06450
2024 010726 000000 06500
2025 010730 000000 06550
2026 010732 010460 06600
2027 010734 000020 06650
2028 010736 000000 000000 06700
2029 06750
2030 010742 005015 052012 051505 06800
2031 011005 105 051122 051117 06850
2032 011045 116 020117 042523 06900
2033 011100 042523 042514 052103 06950
2034 011137 101 046114 052040 07000
2035 011176 011176 07050
2036 011176 020416 011005 011137 07100
2037 07150
2038 07200
2039 07250
2040 07300
2041 07350
2042 07400
2043 07450
2044 07500
2045 07550
2046 07600
2047 07650
2048 011216
2049 011216 010337 011304
2050 011222 012702 013144

```

```

INC R1
BR 50005$
50006$: ; TURN ALL TERMINALS ON AND EXIT TEST
T16:
MOV #T11,RPC(R5)
BIS #TDONE,MFLAGS(R5)
MC #ALLON,RO
JSR PC,MTYPE
50000$:
50001$: RTS PC
; THIS ROUTINE CONVERTS THE SELECT CODE
; TO ASCII FOR OUTPUT IN OCTALC MESSAGE.
CON:
CLR TITEMP
MOV R1,TITEMP
MOV TITEMP, -(SP)
MOV #3, -(SP)
MOV #OCTALC, -(SP)
JSR PC,C3ASC
MOV #OCTALC,RO
50000$:
50001$: RTS PC
TC1BLK: .WORD 2 ; ITERATION COUNT
; CTLCNT
; PASS COUNT
; STATUS FLAGS
; POINTER
; RETURN PC
; START OF SELECT CODES
GSEL: .WORD 20
TITEMP: .WORD 0,0
.NLIST BEX
T1: .ASCIZ '<15><12><12>/TEST 1 SELECTIVE ADDRESSING 15 12 12'
E9: .ASCIZ '/ERROR - THIS SHOULD NOT PRINT *'
E12: .ASCIZ '/NO SELECT CHARACTER SENT<<15><12>'
GP: .ASCIZ '/SELECT CHARACTERS RECOGNIZED =/'
E10: .ASCIZ '/ALL TERMINALS SHOULD BE OFF//15 <12/'
.EVEN
TABL1: .WORD ALLOFF,E9,E10,NSELC,E9,E12,ALLON,GP
;*****
;GETANS
;*****
THIS ROUTINE SETS UP AND READS THE ANSWERBACK
MESSAGE FROM THE TERMINAL UNDER TEST.
GETANS:
MOV R3,25
MOV #T2E0,R2

```

2051	011226	004737	007124			JSR	PC, READS
2052	011232	012702	000005			MOV	#ENG, R2
2053	011236	004737	006654			JSR	PC, SECHO
2054	011242			08000	13:		
(4)	011242	013702	011304			MOV	25, R2
2055	011246	004737	007024			JSR	PC, READIO
2056	011252	032765	100000	000004		BIT	#MERR, MFLAGS, R5
(9)	011260	001405				BEQ	500025
2057	011262	042765	100000	000004		BIC	#MERR, MFLAGS, R5
2058	011270	105011				CLRB	(R1)
2059	011272	000403				BR	500035
(3)	011274				500025:		
2060	011274	105237	013212			INCB	T2CNT1
2061	011300	000760		08350		BR	15
2062	011302				500035:		
2063	011302				500005:		
(3)	011302				500015:		
(2)	011302	000207				RTS	PC
2064	011304	000000			25:	.WORD	0
2065				08500			
2066				08550			
2067				08600			
2068				08650			
2069				08700			
2070				08750			
2071				08800			
2072				08850			
2073				08900			
2074	011306			09000			
2075	011306	012700	020421			MOV	#SCODE, R0
2076	011312	004737	007110			JSR	PC, TYPES
2077	011316	012702	000002			MOV	#STX, R2
2078	011322	004737	006654			JSR	PC, SECHO
2079	011326	012700	011472			MOV	#ANSHOR, R0
2080	011332	004737	007110			JSR	PC, TYPES
2081	011336	013746	013212			MOV	T2CNT1, -(SP)
2082	011342			09337	15:		
(4)	011342	005046				CLR	-(SP)
2083	011344	112116				MOVB	(R1)+, (SP)
2084	011346	012746	000003			MOV	#3, -(SP)
2085	011352	012746	011520			MOV	#OCTALC, -(SP)
2086	011356	004737	007566			JSR	PC, O2ASC
2087	011362	012700	011520			MOV	#OCTALC, R0
2088	011366	004737	007110			JSR	PC, TYPES
2089	011372	105337	013212			DECB	T2CNT1
2090	011376	105737	013212			TSTB	T2CNT1
(9)	011402	003402				BLE	500025
2091	011404	000756		09700		BR	15
2092	011406	000426				BR	500035
(3)	011410				500025:		
2093	011410	012700	017730			MOV	#L1, R0
2094	011414	004737	007110			JSR	PC, TYPES
2095	011420	012700	000023			MOV	#19, R0
2096	011424				500045:		
2097	011424	012702	000040			MOV	#40, R2
2098	011430	004737	006654			JSR	PC, SECHO

```

:*****
:TYPANS THIS ROUTINE PRINTS THE ANSWERBACK MESSAGE
: IN OCTAL FORMAT , AND ASCII FORMAT.
:*****

```

TYPANS:

2099	011434	005300					DEC	RO
2100	011436	005700					TST	RO
(5)	011440	001401					BEG	50005\$
2101	011442	000770					BR	50004\$
(3)	011444					50005\$:		
2102	011444	012700	013220				MOV	#T2BUF,RO
2103	011450	004737	007110				JSR	PC,TYPES
2104	011454	012700	017730				MOV	#L1,RO
2105	011460	004737	007110				JSR	PC,TYPES
2106	011464					50003\$:		
2107	011464	012637	013212				MOV	(SP)+,T2CNT1
2108	011470					50000\$:		
(3)	011470					50001\$:		
(2)	011470	000207					RTS	PC
2109	011472	005015	047101	053523	10100	ANSHDR:	.ASCIZ	<15><12> ANSWERBACK RECVD =
2110	011520	030060	027460	JOC	10150	OCTALC:	.ASCIZ	*000/*
2111		011526			10200		.EVEN	
2112					10250			

```

114 10350
115 10400
116 10450
117 10500
118 10550
119 10600
120 10650
121 10700
122 10750
123
124 011526 012700 013326 10850
125 011526 012701 013220 10900
126 011536 012705 013176 10950
127 011542 004737 007010
128 011546
129
130
131 011546 105737 016134 11050
132 011552 001065
133 011554 012701 013220 11100
134
135
136
137 011560 013737 010734 013214 11150
138 011566 013737 016160 013216
139 011574 006337 013216
140 011600 006337 013216
141 011604 006337 013216
142 011610 062737 016176 013216
143 011616 112777 000200 001372
144 011624
145 011624 023727 013214 000200
146 011632 001435
147
148
149 011634 113737 013214 020424 11800
150 011642 012700 020421 11850
151 011646 004737 007110 11900
152 011652 012703 000310 11950
153 011656 105037 013212 12000
154 011662 004737 011216
155 011666 105737 013212
156 011672 001412
157 011674 113777 013214 001314
158 011702 113737 013214 016134
159 011710 012737 000200 013214
160 011716 000402
161 011720
162 011720 105237 013214
163 011724
164 011724 000737
165 011726

```

