

LA36

TERMINAL (DL11 & KL11)
MD-11-DZLAC-C

EP-DZLAC-C-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

...
...
...
...
...
...
...
...
...
...

...

IDENTIFICATION

PRODUCT CODE: MAINDEC-11)DZLAC-C-0
 PRODUCT NAME: L436 TERMINAL (DL11 & KL11 INTERFACE)
 DATE CREATED: SEPTEMBER 1975
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHOR: ROBERT BAKER

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

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

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

COPYRIGHT (C) 1974, 1975 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.1	ABSTRACT
1.2	REQUIREMENTS
1.3	EQUIPMENT AND ASSIGNMENTS
1.4	SOFTWARE
1.5	PRELIMINARY PROGRAMS
2.1	LOADING PROCEDURE AND INITIALIZATION
3.1	STARTING PROCEDURE
3.2	STARTING ADDRESSES
3.3	SWITCH REGISTER CONTROL WITH I/O TESTS
3.4	SWITCH REGISTER WITHOUT I/O TESTS
3.5	KEYBOARD CONTROL WITH I/O TESTS
3.6	KEYBOARD CONTROL WITHOUT I/O TESTS
4.1	OPERATING PROCEDURE
4.2	SWITCH REGISTER CONTROL
4.3	KEYBOARD CONTROL
5.1	TEST DESCRIPTIONS
5.2	PRINTING TESTS
5.2.1	- DATA PATH TEST
5.2.2	- PRINTABLE CHARACTER TEST
5.2.3	- NON-PRINTABLE CHARACTER TEST
5.2.4	- CARRIAGE RETURN TEST
5.2.5	- MULTIPLE LINE FEED TEST
5.2.6	- SINGLE LINE FEED TEST
5.2.7	- BACKSPACE TEST
5.2.8	- OVERPRINT TEST
5.2.9	- PRINTING FREQUENCY SWEEP TEST
5.2.10	- RIBBON FEED TEST
5.2.11	- PRINTER BELL TEST
5.2.12	- LIFE TEST
5.3	ECHO TESTS
5.3.1	- CHARACTER ECHO TEST
5.3.2	- LINE ECHO TEST, FAST RATE
5.3.3	- LINE ECHO TEST, SLOW RATE
5.3.4	- CHARACTER/COOE ECHO TEST
5.3.5	- SELECTED PATTERN ECHO TEST
5.3.6	- BELL ECHO TEST
5.4	OPTION TESTS
5.4.1	- SECONDARY CHARACTER SET OPTION
5.4.2	- SELECTIVE ADDRESSING OPTION
5.4.3	- AUTO ANSWER BACK OPTION
5.4.4	- TOP OF FORM OPTION
5.4.5	- HORIZONTAL TAB OPTION
5.4.6	- VERTICAL TAB OPTION
5.5	KEYBOARD I/O TESTS

1.0 ABSTRACT

THIS DIAGNOSTIC IS DIVIDED INTO FOUR BASIC SECTIONS:

1. A CHECK OF THE CONSOLE TERMINAL INTERFACE LOGIC
2. A CHECK OF THE PRINTING CHARACTERISTICS AND CONTROL LOGIC
3. AN ECHO PORTION DESIGNED TO CHECK THE KEYBOARD AND TO AID IN THE DIAGNOSIS OF TERMINAL PROBLEMS.
4. A CHECK OF THE VARIOUS L336 OPTIONS.

PATTERNS USED BY THE PRINTING TESTS WERE CHOSEN FOR EASE OF VISUAL VERIFICATION. THE ECHO TESTS WERE DESIGNED FOR MAXIMUM FLEXIBILITY, WITH TEST 24 ALLOWING ANY DESIRED PATTERN TO BE USED.

2.0 REQUIREMENTS

2.1 EQUIPMENT AND ASSIGNMENTS

THE DIAGNOSTIC IS WRITTEN TO RUN ON ALL MODELS OF THE PDP11 COMPUTER WITH EITHER A KL11 OR DL11 CONSOLE TERMINAL INTERFACE. IN ADDITION, THE DIAGNOSTIC IS SETUP SUCH THAT UP TO 16 ADDITIONAL DL11'S ASSIGNED BETWEEN ADDRESSES 776500-776676, CAN BE TESTED. TO DRIVE ANY DL11'S BETWEEN ADDRESSES 775610 AND 775600, DEPOSIT 175610 IN DLADR BEFORE STARTING THE DIAGNOSTIC. TO DRIVE ANY DL11'S BETWEEN ADDRESSES 776000 AND 776176, DEPOSIT 176000 IN DLADR BEFORE STARTING THE DIAGNOSTIC. TO TEST A DL11 AT ANY NON-STANDARD ADDRESS OR TO RE-DEFINE ANY TERMINAL AS THE CONSOLE TERMINAL, SET THE ADDRESS OF THE RECEIVER STATUS REGISTER IN CONADD AND THE RECEIVER INTERRUPT VECTOR IN CONVEC BEFORE STARTING THE DIAGNOSTIC. IF A SLOWER DEVICE IS USED AS THE CONSOLE TERMINAL, ALL L336'S WILL APPEAR TO BE RUNNING AT THE SLOWER BAUD RATE. TO OVERCOME THIS, DEPOSIT THE ADDRESS OF THE RECEIVER STATUS REGISTER OF ANY AVAILABLE DL11 IN CONADD AND THE RECEIVER INTERRUPT VECTOR OF THE SAME DL11 IN CONVEC. (REFER TO THE SYMBOL DEFINITIONS IN THE LISTING FOR THE ABOVE MENTIONED LOCATIONS.)

2.2 STORAGE

THE DIAGNOSTIC PROGRAM USES ALL OF 4K OF MEMORY WITH EXCEPTION OF THE AREA USED BY THE ABSOLUTE LOADER.