```

.LIST BEX
*****
TEST2 AUTO ANSWER BACK OPTION
SINGLE LINE TESTS REQUIRE MANUAL INTERVENTION
*****
.ENABL LSB
TEST2:
MOV #T2,R0
MOV #T2BUF,R1 ;SET UP STACK-2 AS INPUT BUFFER
MOV #T02BLK,R5 ;SET UP POINTER TO MODULE BLOCK
JSR PC,MTYPE
T21: ;IF THE LINE UNDER TEST HASN'T BEEN SIZED
;FOR THE ANSWERBACK OPTION DO SO NOW.
TSTB DLOTH
BNE 50002$
MOV #T2BUF,R1
;CHECK DLOTH ENTRY OF LINE TABLE FOR CURRENT
;LINE. IF LOBYTE IS 0 NO SIZE HAS BEEN DONE.
;IF = 200 LINE SIZED BUT NO ANSWER RECVD.
MOV GSEL,T2TEMP
MOV ONLIN,T2TEMP+2
ASL T2TEMP+2
ASL T2TEMP+2
ASL T2TEMP+2
ADD #LIN00+6,T2TEMP+2
MOVB #200,T2TEMP+2
50003$:
CMP T2TEMP,#200
BEQ 50004$
;SEND EACH POSSIBLE SELECT CODE TO THE
;TERMINAL, THEN REQUEST AN ANSWERBACK.
;IF AN ANSWER IS REVIEWED STORE THE SELECT
;CODE IN DLOTH ENTRY OF THE LINE TABLE.
;OTHERWISE SET DLOTH TO 200.
MOVB T2TEMP,SCODE+3
MOV #SCODE,R0
JSR PC,TYPE5
MOV #200,R3
CLRB T2CNT1
JSR PC,GETANS
TSTB T2CNT1
BEQ 50005$
MOVB T2TEMP,T2TEMP+2
MOVB T2TEMP,DLOTH
MOV #200,T2TEMP
BR 50006$
50005$:
INCB T2TEMP
50006$:
BR 50003$
50004$:

```

2162	011726				
2163					12600
2164	011726				12650
2165					12700
2166					12750
2167					12800
2168	011726	123727	016134	000200	
2169	011734	001015			
2170					12900
2171					12950
2172					13000
2173	011736	012765	012530	000010	
2174	011744	052765	100000	000004	
2175	011752	105065	000004		
2176	011756	012765	013356	000006	
2177	011764	000207			13250
2178	011766	000463			
2179	011770				
2180	011770	012701	013220		
2181					13400
2182					13450
2183					13500
2184	011774	113737	016134	020424	
2185	012002	012700	020421		
2186	012006	004737	007110		
2187	012012	012703	000310		
2188	012016	105037	013212		
2189	012022	004737	011216		
2190					13850
2191	012026	105737	013212		
2192	012032	001015			
2193					14000
2194					14050
2195					14100
2196	012034	012765	012530	000010	
2197	012042	052765	100000	000004	
2198	012050	112765	000001	000004	
2199	012056	012765	013356	000006	
2200	012064	000424			
2201	012066				
2202	012066	012765	012140	000010	
2203					14450
2204					14500
2205					
2206	012074	123727	013212	000024	
2207	012102	003411			
2208					14600
2209					14650
2210					14700
2211	012104	052765	100000	000004	
2212	012112	112765	000002	000004	
2213	012120	012765	013404	000006	
2214	012126				
2215	012126	012701	013220		
2216	012132	004737	011306		
2217	012136				

50002\$:

T22: ; IF THE LINE HAS BEEN SIZED, BUT NO
; SELECT CODE HAS BEEN MAPPED NOTIFY THE
; OPERATOR .

CMPE DLOTH,#200
BNE 50007\$
; ERROR #0 NO ANSWERBACK FROM TERMINAL
; *****

MOV #T24,RPC(R5)
BIS #MERR,MFLAGS(R5)
CLRB MFLAGS(R5)
MOV #E14,POINT(R5)
RTS PC
BR 50010\$

50007\$:

MOV #T2BUF,R1
; GET THE SELECT CODE FROM THE LINE TABLE 3
; REQUEST AN ANSWERBACK.

MOVB DLOTH,SCODE+3
MOV #SCODE,R0
JSR PC,TYPES
MOV #200,R3
CLRB T2CNT1
JSR PC,GETANS
; CHECK FOR ANY RESPONSE FROM TERMINAL

TSTB T2CNT1
BNE 50011\$
; ERROR #1 NO ANSWERBACK RECIEVED.
; *****

MOV #T24,RPC(R5)
BIS #MERR,MFLAGS(R5)
MOVB #1,MFLAGS(R5)
MOV #E14,POINT(R5)
BR 50012\$

50011\$:

MOV #T23,RPC(R5)
; TEST LENGTH OF ANSWERBACK SHOULD BE 20 MAX.

CMPE T2CNT1,#20.
BLE 50013\$
; ERROR #2 ANSWERBACK OVER 20 CHARS LONG.
; *****

BIS #MERR,MFLAGS(R5)
MOVB #2,MFLAGS(R5)
MOV #E15,POINT(R5)

50013\$:

MOV #T2BUF,R1
JSR PC,TYPANS

50012\$:

2213	012136			
2214	012136	000207		
2215	012140			
2216				
2217				
2218				
2219	012140	012765	012206	000010
2220	012146	113737	013212	013213
2221	012154	012701	013220	
2222	012160	012700	020574	
2223	012164			
2224				
2225	012164	105737	013212	
(5)	012170	001404		
2226	012172	112120		
2227	012174	105337	013212	
2228	012200	000771		
(3)	012202			
2229	012202	105037	013210	
2230				
2231	012206			
2232				
2233	012206	012701	013220	
2234	012212	105037	013212	
2235				
2236				
2237	012216	012700	020421	
2238	012222	012703	000310	
2239	012226	004737	011216	
2240	012232	105237	013210	
2241				
2242				
2243				
2244				
2245	012236	105737	013212	
(9)	012242	001012		
2246				
2247				
2248				
2249	012244	052765	100000	000004
2250	012252	012765	013356	000006
2251	012260	112765	000030	000004
2252	012266	000500		
(3)	012270			
2253	012270	105011		
2254				
2255	012272	123737	013212	013213
(9)	012300	001416		
2256	012302	012701	013220	
2257	012306	004737	007110	
2258				
2259				
2260				
2261	012312	052765	100000	000004
2262	012320	112765	000031	000004
2263	012326	012765	013356	000006

```

500105:
RTS PC
T23: ;SAVE COPIES OF THE ANSWERBACK AND IT'S LENGTH
;THEN READ ANSWERBACKS 10 TIMES MORE.
;VERIFY THEY ARE ALL THE SAME.
MOV #T23A,RPC(R5)
MOV T2CNT1,T2CNT2
MOV #T2BUF,R1
MOV #STACK3,RO

500145:
;COPY ANSWERBACK TO STACK3 FOR COMPARISONS
TSTB T2CNT1
BEQ 500155
MOV (R1)+,(RO)+
DECB T2CNT1
BR 500145

500155:
CLRB T2SAV1

T23A: ;RESET INPUT BUFFER POINTER AND ZERO COUNTER
MOV #T2BUF,R1
CLRB T2CNT1
;SEND SELECT SEQUENCE TO TERMINAL
;THEN READ ANSWER
MOV #SCODE,RO
MOV #200,R3
JSR PC,GETANS
INCB T2SAV1
;IF NO ANSWER NOTIFY OPERATOR
;IF OLD ANSWER DIFFERENT FROM NEW ANSWER
;NOTIFY OPERATOR.
TSTB T2CNT1
BNE 500165
; ERROR #30 NO ANSWERBACK DURING TEN READ LOOP
;*****
BIS #MERR,MFLAGS(R5)
MOV #E14,POINT(R5)
MOV #30,MFLAGS(R5)
BR 500175