2.3 PRELIMINARY PROGRAMS

ANY APPLICABLE PDP-11 DIAGNOSTICS SHOULD BE RUN ON THE PROCESSOR. IF ANY ERRORS ARE ENCOUNTERED DURING THE INTERFACE CHECK, REFER TO THE APPROPRIATE INTERFACE DIAGNOSTIC FOR FURTHER HELP IN LOCATING THE PROBLEM IF NEEDED.

3.0 LOADING PROCEDURE AND INITIALIZATION

LOAD THE LA36 DIAGNOSTIC PROGRAM TAPE FOLLOWING NORMAL PROCEDURES. BEFORE STARTING THE PROGRAM, REFER TO THE DESCRIPTION OF THE ROUTINE "DLY". TIME DELAYS USED BY THE PROGRAM ARE A FUNCTION OF THE CPU MODEL AND MEMORY TYPE AND SHOULD BE SET-UP BEFORE RUNNING THE DIAGNOSTIC. THE ROUTINE IS PRESET FOR A PDP-11/05 WITH CORE MEMORY. REFER TO SECTION 2.1 FOR NON-STANDARD TERMINAL ADDRESSES AND FOR TESTING MULTIPLE DL11 INTERFACES.

IF A HARDWARE SWITCH REGISTER DOES NOT EXIST, THE PROGRAM WILL USE THE CONTENTS OF LOCATION 172 AS THE VALUE OF THE SWITCHES. THEREFORE, BE SURE TO LOAD LOCATION 176 WITH THE SWITCH VALUE BEFORE STARTING THE PROGRAM WHEN NOT USING HARDWARE SWITCHES.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

- 200(8) = RUN WITH SWITCH REGISTER CONTROL
 - PERFORM CONSOLE TERMINAL I/O TESTS.
- 204(8) = RUN WITH SWITCH REGISTER CONTROL
 - SKIP CONSOLE TERMINAL I/O TESTS.
- 210(8) = RUN WITH KEYBOARD CONTROL
 - PERFORM CONSOLE TERMINAL I/O TESTS.
- 214(8) = RUN WITH KEYBOARD CONTROL
 - SKIP CONSOLE TERMINAL I/O TESTS.

4.2 SWITCH REGISTER CONTROL WITH I/O TESTS

- A. SET THE SWITCH REGISTER TO 200(8) AND PRESS THE LOAD ADDRESS SWITCH
- B. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.7
- C. SET THE SWITCH REGISTER BIT 8 EQUAL TO 1 OR 0 AND PRESS THE START SWITCH. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER OF DL11'S BEING TESTED. REFER TO SECTION 5.1.5

- D. IF BIT 8 WERE ZERO WHEN STARTING, THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY, AFTER THE ENTIRE SARIES OF I/O TESTS ARE EXECUTED.
- E. IF BIT 8 WAS SET WHEN THE START SWITCH WAS PRESSED, THE ENTIRE SERIES OF I/O TESTS WILL BE EXECUTED AND THE PROGRAM WILL THEN BE WAITING FOR CONTROL VIA THE SWITCH REGISTER. THE CPU WILL HALT AT LOCATION SELHLT.

4.3 SWITCH REGISTER CONTROL - WITHOUT I/O TESTS

SAME AS SECTION 4.2 EXCEPT IN STEP A. SET THE SWITCH REGISTER TO 204(9).

4.4 KEYBOARD CONTROL - WITH I/O TESTS

- A. SET THE SWITCH REGISTER TO 210(8) AND PRESS THE LOAD ADDRESS SWITCH.
- B. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH. IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.7
- C. SET SWITCH 8 AND PRESS THE START SWITCH. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER OF DL11'S BEING TESTED. REFER TO SECTION 5.1.5
- D. IF BIT 8 WAS ZERO WHEN SPARTING, THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY AFTER THE ENTIRE SERIES OF I/O TESTS ARE EXECUTED.
- E. IF BIT 8 WERE SET WHEN THE START SWITCH WAS PRESSED, THE ENTIRE SERIES OF I/O TESTS WILL BE EXECUTED FOLLOWED BY THE SELECT TEST MESSAGE. THE PROGRAM WILL THEN BE WAITING FOR A TEST SELECTION VIA ANY TARMINAL KEYBOARD. RAFER TO SECTION 5.2

4.5 KEYBOARD CONTROL - WITHOUT I/O TESTS

SAME AS SECTION 4.4 EXCEPT IJ STEP A, SET THE SWITCH REGISTER TO 214 (8).

5.0 OPERATING PROCEDURE

THE PROGRAM CAN BE CONTROLLED IN EITHER OF TWO METHODS: BY THE CONSOLE SWITCH REGISTER OR FROM THE KEYBOARD OF THE TERMINAL(S) UNDER TEST.

5.: SWITCH REGISTER CONTROL

THE VARIOUS SWITCHES AND THEIR FUNCTIONS ARE LISTED BELOW. SWITCHES MAY BE CHANGED AND SET AS DESIRED EXCEPT AS NOTED IN THE SPECIFIC SWITCH DESCRIPTIONS. REFER TO THE DETAILED SWITCH DESCRIPTIONS FOR FURTHER, MORE COMPLETE INFORMATION.

SWITCH NUMBER	DESCRIPTION
15	1(UP) = HALT AT END OF TEST 0(DOWN) = CONTINUE TEST SEQUENCE
14	1(UP) = CONTINUE ON ERROR 0(DOWN) = HALT ON ERROR
13	1(UP) = DRIVE ONLY CONSOLE TERMINAL 0(DOWN) = DRIVE ALL TERMINALS
11	1(UP) = LOOP ON INDIVIDUAL TEST 0(DOWN) = NORMAL TEST SEQUENCE
8	1(UP) = RUN TEST ONCE AND HALT 0(DOWN) = LOOP ON TEST SEQUENCE
5-0	TEST NUMBER SELECTION
7-0	NUMBER OF COLUMNS AT START-UP

5.1.1 SWITCH 15

WITH SWITCH 15 IN THE UP POSITION, THE PROGRAM WILL HALT AT THE END OF THE CURRENT TEST. REPLACING SWITCH 15 TO THE DOWN POSITION AND PRESSING CONTINUE WILL CONTINUE THE NORMAL TEST OPERATION. DURING THE HALT, ANY OF THE CONTROL SWITCHES MAY BE CHANGED OR SET AS DESIRED.

5.1.2 SWITCH 14

PLACING SWITCH 14 IN THE UP POSITION WILL CAUSE THE PROGRAM TO CONTINUE ON ERRORS DURING ANY OF THE I-O TESTS ONLY. WITH SWITCH 14 DOWN, THE PROGRAM WILL HALT (AT ERRHLT) ON ANY ERRORS DURING THE I-O TESTS WITH THE LOCATION OF THE ERROR IN RD. PRESSING CONTINUE WILL CAUSE THE PROGRAM TO CONTINUE IF SWITCH 14 IS DOWN. WITH SWITCH 14 UP, PRESSING CONTINUE WILL CAUSE THE PROGRAM TO LOOP ON THE ERROR.

NOTE

ERRORS CAN OCCUR ONLY DURING THE I/O TESTS. THE TERMINAL IS CONNECTED TO A SERIAL LINE AND THERE IS NO INFORMATION RETURNING TO THE CPU FROM THE TERMINAL. THE PROGRAM CANNOT DETECT ERRORS IN THE TERMINAL, ONLY ERRORS IN THE INTERFACE CAN BE DETECTED.

5.1.3 SWITCH 13

PLACING SWITCH 13 IN THE DOWN POSITION WILL CAUSE THE DRIVING OF ALL MULTIPLE TERMINALS DURING THE PRINTER TESTS ONLY. IF SWITCH 13 IS UP, ONLY THE CONSOLE TERMINAL IS DRIVEN.

** NOTE: SWITCH 13 SHOULD ONLY BE CHANGED WHEN THE PROGRAM IS WAITING FOR A TEST SELECTION.

5.1.4 SWITCH 11

PLACING SWITCH 11 UP AT ANY TIME WILL CAUSE THE PROGRAM TO LOOP ON THE CURRENT TEST AS LONG AS SWITCH 11 REMAINS UP. REPLACING SWITCH 2 DOWN WILL CAUSE THE PROGRAM TO RESUME NORMAL OPERATION AT THE COMPLETION OF THE TEST.

5.1.5 SWITCH 8

WITH SWITCH 8 IN THE DOWN POSITION THE PROGRAM WILL CONTINUE TO LOOP THROUGH THE PRESENT TEST SEQUENCE. PLACING SWITCH 8 UP WILL CAUSE THE PROGRAM TO HALT (AT SELHLT) AT THE COMPLETION OF THE CURRENT TEST. AFTER THE HALT SET THE CONTROL SWITCHES AS DESIRED AND SET SWITCHES 5 TO 0 TO THE NEXT DESIRED TEST NUMBER THEN PRESS CONTINUE TO START THE TEST.

WHEN STARTING THE DIAGNOSTIC THE OPERATOR CAN SELECT A SPECIFIC TEST RATHER THAN AUTOMATICALLY STARTING THE PRINTING TEST SEQUENCE BY SETTING SWITCH 8 UP BEFORE STARTING THE DIAGNOSTIC. UPON COMPLETION OF THE I/O TEST SEQUENCE (IF BEING RUN) THE PROGRAM WILL EITHER HALT AT SELHLT WAITING FOR A TEST SELECTION VIA THE SWITCH REGISTER OR PRINT THE SELECT TEST MESSAGE AND WAIT FOR A TEST SELECTION FROM ANY KEYBOARD. REFER TO SECTION 4 FOR FURTHER INFORMATION.

5.1.6 SWITCHES 5 TO 0

SWITCHES 5 TO 0 ARE USED TO SELECT SPECIFIC TESTS WHEN UNDER SWITCH REGISTER CONTROL. TEST NUMBERS ARE ALWAYS IN OCTAL.

5.1.7 SWITCHES 7 TO 0 (AT START-UP ONLY)

AT START-UP ONLY, SWITCHES 7 TO 0 ARE USED TO SET THE DESIRED MAXIMUM NUMBER OF COLUMNS THE DIAGNOSTIC IS TO TEST. IF THE NUMBER SET IS GREATER THAN 132(10) OR LESS THAN 30(10), THE PROGRAM WILL DEFAULT TO 132(10). THE VALUE SET MUST BE IN OCTAL FORM. THUS, FOR NORMAL OPERATION - TESTING THE FULL 132 (10) COLUMNS - LEAVE SWITCHES 7 - 0 DOWN.

5.2 KEYBOARD CONTROL

THE PROGRAM WILL BE UNDER KEYBOARD CONTROL WHENEVER THE DIAGNOSTIC IS STARTED AT LOCATION 210 OR 214. SWITCHES ON THE CONSOLE SWITCH REGISTER WILL HAVE NO EFFECT WHEN UNDER TERMINAL CONTROL EXCEPT FOR SWITCH 15. THE I/O TESTS CANNOT BE SELECTED WHEN UNDER KEYBOARD CONTROL.