500165:
CLRB (R1)
;COMPARE LENGTHS OF ANSWERS
CMPB T2CNT1,T2CNT2
BEQ 500205
MOV #T2BUF,R1
JSR PC,TYPE5
;ERROR #31 INCONSISTANT ANSWERBACKS
;*****
BIS #MERR,MFLAGS(R5)
MOV #31,MFLAGS(R5)
MOV #E14,POINT R5

```

2264	012334	000455				50020\$:	BR	50021\$
(3)	012336						MOV	#T2BUF,R1
2265	012336	012701	013220				MOV	#STACK3,RO
2266	012342	012700	020574				;COMPARE MASSAGES FOR SAME DATA	
2267					17800			
2268					17850			
2269	012346	005037	013214				CLR	T2TEMP
2270	012352	113737	013212	013214			MOV	T2CNT1,T2TEMP
2271	012360	005037	013216				CLR	T2TEMP+2
2272	012364	012737	000001	013216			MOV	#1,T2TEMP+2
(5)	012372	000402					BR	50022\$
(4)	012374					50023\$:	INC	T2TEMP+2
(7)	012374	005237	013216			50022\$:	CMP	T2TEMP+2,T2TEMP
(5)	012400						BGT	50024\$
(5)	012400	023737	013216	013214			CMPB	(R0)+,(R1)+
(7)	012406	003024					BEQ	50025\$
2273	012410	122021					;ERROR #32 INCONSISTANT ANSWERBACKS	
(9)	012412	001421					;*****	
2274					18150			
2275					18200			
2276					18250			
2277	012414	052765	100000	000004			BIT	#MERR,MFLAGS(R5)
2278	012422	112765	000032	000004			MOV	#32,MFLAGS(R5)
2279	012430	012765	013575	000006			MOV	#E21,POINT(R5)
2280	012436	012701	013220				MOV	#T2BUF,R1
2281	012442	113737	013213	013212			MOV	T2CNT2,T2CNT1
2282	012450	013737	013214	013216			MOV	T2TEMP,T2TEMP+2
2283	012456					50025\$:	BR	50023\$
2284	012456	000746				50024\$:	;ECHO ANSWER TO TERMINAL IN ASCII AND	
(3)	012460						;OCTAL FORMATS.	
2285					18700		MOV	#T2BUF,R1
2286					18750		JSR	PC,TYPANS
2287	012460	012701	013220			50021\$:		
2288	012464	004737	011306			50017\$:		
2289	012470						BIT	#MERR,MFLAGS(R5)
2290	012470						BEQ	50026\$
2291	012470	032765	100000	000004			RTS	PC
(9)	012476	001401				50026\$:	;CHECK FOR TEN ITERATIONS	
2292	012500	000207			19050		CMPB	T2SAV1,#10
2293	012502						BNE	50027\$
2294							MOV	#T24,RPC(R5)
2295	012502	123727	013210	000010			RTS	PC
(9)	012510	001005					BR	50030\$
2296	012512	012765	012530	000010		50027\$:	JMP	T23A
2297	012520	000207			19300	50030\$:		
2298	012522	000402				T24:	;RESTORE POINTERS & TEST THE BROADCAST (BEL)	
(3)	012524						;WON'T ACTIVATE THE AUTOANSWER.	
2299	012524	000137	012206		19400			
2300	012530							
2301					19500			
2302	012530				19550			
2303					19600			
2304					19650			
2305	012530	012701	013220				MOV	#T2BUF,R1
2306	012534	105037	013212				CLRB	T2CNT1
2307	012540	012702	000002				MOV	#STX,R2

2308	012544	004737	006654			JSR	PC, SECHO
2309	012550	012700	020421			MOV	#SCODE, R0
2310	012554	112737	000007	020424		MOVB	#7, SCODE+3
2311	012562	004737	007110			JSR	PC, TYPES
2312	012566	012703	000310			MOV	#200, R3
2313	012572	004737	011216			JSR	PC, GETANS
2314	012576	105737	013212			TSTB	T2CNT1
(9)	012602	001415				BEQ	50031\$
2315				20050		; ERROR #4	RECVD ANSWERBACK FROM BROADCAST
2316				20100		; *****	
2317				20150			
2318	012604	052765	100000	000004		BIS	#MERR, MFLAGS(R5)
2319	012612	112765	000004	000004		MOVB	#4, MFLAGS(R5)
2320	012620	012765	013632	000006		MOV	#E22, POINT(R5)
2321	012626	012765	012636	000010		MOV	#T25, RPC(R5)
2322	012634	000207			20400	RTS	PC
2323	012636						
2324	012636				20500	50031\$:	
2325					20550	T25:	; IF IN MULTI LINE MODE SETUP NEXT LINE POINTERS
2326					20600		; IF SINGLE LINE MODE TEST KEYBOARD STUFF.
2327	012636	032737	000040	001364		BIT	#MULTI, PCFLAG
(9)	012644	001424				BEQ	50032\$
2328	012646	004737	002110			JSR	PC, LINMON
2329	012652	012765	012734	000010		MOV	#T25A, RPC(R5)
2330	012660	032737	000400	001364		BIT	#LDONE, PCFLAG
(9)	012666	001406				BEQ	50033\$
2331	012670	042737	000400	001364		BIC	#LDONE, PCFLAG
2332	012676	052765	020000	000004		BIS	#TDONE, MFLAGS(R5)
2333	012704				50033\$:		
2334	012704	012765	011546	000010		MOV	#T21, RPC(R5)
2335	012712	000207			21000	RTS	PC
2336	012714	000512				BR	50034\$
(3)	012716						
2337	012716	113737	016134	020424		50032\$:	MOVB
2338	012724	012700	020421				DLOTH, SCODE+3
2339	012730	004737	007110			MOV	#SCODE, R0
2340	012734				21100	JSR	PC, TYPES
(4)	012734	012765	013034	000010		T25A:	
2341	012742	012701	013220			MOV	#T26, RPC(R5)
2342	012746	012702	000002			MOV	#T2BUF, R1
2343	012752	004737	006654			MOV	#STX, R2
2344					21300	JSR	PC, SECHO
2345					21350	; SET UP TO TEST HERE-IS KEY	SINGLE LINE ONLY
2346	012756	012700	013252				
2347	012762	004737	007110			MOV	#HI, R0
2348	012766	105037	013212			JSR	PC, TYPES
2349	012772	012703	007640			CLRB	T2CNT1
2350					21600	MOV	#4000, R3
2351	012776	004737	011216			; READ ANSWERBACK	
2352	013002	105737	013212			JSR	PC, GETANS
(9)	013006	001012				TSTB	T2CNT1
2353					21750	BNE	50035\$
2354					21800	; ERROR #5	NO ANSWERBACK FROM HERE-IS KEY
2355					21850	; *****	
2356	013010	052765	100000	000004			
2357	013016	112765	000005	000004		BIS	#MERR, MFLAGS(R5)
						MOVB	#5, MFLAGS(R5)

2358	013024	012765	013475	000006		MOV	#E17,POINT(R5)	
2359	013032	000207			22050	RTS	PC	
2360	013034					50035\$:		
2361					22150			
2362	013034				22200	T26:	; TEST CTL-E FUNCTION	
2363					22250			
2364					22300		; SET UP TO TEST CTL-E FUNCTION	
2365					22350			
2366	013034	012765	011726	000010		MOV	#T22,RPC(R5)	
2367	013042	012701	013220			MOV	#T2BUF,R1	
2368	013046	012700	013305			MOV	#CE,R0	
2369	013052	012703	007640			MOV	#4000,R3	
2370	013056	105037	013212			CLRB	T2CNT1	
2371	013062	004737	007110			JSR	PC,TYPES	
2372	013066	004737	011216			JSR	PC,GETANS	
2373	013072	105737	013212			TSTB	T2CNT1	
(9)	013076	001013				BNE	50036\$	
2374					22800	:	ERROR #6 NO ANSWERBACK FROM CTL-E KEY	
2375					22850	:	*****	
2376					22900	:		
2377	013100	052765	100000	000004		BIS	#MERR,MFLAGS(R5)	
2378	013106	112765	000006	000004		MOVB	#6,MFLAGS(R5)	
2379	013114	012765	013540	000006		MOV	#E18,POINT(R5)	
2380	013122	000207			23100	RTS	PC	
2381	013124	000406				BR	50037\$	
(3)	013126					50036\$:		
2382	013126	052765	020000	000004		BIS	#TDONE,MFLAGS(R5)	
2383	013134	012765	011726	000010		MOV	#T22,RPC(R5)	
2384	013142					50037\$:		
2385	013142					50034\$:		
2386	013142	000207			23400	RTS	PC	
2387					23450	:	*****	
2388					23500	:	THIS ROUTINE IS THE KEYBOARD INTERRUPT HANDLER	
2389					23550	:	FOR TESTS #1 AND #2	
2390					23600	:	*****	
2391					23650			
2392	013144	117721	002766		23700	T220:	MOVB	@DVCRXB.(R1)+ ; STORE CHAR IN POINTER
2393	013150	052737	004000	001364	23750		BIS	#DATAIN,PCFLAG ; SET DATA-IN FLAG
2394	013156	012777	000101	002744	23800		MOV	#101,@DLADR ; REENABLE THE RECVR
2395	013164	005037	007564		23850		CLR	DELAYT ; ABORT THE TIMEOUT
2396	013170	000002			23900		RTI	
2397					23950			
2398	013172					50000\$:		
(3)	013172					50001\$:		
(2)	013172	000207						
2399					24050	RTS	PC	
2400					24100	:	*****	
2401	013174	000	003		24150	.DSABL	LSB	
2402	013176	000000			24200	T02BLK:	.BYTE	0,3 ; ITERATION COUNTS
2403	013200	000000			24250		.WORD	0 ; CTLCNT
2404	013202	000000			24300		.WORD	0 ; PASS COUNT
2405	013204	000000			24350		.WORD	0 ; STATUS FLAGS
2406	013206	011546			24400		.WORD	T21 ; POINTER
2407	013210	000000			24450	T2SAV1:	.WORD	0 ; RETURN PC
2408	013212	000			24500	T2CNT1:	.BYTE	0
2409	013213	000			24550	T2CNT2:	.BYTE	0

2428					25450
2429					25500
2430					25550
2431					25600
2432					25650
2433					25700
2434					25750
2435					25800
2436					25850
2437	013700	012700	014376		25900
2438	013704	004737	007010		
2439	013710	012705	014342		
2440	013714	032737	000040	001364	26000
2441	013722	001424			26050
2442	013724	012701	000102		26100
2443	013730	113702	016167		26150
2444	013734	004737	006270		26200
2445	013740	012702	000000		26300
2446	013744	004737	006270		
2447	013750	005301			26400
2448	013752	001372			26450
2449	013754	012700	014427		26500
2450	013760	004737	007010		
2451	013764	052765	020000	000004	26600
2452					26650
2453	013772	000207			26700
2454					26750
2455					26800
2456					26850
2457	013774	012765	013774	000010	26900
2458	014002	012737	014610	014356	26950
2459	014010	012737	014362	014354	27000
2460	014016	012700	014443		27050
2461	014022	004737	007110		27100
2462	014026	012700	014646		27150
2463	014032	112037	014564		27200
2464	014036	112037	014565		27250
2465	014042	111037	014566		27300
2466	014046	012700	014557		27350
2467	014052	004737	007110		27400
2468	014056	013700	014356		27450
2469	014062	112037	014564		27500
2470	014066	112037	014565		27550
2471	014072	112037	014566		27600
2472	014076	010037	014356		27650
2473	014102	012702	014124		27700
2474	014106	004737	007124		27750
2475	014112	012702	035230		27800
2476	014116	004737	007024		27850
2477	014122	000406			27900
2478	014124	005037	007564		27950
2479	014130	052737	004000	001364	28000
2480	014136	000002			28050
2481	014140	012700	017730		28100
2482	014144	004737	007110		28150
2483	014150	042737	004000	001364	28200

```

*****
:TEST3 TOP OF FORM OPTION
:OPERATOR INTERVENTION REQUIRED IN SINGLE LINE MODE
*****
.ENABL LSB

TEST3:  MOV    #T3,RO           ;SU FOR TEST HEADER
        JSR    PC,MTYPE
        MOV    #T03BLK,R5      ;SET UP POINTER TO MODULE BLOCK
        BIT    #MULTI,PCFLAG   ;CHECK FOR SINGLE LINE MODE
        BEQ    3$,             ;SINGLE MODE - BRANCH
1$:     MOV    #66.,R1         ;FILL COUNT FOR 11" FORMS
        MOVW   FF,R2          ;MOVE TO TOP OF FJRM
        JSR    PC,MECHO
2$:     MOV    #ACK,R2
        JSR    PC,MECHO
        DEC    R1
        BNE    2$
        MOV    #DAS,RO
        JSR    PC,MTYPE
        BIS    #TDONE,MFLAGS(R5) ;SET ATTENTION & DONE FLAGS

        RTS    PC

*****
;THIS SECTION FOR SINGLE LINE MANUAL INTERVENTION
3$:     MOV    #3$,PC(R5)      ;SET RETURN PC TO HERE
        MOV    #HDR5,T3SAV1    ;GET LIST OF FORM LENGTHS
        MOV    #FILL3,T3SAV    ;GET FIL COUNT
        MOV    #HDR3,RO
        JSR    PC,TYPES
        MOV    #HDR5A,RO
        ;SET UP MESSAGE WITH
        MOVW   (RO)+,HDR4+5    ;INSTRUCTIONS FOR 3"
        MOVW   (RO)+,HDR4+6    ;FORMS.
        MOVW   (RO),HDR4+7
A3$:    MOV    #HDR4,RO       ;SEND SU MSG
        JSR    PC,TYPES
        MOV    T3SAV1,RO
        MOVW   (RO)+,HDR4+5    ;MODIFY INSTRUCTIONS FOR
        MOVW   (RO)+,HDR4+6    ;NEXT LOOP
        MOVW   (RO)+,HDR4+7
5$:     MOV    #6$,R2         ;SAVE THE LIST POINTER
        JSR    PC,READS       ;PASS 6$ AS VECTOR TO READ ROUTINE
        MOV    #15000.,R2     ;GO SET VECTORS
        JSR    PC,READ10      ;SET UP 15 SEC DELAY
        BR     9$
6$:     CLR    DELAYT         ;ABORT THE TIMEOUT
        BIS    #DATAIN,PCFLAG
        RTI
9$:     MOV    #L1,RO
        JSR    PC,TYPES
        BIC    #DATAIN,PCFLAG ;IN CASE LINE 0 ;SEND CR/LF

```

N06

2484	014156	042765	1000_0	000004	28250	BIC	#MERR,MFLAGS(R5);	IN CASE OF READ ERROR
2485	014164	117737	000164	014360	28300	MOVB	BT3SAV,T3SAV2	;GET FILL COUNT
2486	014172	113702	016167		28350	MOVB	FF,R2	
2487	014176	004737	006654		28400	JSR	PC,SECHO	;DO FORM FEED
2488	014202	012702	000006		28450	7\$:	MOV	#ACK,R2
2489	014206	004737	006654		28500	JSR	PC,SECHO	;SEND FILL CHARS
2490	014212	005337	014360		28550	DEC	T3SAV2	
2491	014216	001371			28600	BNE	7\$;COUNT NOT DONE - BRANCH
2492	014220	012700	014427		28650	MOV	#DAS,R0	;LINE OF DASHES
2493	014224	004737	007110		28700	JSR	PC,TYPES	
2494	014230	012702	000012		28750	MOV	#LF,R2	;SEND CR/LF
2495	014234	004737	006654		28800	JSR	PC,SECHO	
2496	014240	117737	000110	014360	28850	MOVB	BT3SAV,T3SAV2	;GET FILL COUNT
2497	014246	113702	016167		28900	MOVB	FF,R2	;DO FORM FEED
2498	014252	004737	006654		28950	JSR	PC,SECHO	
2499	014256	012702	000006		29000	8\$:	MOV	#ACK,R2
2500	014262	004737	006654		29050	JSR	PC,SECHO	;SEND ACK CHARS
2501	014266	005337	014360		29100	DEC	T3SAV2	
2502	014272	001371			29150	BNE	8\$;COUNT NOT DONE - BRANCH
2503	014274	012700	014427		29200	MOV	#DAS,R0	
2504	014300	004737	007110		29250	JSR	PC,TYPES	;LINE OF DASHES
2505	014304	012702	000012		29300	MOV	#LF,R2	
2506	014310	004737	006654		29350	JSR	PC,SECHO	
2507					29400			
2508	014314	005237	014354		29450	INC	T3SAV	;GET NEW FILL COUNT
2509	014320	023727	014356	014651	29500	CMP	T3SAV1,#HDR5E	;END OF INSTRUCTION LIST?
2510	014326	001247			29550	BNE	A3\$;NO - DO NEXT
2511	014330	052765	020000	000004	29600	BIS	#TDONE,MFLAGS(R5)	;SET ATTENTION & DONE FLAGS
2512					29650			
2513	014336	000207			29700	RTS	PC	
2514					29750			
2515	014340	000	003		29800			
2516	014342	000000			29850	T03BLK:	.BYTE 0,3	; ITERATION COUNT
2517	014344	000000			29900		.WORD 0	;CTLCNT
2518	014346	000000			29950		.WORD 0	;PASS COUNT
2519	014350	000000			30000		.WORD 0	;STATUS FLAGS
2520	014352	013724			30050		.WORD 0	;POINTER
2521					30100		.WORD 1\$;RETURN PC
2522	014354	000000			30150	T3SAV:	.WORD 0	; STORAGE
2523	014356	000000			30200	T3SAV1:	.WORD 0	
2524	014360	000000			30250	T3SAV2:	.WORD 0	
2525					30300			
2526					30350			
2527					30400	.NLIST BEX		
2528	014362	022	025	030	30450	FILL3:	.BYTE 18.,21.,24.,33.,36.,42.	;FILL COUNTS FOR TEST 3
2529	014370	060	063	102	30500		.BYTE 48.,51.,66.,72.,84.,18.	
2530	014376	005015	052012	051505	30550	T3:	.ASCIZ <15><12><12>/TEST 3 TOP OF FORMS/<15><12>	
2531	014427	075	036475	036475	30600	DAS:	.ASCIZ /=====/<15>	
2532	014443	120	042522	051523	30650	HDR3:	.ASCIZ /PRESS TOF RESET SWITCH/<15><12>	
2533	014473	101	052106	051105	30700		.ASCIZ /AFTER EACH SWITCH SETTING/<15><12>	
2534	014526	054524	042520	042040	30750		.ASCIZ /TYPE DELETE WHEN READY/<15><12>	
2535	014557	055	042523	020124	30800	HDR4:	.ASCIZ /-SET 3 INCH FORM FEED/	
2536		014610			30850		.EVEN	
2537	014610	027063	065		30900	HDR5:	.ASCIZ /3.5/	
2538	014613	040	032040		30950		.ASCIZ / 4/	
2539	014616	027065	065		31000		.ASCIZ /5.5/	

2540	014621	040	033040	31050	.ASCII	/	6/
2541	014624	020040	067	31100	.ASCII	/	7/
2542	014627	040	034040	31150	.ASCII	/	8/
2543	014632	027070	065	31200	.ASCII	/	8.5/
2544	014635	040	030461	31250	HDRSB: .ASCII	/	11/
2545	014640	030440	062	31300	.ASCII	/	12/
2546	014643	040	032061	31350	.ASCII	/	14/
2547	014646	020040	063	31400	HDRSA: .ASCII	/	3/
2548	014651			31450	HDRSE:		
2549		014652		31500	.EVEN		
2550				31550	.LIST BEX		

2552					31650
2553					31700
2554					31750
2555					31800
2556					31850
2557					31900
2558					31950
2559					32000
2560					32050
2561					32100
2562	014652	012705	015246		32150
2563	014656	012700	015320		32200
2564					32250
2565	014662	004737	007010		
2566	014666	012737	015270	015264	32350
2567	014674	012765	014702	000010	32400
2568	014702	012702	000033		32450
2569	014706	013737	016146	015262	32500
2570					32550
2571	014714	004737	006270		
2572	014720	012702	000062		32650
2573	014724	004737	006270		
2574	014730	117737	000330	015316	32750
2575	014736	005237	015264		32800
2576	014742	105077	000316		32850
2577	014746	013701	015316		32900
2578	014752	012700	017730		32950
2579					33000
2580	014756	004737	007010		
2581	014762	163737	015316	015262	33100
2582	014770	002434			33150
2583					33200
2584	014772	005301			33250
2585	014774	001405			33300
2586	014776	012702	000056		33350
2587	015002	004737	006270		
2588	015006	000771			33450
2589	015010	012702	000033		33500
2590					33550
2591	015014	004737	006270		
2592	015020	012702	000061		33650
2593	015024	004737	006270		
2594					33750
2595	015030	012702	000010		33800
2596	015034	004737	006270		
2597					33900
2598	015040	012702	000126		33950
2599	015044	004737	006270		
2600	015050	105277	000210		34050
2601	015054	013701	015316		34100
2602	015060	000740			34150
2603					34200
2604					34250
2605	015062	012737	000000	015266	34300
2606	015070	117737	000170	015316	34350
2607	015076	001440			34400