TO STOP A TEST AT ANY TIME, TYPE THE "RUBOUT" OR "DELETE" KEY ON ANY KEYBOARD. ANY TERMINAL MAY STOP THE TEST AND SELECT THE NEXT TEST IF SWITCH 13 IS DOWN. WHEN A TEST IS STOPPED BY TYPING A "RUBOUT" OR DELETE, THE TEST WILL TERMINATE AND THE FOLLOWING MESSAGE WILL BE TYPED:

SELECT TEST NUMBER

AT THIS TIME, TYPE THE DESIRED TEST NUMBER FOLLOWED BY ANY ONE OF THE FOLLOWING CONTROL CHARACTERS:

- . (PERIOD) = RUN THE SELECTED TEST ONCE AND RETURN FOR ANOTHER TEST SELECTION.
- L = LOOP ON THE SELECTED TEST UNTIL A "RUBOUT" IS TYPED.
- S = START THE TEST SEQUENCE WITH THE SELECTED TEST. CONTINUE TO LOOP ON THE PRINTING TEST SEQUENCE UNTIL A "RUBOUT" IS TYPED.

THE "L" OR "S" MAY BE EITHER UPPER OR LOWER CASE, BUT THE TEST NUMBER MUST ALWAYS BE A 2 DIGIT OCTAL NUMBER. THE TEST NUMBER AND TERMINATOR ARE ECHOED BY THE PROGRAM, THUS EACH CHARACTER WILL BE PRINTED TWICE IF THE TERMINAL IS IN HALF DUPLEX. FOR ALL ECHO TESTS, THE "L" AND "S" WILL ONLY RUN THE TEST ONCE (THE SAME AS IF TYPING A PERIOD). FOR ALL OPTION TESTS, THE "S" WILL ONLY RUN THE TEST ONCE (THE SAME AS IF TYPING A PERIOD). HOWEVER, TYPING AN "L" WILL CAUSE THE PROGRAM TO LOOP ON THE SELECTED TEST. IF AN ERROR IS DETECTED IN THE TEST SELECTION (ILLEGAL TEST NUMBER OR CONTROL CHARACTER) A QUESTION MARK IS PRINTED AND THE MESSAGE WILL BE REPEATED.

EXAMPLE:

!!!	AAA	AAA
:::	BBB	BBB
###	CCC	CCC
\$\$\$	DDD	DDD
%%%	EEE	EEE
888	FFF	FFF
...	GGG	GGG
((HHH	HHH
)))	III	III
***	JJJ	JJJ
+++	KKK	KKK
...	LLL	LLL
--	MMM	MMM
...	NNN	NNN
///	OOO	OOO
000	PPP	PPP
111	QQQ	QQQ
222	RRR	RRR
333	SSS	SSS
444	TTT	TTT
555	UUU	UUU
666	VVV	VVV
777	WWW	WWW
888	XXX	XXX
999	YYY	YYY
:::	ZZZ	ZZZ
...	III	
<<<	\ \ \	
===]]]	
>>>		
???		

6.1.3 TEST 2 - NON-PRINTABLE CHARACTER TEST

THIS TEST CHECKS ALL NON-PRINTABLE CHARACTERS THAT HAVE NO CONTROL FUNCTION IN THE LA36 TERMINAL OR THE LA36 OPTIONS (SUCH AS CR, LF, BS, & BEL). FIRST THE ASCII CODE WILL BE PRINTED FOLLOWED BY THE MNEMONIC AFTER A FEW SEPARATING SPACES. FOLLOWING THE MNEMONIC, THE ACTUAL CONTROL CHARACTER WILL BE SENT THREE TIMES AND NOTHING SHOULD HAPPEN AT THE PRINTER. THIS PATTERN IS REPEATED, THREE TIMES ON A LINE, UNTIL ALL OF THE NON-PRINTING CHARACTERS HAVE BEEN TESTED.

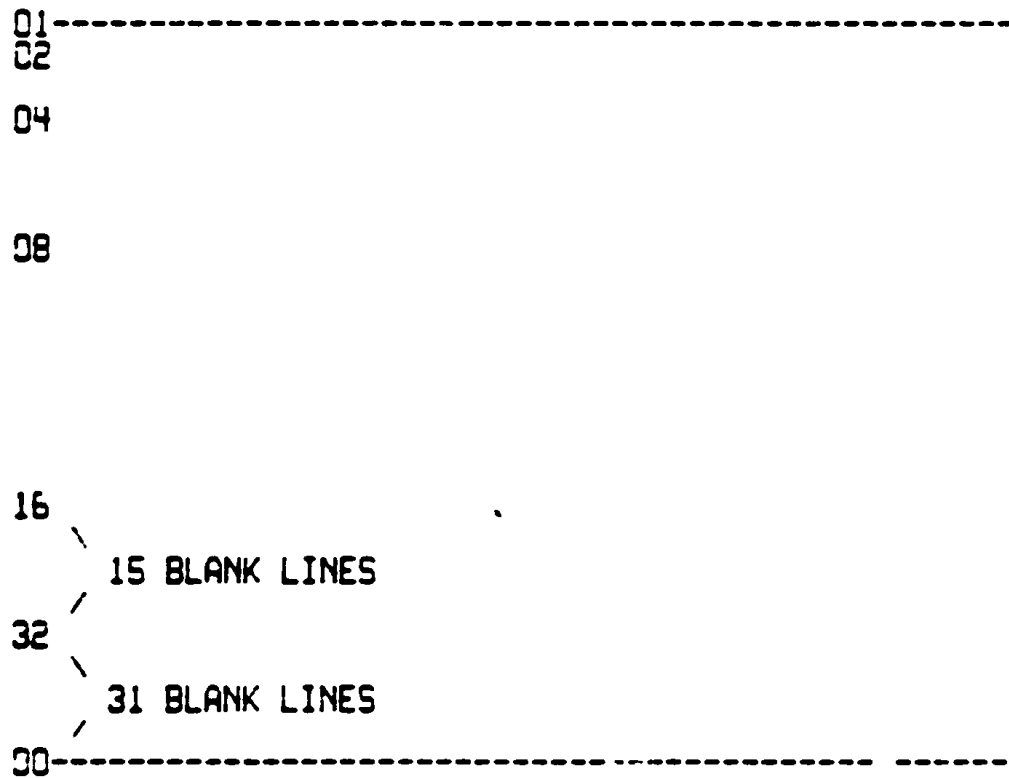
WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