```

.DSABL LSB
*****
TEST4 HORIZONTAL TAB OPTION
      IF USING OTHER THAN 132 COL PAPER CHANGE LOC "WIDTH"
      TO APPROPRIATE VALUE. SEE WM COMMAND.
*****
TEST4:  MOV      #T04BLK,R5          ;SET UP POINTER TO MODULE BLOCK
        MOV      #T4,R0
        ; PRINT TEST HEADER
        JSR      PC,MTYPE
T41:    MOV      #TABL4,T4SAV2
        MOV      #T42,APC(R5)
T42:    MOV      #ESC,R2
        MOV      WIDTH,T4SAV1
        ; SEND ESC-2 TO RESET ALL TABS.
        JSR      PC,MECHO
        MOV      #'2,R2
        JSR      PC,MECHO
        MOV      #T4SAV2,TAB          ;GET TAB COUNT FROM TABL4
        INC      T4SAV2              ;INITIALIZE COUNT TO ZIP
        CLRB     #T4SAV2
        MOV      TAB,R1
        MOV      #L1,R0
        ; SEND CR/LF
        JSR      PC,MTYPE
3$:     SUB      TAB,T4SAV1          ;SU TAB COUNT PER LINE
        BLY     #5                  ;FINISHED THIS LINE - BRANCH
4$:     DEC      R1
        BEQ     #5
        MOV      #' ,R2
        JSR      PC,MECHO
        BR      #4$
5$:     MOV      #ESC,R2          ;SET TAB
        ; SEND ESC-1 TO SET A TAB
        JSR      PC,MECHO
        MOV      #'1,R2
        JSR      PC,MECHO
        ; SEND A BACKSPACE
        MOV      #10,R2
        JSR      PC,MECHO
        ; PRINT A V FOR REFERENCE
        MOV      #'V,R2
        JSR      PC,MECHO
        INCB     #T4SAV2            ;INCB TAB COUNT
        MOV      TAB,R1           ;GET TAB POS AGAIN
        BR      #3$               ;FORMAT NEXT SECTION
        ;LINE SHOULD LOOK LIKE THIS: ...V.....V.....V..ETC
6$:     MOV      #3,COUNT          ;DO 3 LINES OF TABS
7$:     MOV      #T4SAV2,TAB      ;GET TAB COUNT
        BEG     #1$               ;=0? - BRANCH OUT

```

2608	015100	005237	015264	34450	INC	T4SAV2		
2609	015104	012700	017730	34500	MOV	#L1,R0		
2610				34550	:	SEND A CR/LF		
2611	015110	004737	007010		JSR	PC, MTYPE		
2612	015114	012702	000011	34650	8\$:	MOV	#11,R2	
2613				34700	:	SEND A HORIZ-TAB		
2614	015120	004737	006270		JSR	PC, MECHO		
2615	015124	117737	000134	015260	34800	MOVB	#T4SAV2, T4SAV	: GET FILL COUNT TABS 2
2616	015132	012702	000006	34850	9\$:	MOV	#ACK, R2	
2617				34900	:	SEND FILL CHARACTERS		
2618	015136	004737	006270		JSR	PC, MECHO		
2619	015142	005337	015260	35000	DEC	T4SAV		
2620	015146	001371		35050	BNE	9\$		
2621	015150	012702	000130	35100	MOV	#X, R2		
2622				35150	:	PRINT A Y UNDER EACH V		
2623	015154	004737	006270		JSR	PC, MECHO		
2624	015160	005337	015316	35250	DEC	TAB		: DEC TAB COUNT
2625	015164	001353		35300	BNE	8\$: MORE TABS - BRANCH
2626	015166	005337	015264	35350	DEC	T4SAV2		: FIX POINTER
2627	015172	005337	015266	35400	10\$:	DEC	COUNT	: DO 3 LINES
2628	015176	001334		35450	BNE	7\$: NOT DONE - BRANCH
2629	015200	012700	017051	35500	11\$:	MOV	#L3, R0	
2630	015204	004737	007010	35550	JSR	PC, MTYPE		
2631	015210	062737	000002	015264	35600	ADD	#2, T4SAV2	: GET NEXT TABLE ENTR.
2632	015216	023727	015264	015315	35650	CMP	T4SAV2, #TAB-1	: END OF TABLE?
2633	015224	001226		35700	BNE	T42		: NO - DO NEXT SET
2634				35750				
2635	015226	052765	020000	000004	35950	BIS	#TDONE, MFLAGS(R5)	: SET ATTENTION AND DONE FLAGS
2636	015234	012765	014666	000010	36000	MOV	#T41, R0(R5)	
2637	015242	000207		36050	RTS	PC		
2638				36100				
2639	015244	000	004	36150				
2640	015246	000000		36200	T04BLK:	.BYTE	0, 4	: ITERATION COUNTS
2641	015250	000000		36250		.WORD	0	: CTLCNT
2642	015252	000000		36300		.WORD	0	: PASS COUNT
2643	015254	000000		36350		.WORD	0	: STATUS FLAGS
2644	015256	014702		36400		.WORD	0	: POINTER
2645				36450		.WORD	T12	: RETURN PC
2646	015260	000000		36500	T4SAV:	.WORD	0	: STORAGE
2647	015262	000000		36550	T4SAV1:	.WORD	0	
2648	015264	000000		36600	T4SAV2:	.WORD	0	
2649				36650				
2650				36700				
2651	015266	000002		36750	COUNT:	.WORD	2	
2652	015270	004	000	002	TABL4:	.BYTE	4, 0, 2	: TAB, TAB COUNT, FILL COUNT
2653	015273	010	000	004		.BYTE	8, 0, 4	
2654	015276	011	000	005		.BYTE	9, 0, 5	: TABLE FOR TEST 4
2655	015301	020	000	010		.BYTE	16, 0, 8	
2656	015304	022	000	012		.BYTE	18, 0, 10	
2657	015307	040	000	021		.BYTE	32, 0, 17	
2658	015312	100	000	041		.BYTE	64, 0, 33, 0	
2659	015315	000						
2659	015316	000000		37150	TAB:	.WORD	0	
2660	015320	005015	052012	051505	T4:	.ASCII	'15' '12' '12'	TEST 4 HORIZONTAL TAB 15 12
	015326	020124	020064	047510				
	015334	044522	047532	052116				

E07

CELAFAD LA36 TERM TST MACY11 30A1052 03-JAN-77 00:01 PAGE 3-17
CELAFAD.P11 03-JAN-78 11:20 LA36 OPTION TESTS

FEV 0082

015342 046101 052040 041101
015350 005015 000

266:

37250 .EVEN

2663					37350
2664					37400
2665					37450
2666					37500
2667					37550
2668					37600
2669					37650
2670					37700
2671					37750
2672	015354	012700	016040		37800
2673	015360	012705	016014		37850
2674	015364	004737	007010		
2675	015370	032737	000040	001364	37950
2676	015376	001046			38000
2677	015400	012700	014635		38050
2678	015404	112037	014564		38100
2679	015410	112037	014565		38150
2680	015414	112037	014566		38200
2681					38250
2682	015420	012700	014443		38300
2683	015424	004737	007110		
2684	015430	012700	014557		38400
2685	015434	004737	007110		
2686	015440	012702	015500		38500
2687	015444	004737	007124		38550
2688	015450	012702	035230		38600
2689	015454	004737	007024		38650
2690	015460	032737	004000	001364	38700
2691	015466	001770			38750
2692	015470	042737	004000	001364	38800
2693	015476	000406			38850
2694					38900
2695					38950
2696	015500	005037	007564		39000
2697	015504	052737	004000	001364	39050
2698	015512	000002			39100
2699					39150
2700					39200
2701					39250
2702	015514	012737	000002	015266	39300
2703	015522	012765	015514	000010	39350
2704	015530	012737	000001	016032	39400
2705	015536	005037	016036		39450
2706	015542	012737	000014	016034	39500
2707	015550	012702	000033		39550
2708					39600
2709	015554	004737	006270		
2710	015560	012702	000064		39700
2711	015564	004737	006270		
2712	015570	013701	016032		39800
2713	015574	012702	000012		39850
2714					39900
2715	015600	004737	006270		
2716	015604	005301			40000
2717	015606	001372			40050
2718	015610	012702	000033		40100

```

*****
TESTS VERTICAL TAB OPTION
SINGLE LINE TEST REQUIRES OPERATOR INTERVENTION
*****
.ENABL LSB

TESTS:  MOV    #15,R0      ;SU TEST HEADER
        MOV    #TOSBLK,R5 ;SET UP POINTER TO MODULE BLOCK
        JSR    PC,MTYPE
        BIT    #MULTI,PCFLAG ;MULTI LINE MODE?
        BNE    4$          ;YES - BRANCH OVFR INTERVENTION
        MOV    #HDR5B,R0   ;SET UP INSTRUCTIONS
        MOVB   (R0)+,HDR4+5
        MOVB   (R0)+,HDR4+6
        MOVB   (R0)+,HDR4+7

; TYPE INSTRUCTIONS
1$:  MOV    #HDR3,R0
     JSR    PC,TYPES
     MOV    #HDR4,R0
     JSR    PC,TYPES
     MOV    #3$,R2        ;SU FOR INTERRUPT TO 3$
     JSR    PC,READS      ;INITIALIZE VECTOR AREA
2$:  MOV    #15000,R2     ;ALLOW 15 SEC.
     JSR    PC,READIO
     BIT    #DATAIN,PCFLAG
     BEQ    2$
     BIC    #DATAIN,PCFLAG
     BR     4$

;***** THIS SECTION HANDLES RECVR INTERRUPTS*****
3$:  CLR    DELAYT        ;ABORT THE TIMEOUT
     BIS    #DATAIN,PCFLAG ; FLAG RECIEVED CHAR.
10$: RTI