6.1.5 TEST 4 - MULTIPLE LINE FEED TEST

THIS TEST CHECKS THE LINE FEED CAPABILITY OF THE PRINTER BY SENDING VARIOUS GROUPS OF LINE FEEDS INTERSPACED WITH REFERENCE LINES. THE NUMBER PRINTED AS THE REFERENCE LINE INDICATES THE NUMBER OF LINE FEEDS THAT FOLLOW. THE FIRST AND LAST LINES ALSO CONTAIN A STRING OF DASHES AS REFERENCE POINTS FOR MEASURING, THE TOTAL DISTANCE IS 63(10) LINES BETWEEN THE TWO DASHED LINES.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY CARRIAGE RETURN, THE NUMBER PRINTED WILL INDICATE ONE LESS THAN THE NUMBER OF LINE FEEDS (THE NUMBER OF BLANK LINES) THAT FOLLOW. THE TOTAL DISTANCE BETWEEN THE TWO DASHED LINES WILL THEN BE 69 LINES.

EXAMPLE:



6.1.6 TEST 5 - SINGLE LINE FEED TEST

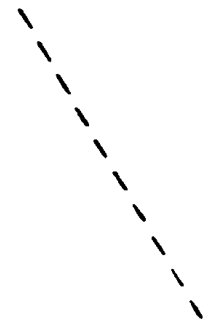
THIS TEST IS DESIGNED TO CHECK THE TIMING OF SINGLE LINE FEEDS AND THE CAPABILITY OF DOING LINE FEEDS IN ALL COLUMNS. TWO REFERENCE LINES ARE USED BY THIS TEST (AND TEST 6) WHICH ALSO CAN BE USED TO EASILY CHECK THE NUMBER OF COLUMNS THE PRINTER IS PRINTING.

THE FIRST REFERENCE LINE CONTAINS 130(10) ZEROES FOLLOWED BY TWO 2'S IF TESTING 132(10) COLUMNS. IF LESS THAN 132 COLUMNS, THE LINE WILL CONTAIN 0'S FOR TWO LESS THAN THE MAXIMUM NUMBER OF COLUMNS FOLLOWED BY THE TWO 2'S. THIS REFERENCE LINE IS A QUICK CHECK FOR 132(10) COLUMNS IF TESTING THE FULL 132(10) COLUMNS. THE SECOND REFERENCE LINE PRINTS A STRING OF NUMBERS (1 TO 9 & 0) REPEATED TO THE MAXIMUM COLUMN. THIS LINE, AGAIN, CAN BE USED AS A QUICK CHECK OF THE NUMBER OF COLUMNS.

THE LINE FEED TEST IS ACCOMPLISHED BY: PRINTING THE FIRST REFERENCE LINE OF 0'S AND TWO 2'S; THEN EITHER SENDING 60(10) 3'S, IF TESTING 132(10) COLUMNS, OR WAITING 1.8 SECONDS FOR AN LCV, IF TESTING LESS THAN 132(10) COLUMNS. IF TESTING 132(10) COLUMNS, NOTHING SHOULD HAPPEN, EXCEPT FOR AN LCV, AT THE END OF THE LINE. THE 3'S SHOULD BE LOST AND NEVER PRINTED. AFTER THE LCV, WITH THE PRINT HEAD AT THE EXTREME RIGHT, A CARRIAGE RETURN - LINE FEED WILL BE SENT FOLLOWED BY REPEATED BACKSLASHES "\" AND LINEFEEDS TO PRINT A DIAGONAL LINE DOWN THE PAPER. WHEN A BACKSLASH IS PRINTED IN THE MAXIMUM COLUMN, A CARRIAGE RETURN WILL BE SENT IMMEDIATELY AFTER THE LINE FEED AND THE SECOND REFERENCE LINE OF SEQUENTIAL NUMBERS WILL BE PRINTED. AFTER COMPLETING THE LINE, A CARRIAGE RETURN - LINE FEED WILL BE SENT AND THE PROGRAM WILL WAIT ONE SECOND FOR THE CARRIAGE RETURN FUNCTION TO COMPLETE. AFTER THE DELAY, THE REFERENCE LINE WILL BE REPEATED, THE LAST LINE BEING GUARANTEED TO BE CORRECT. ANY TIMING PROBLEMS DURING THE LINE FEEDS WILL SHOW AS MISS PRINTS OR MISSING CHARACTERS DURING THE FIRST 16(10) CHARACTERS OF THE MIDDLE REFERENCE LINE. ALSO, ANY PAPER FEED PROBLEMS WILL CAUSE MISS-ALIGNMENT OF THE SLASHES FORMING THE DIAGONAL LINE.

EXAMPLE:

0000000022



1234567890

1234567890

6.1.7 TEST 6 - BACKSPACE TEST

THIS TEST IS DESIGNED TO TEST THE PRINT TIMING AS IN TEST 5 AS WELL AS THE BACKWARD AND FORWARD MOVEMENT OF THE PRINT SOLENOID HEAD.