;*****
4$:  MOV    #2,COUNT
     MOV    #4$,RPC(R5)   ;SET RETURN TO 4$
     MOV    #1,LINES
     CLR    TABS
     MOV    #12.,MAX
     MOV    #ESC,R2      ;RESET ALL TABS
; FSC-4 RESETS THE TABS.
     JSR    PC,MECHO
     MOV    #4,R2
     JSR    PC,MECHO
5$:  MOV    LINES,R1     ;GET LINE COUNT
6$:  MOV    #12.,R2
; SEND LINE FEED.
     JSR    PC,MECHO
     DEC    R1
     BNE    6$
     MOV    #ESC,R2     ;SET TAB

```

2719	015614	004737	006270			JSR	PC,MECHO		
2720	015620	012702	000063		40200	MOV	#3,R2		
2721					40250	: ESC-3	SETS A	TAB LOCATION.	
2722	015624	004737	006270			JSR	PC,MECHO		
2723	015630	012700	014427		40350	MOV	#DAS,RO		
2724					40400				
2725	015634	004737	007010			JSR	PC,MTYPE		
2726	015640	005237	016032		40500	INC	LINES		
2727	015644	023737	016032	016034	40550	CMP	LINES,MAX	: 11 TABS SET	
2728	015652	001346			40600	BNE	SS	: NO - BRANCH	
2729	015654	012737	000001	016032	40650	7S:	MOV	#1,LINES	: RESET LINE COUNT
2730	015662	012737	000001	016030	40700	MOV	#1,TSSAV1	: FILL COUNT	
2731	015670	012702	000013		40750	8S:	MOV	#13,R2	
2732					40800	: SEND A VERT-TAB COMMAND.			
2733	015674	004737	006270			JSR	PC,MECHO		
2734	015700	012702	000006		40900	9S:	MOV	#ACK,R2	
2735					40950	: SEND A FILL CHARACTER.			
2736	015704	004737	006270			JSR	PC,MECHO		
2737	015710	005337	016030		41050	DEC	TSSAV1		
2738	015714	001371			41100	BNE	9S		
2739					41150				
2740					41200	: CONVERT NO. OF LINES FOR OUTPUT MSG.			
2741					41250				
2742	015716	013746	016032		41300	MOV	LINES, -(SP)		
2743	015722	012746	016071		41400	MOV	#T52, -(SP)		
2744	015726	004737	010006		41450	JSR	PC,BIN20A		
2745	015732	012700	016071		41500	MOV	#T52,RO		
2746	015736	004737	007010		41550	JSR	PC,MTYPE		
2747	015742	012700	014427		41600	MOV	#DAS,RO	: SU LINE OF DASHES	
2748	015746	004737	007010			JSR	PC,MTYPE		
2749	015752	005237	016032		41700	INC	LINES	: NEW LINE COUNT	
2750	015756	013737	016032	016030	41750	MOV	LINES,TSSAV1	: FILL COUNT = #LINES	
2751	015764	023737	016032	016034	41800	CMP	LINES,MAX	: 11 TABS DONE?	
2752	015772	001336			41850	BNE	8S	: NO - CONTINUE	
2753	015774	005337	015266		41900	DEC	COUNT	: DO 2 PAGES TOTAL	
2754	016000	001325			41950	BNE	7S	: RE-DO PAGE	
2755	016002	052765	020000	000004	42000	BIS	#TDONE,MFLAGS(R5)	: SET ATTENTION & DONE FLAGS	
2756					42050				
2757	016010	000207			42100	RTS	PC		
2758					42150				
2759					42200				
2760	016012	000	002		42250	: ITERATION COUNTS			
2761	016014	000000			42300	T05BLK: .BYTE	0,2	: CTLCNT	
2762	016016	000000			42350	.WORD	C	: PASS COUNT	
2763	016020	000000			42400	.WORD	0	: STATUS FLAGS	
2764	016022	000000			42450	.WORD	0	: POINTER	
2765	016024	015420			42500	.WORD	T51	: RETURN PC	
2766					42550				
2767	016026	000000			42600	TSSAV: .WORD	0		
2768	016030	000000			42650	TSSAV1: .WORD	0		
2769					42700				
2770	016032	000000			42750	LINES: .WORD	0		
2771	016034	000000			42800	MAX: .WORD	0		
2772	016036	000000			42850	TABS: .WORD	0		
2773	016040	005015	052012	051505	42900	T5: .ASCIZ	'15'12'12' TEST 5 VERTICAL TAB <15'12		
	016046	020124	020065	042526					

H07

	016054	052122	041511	046101				
	016062	052040	041101	005015				
	016070	000						
2774	016071	060	030060	030060	42950	T52:	.ASCIZ	'00000'
	016076	000						
2775		016100			43000		.EVEN	
2776					43050		.DSMBL	LSB

2778				43150	
2779				43200	
2780				43250	
2781				43300	
2782				43350	
2783	016100	000000	000000	000000	43400
	016106	000000	000000	000000	
2784	016114	000004			43450
2785	016124	177570			43500
2786					43550
2787					43600
2788					43650
2789	016126	000000			43700
2790	016130	000000			43750
2791	016132	000000			43800
2792	016134	000000			43850
2793	016136	000000			43900
2794	016140	000000			43950
2795	016142	000000			44000
2796	016144	000000			44050
2797					44100
2798					44150
2799	016146	000204			44200
2800	016150	000000			44250
2801	016152	000000			44300
2802	016154	000000			44350
2803	016156	000000			44400
2804	016160	000000			44450
2805	016162	000000			44500
2806	016164	016114			44550
2807	016166	177			44600
2808	016167	014			44650
2809					44750
2810					44800

```

.SBTTL STORAGE & CONSTANTS
*****
PROGRAM STORAGE, CONSTANTS, AND VARIABLES
*****
TEMP: .EVEN
      .WORD 0.0,0.0,0.0 ;TEMPORARY WORK AREA
INBUF: .BLKW 4 ;INPUT BUFFER
SWR: 177570 ;SWITCH REGISTER POINTER
          ;MAY BE CHANGED TO 176
***** I/O DRIVER WORK AREA *****
DLFLAG: .WORD 0 ;LINE FLAG WORD
DLADR: .WORD 0 ;LINE ADDR WORD
DLVEC: .WORD 0 ;LINE VECTOR WORD
DLOTH: .WORD 0 ;LINE "OTHER WORD
DVCRXB: .WORD 0 ;RECIEVER DATA BUFFER
DVCTXS: .WORD 0 ;TRANSMI STATUS REGISTER
DVCTXB: .WORD 0 ;TRANSMIT DATA BUFFER
TXVEC: .WORD 0 ;TRANSMIT INTERRUPT VECTOR
***** GENERAL USE *****
WIDTH: .WORD 132.
SAVE: .WORD 0
NEXT: .WORD 0 ;NEXT TEST NO.
INTEST: .WORD 0 ;CURRENT TEST
TESTAD: .WORD 0 ;CERRENT TEST PC.
ONLIN: .WORD 0 ;CURRENT LINE UNDER TEST
NXTLIN: .WORD 0 ;NEXT LINE TO TEST
PTR: INBUF ;INPUT BUFFER POINTER
DEL: .BYTE 177
FF: .BYTE 14
      .EVEN

```

Address	Value	Label	Description
2812	44900	LINE INTERFACE TABLE	
2813	44950		
2814	45000		
2815	45050		
2816	016170 000000 177560 000000 45100	LIN00: .WORD	0,177560,0,0 ;CONSOLE INTERFACE
2817	016176 000000 175610 000000 45150	LIN01: .WORD	0,175610,0,400 ;DL11-C,D,E LINES
2818	016200 000000 175620 000000 45200	LIN02: .WORD	0,175620,0,1000
2819	016206 000400 175630 000000 45250	LIN03: .WORD	0,175630,0,1400
2820	016210 000000 175640 000000 45300	LIN04: .WORD	0,175640,0,2000
2821	016216 001000 176500 000000 45350	LIN05: .WORD	0,176500,0,2400
2822	016220 000000 176510 000000 45400	LIN06: .WORD	0,176510,0,3000
2823	016226 001400 176520 000000 45450	LIN07: .WORD	0,176520,0,3400 ;FIRST WORD : FLAGS
2824	016230 000000 176530 000000 45500	LIN10: .WORD	0,176530,0,4000 ;BIT 15 = DVC PRESENT
2825	016236 002000 176550 000000 45550	LIN11: .WORD	0,176550,0,4400 ;BIT 7 = DVC SELECTED
2826	016240 000000 176560 000000 45600	LIN12: .WORD	0,176560,0,5000 ;BIT 4 = ABORT FLAG
2827	016246 002400 176570 000000 45650	LIN13: .WORD	0,176570,0,5400 ;BIT 3 THRU
2828	016250 003000 175700 000000 45700	LIN14: .WORD	0,175700,0,6000 ;BIT 0 = ERROR COUNT
2829	016256 003400 175710 000000 45750	LIN15: .WORD	0,175710,0,6400
2830	016266 003400 175720 000000 45800	LIN16: .WORD	0,175720,0,7000
2831	016270 000000 175730 000000 45850	LIN17: .WORD	0,175730,0,7400 ;THIRD WORD WILL CONTAIN
2832	016276 004000 175740 000000 45900	LIN20: .WORD	0,175740,0,10000 ;THE DEVICES INTERRUPT
2833	016300 000000 176540 000000 45950	LIN21: .WORD	0,176540,0,10400 ;VECTOR (SUPPLIED BY PROGRAM)
2834	016306 004400 176550 000000 46000	LIN22: .WORD	0,176550,0,11000
2835	016310 000000 176560 000000 46050	LIN23: .WORD	0,176560,0,11400
2836	016316 005000 176570 000000 46100	LIN24: .WORD	0,176570,0,12000 ;WORD FOUR :
2837	016320 005400 176600 000000 46200	LIN25: .WORD	0,176600,0,12400 ;BITS 7 THRU 0
2838	016326 005400 176610 000000 46250	LIN26: .WORD	0,176610,0,13000 ;WILL BE SET TO
2839	016330 006000 175750 000000 46300	LIN27: .WORD	0,175750,0,13400 ;UNIQUE SELECT CODE
2840	016336 006000 175760 000000 46350	LIN30: .WORD	0,175760,0,14000
2841	016340 006400 175770 000000 46400	LIN31: .WORD	0,175770,0,14400 ;BITS 13 THRU 9
2842	016346 006400		

Line No.	Address	Value	Label	Format	Start	End	Binary Line No.
2843	016506	014400	176000	000000	46450	LIN32: .WORD	0,176000,0,15000
	016510	000000					
	016516	015000					
2844	016520	000000	176010	000000	46500	LIN33: .WORD	0,176010,0,15400
	016526	015400					
2845	016530	000000	176020	000000	46550	LIN34: .WORD	0,176020,0,16000
	016536	016000					
2846	016540	000000	176030	000000	46600	LIN35: .WORD	0,176030,0,16400
	016546	016400					
2847	016550	000000	176040	000000	46650	LIN36: .WORD	0,176040,0,17000
	016556	017000					
2848	016560	000000	176620	000000	46700	LIN37: .WORD	0,176620,0,17400
	016566	017400					
2849	016570	000000	176630	000000	46750	LIN40: .WORD	0,176630,0,20000
	016576	020000					
2850	016600	000000	176640	000000	46800	LIN41: .WORD	0,176640,0,20400
	016606	020400					
2851	016610	000000	176650	000000	46850	LIN42: .WORD	0,176650,0,21000
	016616	021000					
2852	016620	000000	176660	000000	46900	LIN43: .WORD	0,176660,0,21400
	016626	021400					
2853	016630	000000	176670	000000	46950	LIN44: .WORD	0,176670,0,22000
	016636	022000					
2854	016640	000000	176050	000000	47000	LIN45: .WORD	0,176050,0,22400
	016646	022400					
2855	016650	000000	176060	000000	47050	LIN46: .WORD	0,176060,0,23000
	016656	023000					
2856	016660	000000	176070	000000	47100	LIN47: .WORD	0,176070,0,23400
	016666	023400					
2857	016670	000000	176100	000000	47150	LIN50: .WORD	0,176100,0,24000
	016676	024000					
2858	016700	000000	176110	000000	47200	LIN51: .WORD	0,176110,0,24400
	016706	024400					
2859	016710	000000	176120	000000	47250	LIN52: .WORD	0,176120,0,25000
	016716	025000					
2860	016720	000000	176130	000000	47300	LIN53: .WORD	0,176130,0,25400
	016726	025400					
2861	016730	000000	176140	000000	47350	LIN54: .WORD	0,176140,0,26000
	016736	026000					
2862	016740	000000	176150	000000	47400	LIN55: .WORD	0,176150,0,26400
	016746	026400					
2863	016750	000000	176160	000000	47450	LIN56: .WORD	0,176160,0,27000
	016756	027000					
2864	016760	000000	176170	000000	47500	LIN57: .WORD	0,176170,0,27400
	016766	027400					
2865	016770	177777			47550	TABEND: .WORD	-1
2866					47600		