THE TEST CONSISTS OF THE SAME FIRST REFERENCE LINE AS IN TEST 5 THEN A CARRIAGE RETURN-LINE FEED. A FULL LINE IS THEN PRINTED USING THE FOLLOWING PATTERN:

FORWARD SLASH "/"
BACKSPACE "
BACK SLASH "\"

THIS PATTERN PRODUCES A LINE OF ALL X'S. THE TWO SLASHES SHOULD CROSS EXACTLY AT THE MIDDLE, PRODUCING THE X CHARACTER. WHEN THE LINE IS COMPLETED A CARRIAGE RETURN-LINE FEED IS SENT AND THE LAST TWO REFERENCE LINES ARE PRINTED AS IN TEST 5. ANY TIMING PROBLEMS WILL SHOW IN THE FIRST 16(10) CHARACTERS OF THE MIDDLE REFERENCE LINE; AGAIN AS IN TEST 5.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

EXAMPLE:

00000000000000000000000000000022
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
123456789012345678901234567890
123456789012345678901234567890

6.1.9 TEST 7 - OVERPRINT TEST

THIS TEST IS DESIGNED TO CHECK THE SPACING AND REPEATABLE PRINTING CHARACTERISTICS OF THE PRINTER. THREE ROWS OF CHARACTERS ARE EACH OVERPRINTED TWO TIMES. THE ROWS CONSIST OF THE FOLLOWING CHARACTERS ALTERNATED ACROSS THE LINE:

ROW 1	M-SP
ROW 2	SP-Q
ROW 3	S-SP

THE RESULTING PATTERN WILL BE A CHECKERBOARD PATTERN AND THE OVERPRINTED CHARACTERS SHOULD BE ALIGNED PROPERLY WITH THE INITIAL CHARACTERS.

EXAMPLE:

```

M M M M M M M M M M M M M M M
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
S S S S S S S S S S S S S S S

```

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THE LINES WILL NOT BE OVERPRINTED. THERE WILL BE THREE LINES OF EACH CHARACTER WITH A BLANK LINE BETWEEN EACH GROUP OF CHARACTERS. THE CHARACTERS IN EACH GROUP SHOULD BE IN THE SAME COLUMNS.

EXAMPLE:

```

M M M M M M M M M M M
M M M M M M M M M M M
M M M M M M M M M M M

Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q

S S S S S S S S S S S
S S S S S S S S S S S
S S S S S S S S S S S

```


6.1.11 TEST 12 - PRINTER BELL TEST

THIS TEST CHECKS THE PRINTER BELL BUFFER TO INSURE THAT EIGHT BELLS ARE DISTINCTLY HEARD, EVEN WHEN SENT AT THE MAXIMUM TRANSFER RATE. THE PROGRAM SENDS 8 BELL CODES AT THE MAXIMUM RATE TO THE PRINTER THEN WAITS 2.5 SECONDS TO ALLOW THE OPERATOR TO HEAR THE BELLS.

6.1.12 TEST 17 - LIFE TEST

THIS TEST RUNS CONTINUOUSLY AND IS RUN AS AN INDIVIDUAL, SPECIAL TEST. IT IS NOT PART OF THE STANDARD PRINTING TEST SEQUENCE.

THIS TEST PRINTS 2 LINES OF EACH PRINTABLE CHARACTER AND THEN REPEATS CONTINUOUSLY. THE SECOND LINE OF EACH CHARACTER IS OVERPRINTED 4 TIMES TO CONSERVE PAPER. AT THE END OF EACH COMPLETE PASS THROUGH THE CHARACTER SET A MESSAGE IS PRINTED INDICATING THE NUMBER OF PASSES EXECUTED. IF ANY CHARACTER (EXCEPT "RUBOUT") IS TYPED ON THE KEYBOARD DURING THIS TEST, THE PATTERN WILL CHANGE AND RESTART WITH THE TYPED CHARACTER. THIS WILL ONLY HAPPEN IF KEYBOARD CONTROL IS IN USE.

EXAMPLE:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

IF THE AUTO LINE FEED OPTION IS SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THE TEST WILL PRINT SIX LINES OF EACH CHARACTER WITH A BLANK LINE BETWEEN THE FIRST AND SECOND LINES AS WELL AS BETWEEN EACH GROUP OF CHARACTERS.

EXAMPLE:

```
AAAAAAAAAAAAAAAA  
  
AAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
  
BBBBBBBBBBBBBBBB  
  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB
```

6.2 ECHO TESTS

THESE TESTS ARE DESIGNED AS A TEST OF THE KEYBOARD AND AN AID IN ISOLATING TROUBLES WITHIN THE TERMINAL. AT THE BEGINNING OF EACH TEST, THE TEST NUMBER WILL BE PRINTED INDICATING WHICH TEST IS BEING EXECUTED. TYPING A "RUBOUT" OR "DELETE" AT ANY TIME, WHETHER IN KEYBOARD CONTROL OR NOT, WILL EXIT THE CURRENT ECHO TEST AND PRINT A TEST TERMINATION MESSAGE. IF IN KEYBOARD CONTROL, THE SELECT TEST MESSAGE WILL BE PRINTED AND THE PROGRAM WILL AWAIT A TEST SELECTION AS USUAL. IN SWITCH REGISTER CONTROL, THE PROGRAM WILL HALT (AT SELLHLT) WAITING FOR CONTROL VIA THE SWITCH REGISTER. A DETAILED DESCRIPTION OF EACH TEST FOLLOWS:

6.2.1 TEST 20 - CHARACTER ECHO TEST

THIS TEST IS DESIGNED TO OPERATE THE TERMINAL IN A SIMULATED LOCAL MODE. ANY CHARACTER TYPED ON THE KEYBOARD (EXCEPT A "RUBOUT") WILL BE ECHOED TO THE PRINTER.

IF THE LA36 TERMINAL IS IN HALF DUPLEX WITH THE AUTO LINE FEED OPTION AVAILABLE, TYPING A CARRIAGE RETURN MAY CAUSE A GARBLED RESPONSE ON THE TERMINAL DURING THIS TEST.