```

2868 47700
2869 47750
2870 47800
2871 47850
2872 016772 005015 055103 040514 47900
2873 017030 005015 042522 052123 47950
2874 017051 015 005012 000012 48000
2875 017056 047503 046515 047101 48050
2876 017103 123 020040 020040 48100
2877 017135 115 020040 020040 48150
2878 017166 020121 020040 020040 48200
2879 017216 047122 020040 020040 48250
2880 017245 104 020116 020040 48300
2881 017275 101 020116 020040 48350
2882 017324 020124 020040 020040 48400
2883 017356 047127 020040 020040 48450
2884 017413 114 020040 020040 48500
2885 017443 110 020040 020040 48550
2886 017473 103 020040 020040 48600
2887 017542 020116 020040 020040 48650
2888 017574 020120 020040 020040 48700
2889 017632 051505 020103 020040 48750
2890 017676 005015 046012 047111 48800
2891 017730 005015 000 000 48850
2892 017733 060 020060 020040 48900
2893 017743 060 030060 020060 48950
2894 017752 030060 020060 020040 49000
2895 017761 052 051105 047522 49050
2896 020017 040 020040 037477 49100
2897 020032 026455 046055 047111 49150
2898 020055 116 020117 047111 49200
2899 020105 104 047522 050120 49250
2900 020117 052 005015 000 49300
2901 020123 015 051012 040505 49350
2902 020136 005015 040520 051523 49400
2903 020166 005015 025052 020052 49450
2904 49500
2905 020215 015 050012 043103 49550
2906 020242 054105 042503 051523 49600
2907 020270 030060 042040 047522 49650
2908 020306 047516 046040 047111 49700
2909 020345 114 047111 020105 49750
2910 020370 047503 051516 046117 49800
2911 49850
2912 020411 004 003401 000002 49900
2913 020416 000404 000 000 49950
2914 020421 004 001401 000000 50000
2915 020426 000404 001003 000 50050
2916 020434 020434 50100
2917 020574 000060 50150
2918 020670 000036 50200
2919 000000 50250
2920 001102 50300

```

```

; * * * * *
; SYSTEM MESSAGES
; .NLIST BEX
PROGID: .ASCII <15><12>/CZLAFAD LA36 OPTIONS TESTS/
L3: .ASCII <15><12>/RESTART AT 1372/
HEADR1: .ASCIZ <15><12><12><12>
COMSUM: .ASCII /S SINGLE LINE MODE/<15><12><1>
.ASCII /M MULTI-LINE MODE/<15><12><1>
.ASCII /Q SEQUENCE TESTS/<15><12><1>
.ASCII /RN RUN TEST "N" /<15><12><1>
.ASCII /DN DROP LINE "N" /<15><12><1>
.ASCII /AN ADD LINE "N" /<15><12><1>
.ASCII /T TYPE LINE TABLE /<15><12><1>
.ASCII /WN CHANGE "WIDTH" TO N/<15><12><1>
.ASCII /L LOOP ON ERROR /<15><12><1>
.ASCII /H HALT ON ERROR /<15><12><1>
.ASCII /C CLEAR ; RESETS H & L COMMANDS/<15><12><1>
.ASCII /N INHIBIT REPORTS /<15><12><1>
.ASCII /P PRINT ERROR REPORTS /<15><12><1>
.ASCII /ESC TO EXECUTE COMMAND STRING/<15><12>/12
HEADR2: .ASCII <15><12><12>/LINE# ADDR VECTOR SEL/
L1: .ASCIZ <15><12>
LIN: .ASCII /00 17/
DLAD: .ASCII /0000 /
DLV: .ASCIZ /000 /
ER0: .ASCIZ /*ERROR 000 TEST 00 LINE 00/<15><12><7>
ER1: .ASCIZ / "???" /<15><12><7>
ER2: .ASCIZ /---LINE INVALID/<15><12><7>
ER7: .ASCIZ /NO INTERRUPT ON TXMIT/<15><12>
DR: .ASCIZ /DROPPED/<15><12>
S1: .ASCIZ /*/<15><12>
RDY: .ASCIZ <15><12>/READY /<15><12>
EOPM: .ASCIZ <15><12>/PASS 0000 TEST 00/<15><12>
EOTM: .ASCIZ <15><12>/*** END OF TEST 00/<15><12>
SW: .ASCIZ <15><12>/PCFLAG : 000000 /<15><12>
DRO: .ASCII /EXCESSIVE ERRORS..LINE/
DR1: .ASCIZ /00 DROPPED/<15><12><7>
E19: .ASCIZ /NO LINES AVAILABLE FOR TEST/<15><12><7>
E20: .ASCIZ /LINE RE-SELECTED/<15><12>
CTLM: .ASCIZ /CONSOLE CONTROL?/
ALLON: .ASCIZ <4><1><7><2> ;SELECT ALL ESCAPE SEQUENCE
ALLOFF: .ASCIZ <4><1> ;DESELECT ALL SEQUENCE
SCODE: .ASCIZ <4><1><3><000> ;SELECT UNIQUE SEQUENCE
NSELC: .ASCIZ <4><1><3><2> ;BAD SELECT SEQUENCE
.EVEN
STACK2: .BLKW 48.
STACK3: .BLKW 30.
ENDS: .WORD 0
.END START

```

ABO = 000020
 ACK = 000006
 ADDC = 000004
 ALLOFF = 020416
 ALLON = 020411
 ANSHDR = 011472
 ATTN = 000200
 A2BIN = 007676
 A2SAV = 010004
 A3\$ = 014046
 BIN2DA = 010006
 BIT0 = 000001
 BIT00 = 000001
 BIT01 = 000002
 BIT02 = 000004
 BIT03 = 000010
 BIT04 = 000020
 BIT05 = 000040
 BIT06 = 000100
 BIT07 = 000200
 BIT08 = 000400
 BIT09 = 001000
 BIT1 = 000002
 BIT10 = 002000
 BIT11 = 004000
 BIT12 = 010000
 BIT13 = 020000
 BIT14 = 040000
 BIT15 = 100000
 BIT2 = 000004
 BIT3 = 000010
 BIT4 = 000020
 BIT5 = 000040
 BIT6 = 000100
 BIT7 = 000200
 BIT8 = 000400
 BIT9 = 001000
 BPTVEC = 000014
 BUILD = 005640
 CATCH = 006126
 CE = 013305
 CFLAGS = 002032
 CHARS = 010310
 CHKW = 005006
 CMDERR = 005062
 CNTDA = 010160
 COMSUM = 017103
 CON = 010662
 CONSON = 003332
 COUNT = 015266
 CR = 000015
 CRLF = 000200
 CSI = 004000

CTLBLK = 001364
 CTLC = 000003
 CTLCNT = 000000
 CTLG = 000007
 CTLGX = 003064
 CTLH = 000010
 CTLK = 000013
 CTLL = 000014
 CTLM = 020370
 CTLN = 000016
 CTLP = 000020
 DAS = 014427
 DATA = 004776
 DATAIN = 004000
 DATA2 = 005000
 DDISP = 177570
 DECODE = 004230
 DECSAV = 004774
 DECTBL = 004656
 DEL = 016166
 DELAYM = 007530
 DELAYR = 104006
 DELAYT = 007564
 DIGITS = 010162
 DLADR = 017743
 DLADR = 016130
 DLFLAG = 016126
 DLOTH = 016134
 DLP = 100000
 DLV = 017752
 DLVEC = 016132
 DR = 020105
 DROPC = 000010
 DRO = 020242
 DR1 = 020270
 DSWR = 177570
 DTEND = 004774
 DVCRXB = 016136
 DVCTXB = 016142
 DVCTXS = 016140
 ECHO = 005044
 EMTABL = 006230
 EMTBOS = 006162
 EMTVEC = 000030
 ENDS = 020670
 ENQ = 000005
 EOL = 004000
 EOP = 020000
 EOPM = 020136
 EOT = 040000
 EOTM = 020166
 ERROR = 005124
 ERRSAV = 005434

ERRVEC = 000004
 ERO = 017761
 ER1 = 020C17
 ER2 = 020032
 ER7 = 020055
 FSC = 000033
 ETX = 000003
 ETYPE = 006240
 E10 = 011137
 E12 = 011045
 E14 = 013356
 E15 = 013404
 E16 = 013440
 E17 = 013475
 E18 = 013540
 E19 = 020306
 E20 = 020345
 E21 = 013575
 E22 = 013632
 E9 = 011005
 FF = 016167
 FILL3 = 014362
 FLAGDA = 010161
 FLAG1 = 000001
 FLAG2 = 000002
 GETANS = 011216
 GETSRC = 003350
 GETSWS = 003130
 GNL = 002556
 GN1 = 002600
 GN2 = 002604
 GN3 = 002612
 GN4 = 002644
 GN5 = 002650
 GP = 011100
 GSEL = 010734
 GVL = 002332
 G1A = 002340
 G1B = 002344
 G1C = 002356
 G1D = 002370
 HALTC = 000200
 HALTOE = 100000
 HDR3 = 014443
 HDR4 = 014557
 HDR5 = 014610
 HDR5A = 014646
 HDR5B = 014635
 HDR5E = 014651
 HEADR1 = 017056
 HEADR2 = 017676
 HI = 013252
 HT = 000011

ICNT = 177776
 INBUF = 016114
 INHR = 000040
 INHRPT = 020000
 INTEST = 016154
 INTRAP = 007162
 IOTVEC = 000020
 ISP = 005726
 ISP2 = 022626
 ITRAP = 104004
 LOONE = 000400
 LF = 000012
 LIN = 017733
 LINENO = 001370
 LINES = 016032
 LINESE = 002204
 LINMON = 002110
 LIN00 = 016170
 LIN01 = 016200
 LIN02 = 016210
 LIN03 = 016220
 LIN04 = 016230
 LIN05 = 016240
 LIN06 = 016250
 LIN07 = 016260
 LIN10 = 016270
 LIN11 = 016300
 LIN12 = 016310
 LIN13 = 016320
 LIN14 = 016330
 LIN15 = 016340
 LIN16 = 016350
 LIN17 = 016360
 LIN20 = 016370
 LIN21 = 016400
 LIN22 = 016410
 LIN23 = 016420
 LIN24 = 016430
 LIN25 = 016440
 LIN26 = 016450
 LIN27 = 016460
 LIN30 = 016470
 LIN31 = 016500
 LIN32 = 016510
 LIN33 = 016520
 LIN34 = 016530
 LIN35 = 016540
 LIN36 = 016550
 LIN37 = 016560
 LIN40 = 016570
 LIN41 = 016600
 LIN42 = 016610
 LIN43 = 016620

LIN44 = 016630
 LIN45 = 016640
 LIN46 = 016650
 LIN47 = 016660
 LIN50 = 016670
 LIN51 = 016700
 LIN52 = 016710
 LIN53 = 016720
 LIN54 = 016730
 LIN55 = 016740
 LIN56 = 016750
 LIN57 = 016760
 LOOPC = 000100
 LOOPOE = 040000
 LOOP1 = 001454
 LOOP2 = 001466
 L1 = 017730
 L3 = 017051
 MACHER = 000004
 MAJOR = 003000
 MAX = 016034
 MACHO = 006270
 MERR = 100000
 MERRN = 100377
 MFLAGS = 000004
 MSAVE = 006646
 MTW = 002432
 MTW1 = 002452
 MTYPE = 007010
 MULTI = 000040
 NEWMOD = 001000
 NEWTST = 002000
 NEXT = 016152
 NOOP = 000240
 NOP = 000240
 NREQ = 000340
 NSELC = 020426
 NXTLIN = 016162
 OCTALC = 011520
 ONLIN = 016160
 O2ASC = 007566
 PASCNT = 000002
 PCFLAG = 001364
 PIRQ = 177772
 PIRQVE = 000240
 POINT = 000006
 PRI = 010424
 PRINT = 000020
 PRINTT = 010000
 PRI0 = 000000
 PRI4 = 000200
 PRI7 = 000340
 PROGID = 016772

PRTLTB	007212	SWR	016124	TKV	= 000060	T4SAV1	015262	\$NSKO	= 000300
PRTTBL	= 104002	SWRTST	003006	TKVEC	= 000060	T4SAV2	015264	\$NSK1	= 000110
PRO	= 000000	SWTEST	000172	TF8	= 177566	T41	014666	\$NSK10	= 000110
PR1	= 000040	SW0	= 000001	TPS	= 177564	T42	014702	\$NSK11	= 000110
PR2	= 000100	SW00	= 000001	TPVEC	= 000064	T5	016040	\$NSK12	= 000110
PR3	= 000140	SW01	= 000002	TRAPVE	= 000034	T5SAV	016026	\$NSK2	= 000110
PR4	= 000200	SW02	= 000004	TRTVEC	= 000014	T5SAV1	016030	\$NSK3	= 000110
PR5	= 000240	SW03	= 000010	TSCCNT	002034	T51	015420	\$NSK4	= 000110
PR6	= 000300	SW04	= 000020	TSCPTR	002034	T52	016071	\$NSK5	= 000110
PR7	= 000340	SW05	= 000040	TSTBL	002040	UPDATE	002660	\$NSK6	= 000110
PS	= 177776	SW06	= 000100	TSTCTL	001446	WIDTH	016146	\$NSK7	= 000110
PSW	= 177776	SW07	= 000200	TSTMOM	= 050000	\$BGNLE	= 177777	\$SAVLE	= 177777
PTR	016164	SW08	= 000400	TXTRAP	007416	\$ERFLG	= 000400	\$SSKO	= 050023
PWRVEC	= 000024	SW09	= 001000	TXVEC	016144	\$FSAND	= 000310	\$SVPC	= 000204
RDSAV	003776	SW1	= 000002	TYPANS	011306	\$FSBAD	= 000401	\$SWR	= 160000
RDY	020123	SW10	= 002000	TYPE	= 104000	\$FSBLA	= 000170	\$TAGLE	= 177777
READIO	007024	SW11	= 004000	TYPES	007110	\$FSCAS	= 000150	\$TAGNU	= 050040
READKB	003466	SW12	= 010000	TO	010366	\$FSDEC	= 000220	\$TEMP	= 000300
READS	007124	SW13	= 020000	TOOBLK	010352	\$FSDO	= 000340	\$TN	= 000001
READY	= 000200	SW14	= 040000	TO1	010200	\$FSFAL	= 000405	\$TSKO	= 050034
REPORT	005436	SW15	= 100000	TO1BLK	010722	\$FSGOO	= 000400	\$TSK1	= 050037
RESTR	001372	SW2	= 000004	TO2BLK	013176	\$FSIF	= 000110	\$TSK10	= 050023
RESVEC	= 000010	SW3	= 000010	TO3BLK	014342	\$FSINC	= 000210	\$TSK11	= 050024
RPC	= 000010	SW4	= 000020	TO4BLK	015246	\$FSL00	= 000200	\$TSK2	= 050024
RUB	010364	SW5	= 000040	TO5BLK	016014	\$FSNAM	= 000160	\$TSK3	= 050023
R6	= %000006	SW6	= 000100	T1	010742	\$FSNO	= 000403	\$TSK4	= 050025
R7	= %000007	SW7	= 000200	T1TEMP	010736	\$FSOR	= 000320	\$TSK5	= 050022
SAVE	016150	SW8	= 000400	T11	010460	\$FSRTN	= 000300	\$TSK6	= 050017
SCODE	020421	SW9	= 001000	T12	010554	\$FSSEL	= 000140	\$TSK7	= 050021
SEC	010434	S1	020117	T13	010566	\$FSTHE	= 000330	\$SARGC	= 000000
SECHO	006654	TAB	015316	T16	010634	\$FSTRU	= 000404	\$SBYTE	= 000402
SEL	= 000200	TABDA	010146	T2	013326	\$FSUNT	= 000130	\$SCASE	= 000000
SELERR	005100	TABEND	016770	T2BUF	013220	\$FSWHI	= 000120	\$SDST	= 000000
SEQ	= 000100	TABL1	011176	T2CNT1	013212	\$FSYES	= 000402	\$SELOC	= 000402
SETIO	005600	TABL4	015270	T2CNT2	013213	\$HD	= 000003	\$SERFL	= 000000
SI	= 000017	TABS	016036	T2SAV1	013210	\$IFLEV	= 177777	\$SFLAG	= 000001
SO	= 000016	TBITVE	= 000014	T2TEMP	013214	\$ISKO	= 000001	\$SFRM	= 000000
SOH	= 000001	TDONE	= 020000	T21	011546	\$ISK1	= 000001	\$SLOC	= 013076
SSWR	000176	TEMP	016100	T22	011726	\$ISK10	= 000001	\$SLOCN	= 000000
STACK	= 001100	TEMPF	005002	T220	013144	\$ISK11	= 000001	\$SREG	= 177777
STACK2	020434	TEMPT	005004	T23	012140	\$ISK2	= 000001	\$SRETU	= 000000
STACK3	020574	TESTAD	016156	T23A	012206	\$ISK3	= 000001	\$SRTN1	= 050007
START	001102	TESTNO	001366	T24	012530	\$ISK4	= 000001	\$SRTN2	= 050001
START2	001172	TESTO	010164	T25	012636	\$ISK5	= 000001	\$SSRC	= 000000
START3	001230	TEST1	010444	T25A	012734	\$ISK6	= 000001	\$STGSV	= 000000
STKLMT	= 177774	TEST2	011526	T26	013034	\$ISK7	= 000001	\$STGS1	= 000000
STRAP	007510	TEST3	013700	T3	014376	\$LOCTA	= 177777	\$STGS2	= 000000
STX	= 000002	TEST4	014652	T3SAV	014354	\$LSTCN	= 177777	\$STO	= 000000
SUTEST	002056	TEST5	015354	T3SAV1	014356	\$LSTIN	= 000000	\$STAG	= 050000
SW	020215	TIMER	007562	T3SAV2	014360	\$LSTST	= 177777		= 020672
SWCTL	= 000020	TKB	= 177562	T4	015320	\$LSTTA	= 000000		
SWLINE	000174	TKS	= 177560	T4SAV	015260	\$NESTL	= 177777		

. ABS. 020672 000

ERRORS DETECTED: 0

CZLAFD,CZLAFD.LST=SYSDAC.SML/ML,SPMAC.SML ML,CZLAFD.F11
RUN-TIME: 120 101 .6 SECONDS
RUN-TIME RATIO: 60633/222=272.2
CORE USED: 19K (37 PAGES)

C08