6.2.2 TEST 21 - LINE ECHO TEST, FAST RATE

THIS TEST CONTINUALLY SENDS FULL LINES OF ANY CHARACTER UP TO THE MAXIMUM COLUMN WIDTH. THE TEST PRINTS A "Q" CHARACTER WHEN STARTED UNTIL A KEY IS TYPED ON THE KEYBOARD. THE PROGRAM WILL THEN SEND THE TYPED CHARACTER UNTIL ANOTHER CHARACTER IS TYPED OR THE TEST IS TERMINATED BY TYPING A "RUBOUT". THE CHARACTERS ARE TRANSMITTED AT THE MAXIMUM RATE WITH A CARRIAGE RETURN-LINE FEED INSERTED AFTER EVERY 132(10) PRINTABLE CHARACTERS.

IF THE LA36 IS IN HALF DUPLEX WHEN RUNNING THIS TEST, CHARACTERS MAY BE LOST OR GARBLED WHENEVER A CHARACTER IS TYPED ON THE KEYBOARD.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

6.2.3 TEST 22 - LINE ECHO TEST, SLOW RATE

THIS TEST IS IDENTICAL TO TEST 21 EXCEPT A DELAY OF 1.8 SECONDS IS INSERTED BETWEEN EACH CHARACTER TO ALLOW THE PRINT HEAD TO PERFORM AN LCV BETWEEN CHARACTERS.

6.2.4 TEST 23 - CHARACTER CODE ECHO TEST

THIS TEST WILL PRINT THE OCTAL CODE RECEIVED BY THE PROCESSOR FOLLOWED BY THE CHARACTER OR THE MNEMONIC OF THE CHARACTER EVERY TIME A KEY IS PRESSED ON THE KEYBOARD. THE PARITY OF THE RECEIVED CODE WILL BE INDICATED AS EITHER ODD OR EVEN. ALLOW SUFFICIENT TIME BETWEEN CHARACTERS FOR THE LINE TO BE PRINTED.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

EXAMPLE:

301	A	ODD
263	3	ODD
215	CR	EVEN
240	SP	EVEN

6.2.5 TEST 24 - SELECTED PATTERN ECHO TEST

THIS TEST IS DESIGNED TO GIVE MAINTENANCE THE FLEXIBILITY TO CHOOSE THEIR OWN PATTERNS FOR ISOLATING ANY SPECIFIC PROBLEMS WHICH MAY ARISE IN THE FIELD.

TYPE ANY CHARACTERS (EXCEPT CONTROL-C AND RUBOUT) AND EACH CHARACTER WILL BE ECHOED AS TYPED. A MAXIMUM OF 256(10) CHARACTERS MAY BE INPUTTED. NO CARRIAGE RETURNS OR LINE FEEDS ARE INSERTED BY THE PROGRAM. ALL CHARACTERS MUST BE INPUTTED BY THE OPERATOR. TO TERMINATE THE INPUT STRING TYPE A CONTROL-C, THE PROGRAM WILL THEN CONTINUALLY ECHO THE INPUTTED PATTERN. TO STOP THE PRINTING, TYPE CONTROL-C. THE PROGRAM WILL STOP PRINTING THE PATTERN AND WILL WAIT FOR EITHER ANOTHER PATTERN INPUT TERMINATED BY A CONTROL-C, OR THE SAME PATTERN MAY BE USED AGAIN BY TYPING CONTROL-C. TO EXIT THE TEST AT ANY TIME, TYPE A "RUBOUT".

WHEN ANY OPTIONS ARE AVAILABLE, BE CAREFUL WHAT CHARACTERS OR CHARACTER SEQUENCES ARE SELECTED.

6.2.6 TEST 25 - BELL ECHO TEST

THIS TEST IS DESIGNED TO TEST THE BELL ON COLUMN 64 IF TYPING HAS OCCURED ON THE LINE. THE TEST PRINTS A MESSAGE:

TYPE ANY PRINTABLE CHARACTER AND LISTEN FOR BELL

AFTER THE TEST MESSAGE IS PRINTED, TYPE ANY PRINTABLE CHARACTER ON THE KEYBOARD. THE CHARACTER WILL BE ECHOED AND THE BELL SHOULD RING. THE MESSAGE WILL THEN BE TYPED AGAIN. TYPE THE "RUBOUT" KEY TO TERMINATE THE TEST AT ANY TIME.

6.3 OPTION TESTS

THESE TESTS ARE DESIGNED AS A TEST OF THE VARIOUS OPTIONS IN WHATEVER COMBINATIONS THEY ARE AVAILABLE IN THE LA36. AT THE BEGINNING OF EACH TEST, THE TEST NUMBER WILL BE PRINTED INDICATING WHICH TEST IS BEING EXECUTED. TYPING A "RUBOUT" OR "DELETE" AT ANY TIME, WHETHER IN KEYBOARD CONTROL OR NOT, WILL EXIT THE CURRENT OPTION TEST. A DETAILED DESCRIPTION OF EACH TEST FOLLOWS:

6.3.1 TEST 30 - SECONDARY CHARACTER SET OPTION

THIS TEST IS DESIGNED TO TEST THE SECONDARY CHARACTER SET OPTION, TESTING THE ABILITY TO SELECT EITHER CHARACTER SET UNDER SOFTWARE CONTROL FROM THE CPU AND PRINTING THE CORRECT CHARACTERS WITHIN EACH CHARACTER SET.

A NUMBER IS PRINTED AT THE LEFT MARGIN INDICATING WHICH CHARACTER SET IS BEING PRINTED. #1 INDICATES THE PRIMARY SET AND #2 INDICATES THE SECONDARY SET (APL). AFTER THE NUMBER, THE APPROPRIATE SHIFT IN (SI) OF SHIFT OUT (SO), WILL BE SENT FOLLOWED BY THE ENTIRE PRINTABLE CHARACTER SET. IF LESS THAN 98 COLUMNS ARE BEING TESTED, A CARRIAGE RETURN - LINE FEED WILL BE INSERTED IN THE APPROPRIATE PLACES. THIS WILL BE REPEATED, ALTERNATING BETWEEN PRIMARY AND SECONDARY SETS UNTIL 16 LINES HAVE BEEN PRINTED (IF USING 98 OR MORE COLUMNS). THERE WILL BE A BLANK LINE BETWEEN EACH PAIR OF LINES TO SEPARATE EACH GROUPING. CHANGE LOCATION "T30SC" AT THE END OF TEST30 TO 377 (8) IF USING 8 BIT SELECTION CODE RATHER THAN THE SI AND SO TO SELECT CHARACTER SETS. THE TEST WILL THEN SET OR CLEAR BIT 8 INSTEAD OF SENDING THE SI OR SO TO SELECT CHARACTER SETS.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EACH RECEIVED CARRIAGE RETURN, THERE WILL BE EXTRA BLANK LINES EVERY PLACE A CARRIAGE RETURN IS SENT.

EXAMPLE:

```
#1= !"#5%8'().....PRIMARY CHARACTER SET.....  
#2= !"#5%8'().....SECONDARY CHARACTER SET.....  
  
#1= !"#5%8'().....PRIMARY CHARACTER SET.....  
#2= !"#5%8'().....SECONDARY CHARACTER SET.....
```

6.3.2 TEST 31 - SELECTIVE ADDRESSING OPTION

THIS TEST IS DESIGNED TO TEST THE VARIOUS FUNCTIONS OF THE SELECTIVE ADDRESSING OPTION. THE TEST FIRST SENDS AN "EOT" (004) TO DISABLE ALL TERMINALS AND TRIES TO PRINT AN ERROR MESSAGE. THE ERROR MESSAGE SHOULD NOT BE PRINTED ON ANY TERMINAL WITH THE SELECTIVE ADDRESSING OPTION. THEN A "BEL" (007) AND "STX" (002) ARE SENT TO SELECT ALL TERMINALS. AT THIS POINT THE TEST NUMBER IS PRINTED ON ALL TERMINALS. THUS, IF AN ERROR MESSAGE IS PRINTED BEFORE THE TEST NUMBER, THE EOT DID NOT DE-SELECT THE TERMINAL WHERE THE MESSAGE WAS PRINTED.

THE TEST NEXT SENDS AN EOT DIRECTLY FOLLOWED BY A STX, WITH NO SELECT CHARACTER. AGAIN, THE ERROR MESSAGE IS SENT TO ALL TERMINALS, WHICH SHOULD NOW BE ALL DE-SELECTED. THE ERROR MESSAGE SHOULD NOT BE PRINTED ON ANY TERMINAL WITH THE SELECTIVE ADDRESSING OPTION.

THE NEXT SERIES OF CHECKS ARE MADE ON THE GROUP SELECT CHARACTER. A TABLE LOCATED AT THE END OF TEST 31 IN THE LISTING IS USED TO TEST VARIOUS GROUP SELECT CHARACTERS. THE FIRST ZERO ENCOUNTERED IN THE TABLE WILL INDICATE THE END OF THE TABLE AND THE TEST WILL GO TO THE NEXT SERIES OF CHECKS ON THE OPTION. THE TABLE IS PRESET WITH A SINGLE GROUP SELECT CHARACTER, THE LETTER "G", BUT ALLOWS ROOM TO TEST UP TO 8 DIFFERENT SELECT CODES. THIS TABLE SHOULD BE CHANGED TO CONTAIN THE VARIOUS GROUP SELECT CHARACTERS DESIRED TO TEST WITH ONE ASCII CODE PER LOCATION. THE TEST WILL THEN USE THE VARIOUS GROUP SELECT CHARACTERS TO SELECT TERMINALS AND PRINT A MESSAGE ON EACH SELECTED TERMINAL INDICATING THE GROUP SELECT CHARACTER USED. CHECK THAT THE CORRECT GROUP SELECT CHARACTER HAS ENABLED EACH TERMINAL. ALSO, IT MAY BE HELPFUL TO PLACE UNUSED SELECT CHARACTERS IN THE TABLE TO CHECK THAT THEY DO NOT SELECT TERMINALS. IF AN ERROR MESSAGE WAS PRINTED BETWEEN THE TEST NUMBER AND THE GROUP SELECT MESSAGE, THE TERMINAL WHERE THE MESSAGE WAS PRINTED WAS SELECTED BY AN EOT AND STX WITH NO SELECT CHARACTER BETWEEN THEM.

THE LAST SERIES OF CHECKS ARE MADE ON THE UNIQUE SELECT CHARACTER. A TABLE LOCATED AT THE END OF TEST 31 IN THE LISTING IS USED TO TEST VARIOUS UNIQUE SELECT CHARACTERS. THE FIRST ZERO ENCOUNTERED IN THE TABLE WILL INDICATE THE END OF THE TABLE. THE PROGRAM WILL SELECT ALL TERMINALS USING THE BEL CODE BEFORE EXITING THE TEST. THE TABLE IS PRESET WITH A SINGLE UNIQUE SELECT CHARACTER, THE LETTER "U", BUT ALLOWS ROOM TO TEST UP TO 16 DIFFERENT UNIQUE SELECT CODES. THIS TABLE SHOULD BE CHANGED TO CONTAIN THE VARIOUS UNIQUE SELECT CHARACTERS DESIRED TO TEST, WITH ONE ASCII CODE PER LOCATION. MAKE SURE THAT EACH CHARACTER IN THE TABLE IS A VALID UNIQUE SELECT CODE OR THE DIAGNOSTIC WILL HANG DURING THIS PORTION OF THE TEST. USING EACH UNIQUE SELECT CHARACTER IN TURN, THE TEST WILL PERFORM THE REMAINING CHECKS OF THE SELECTIVE ADDRESSING OPTION.

THE TEST WILL SEND AN EOT FOLLOWED BY THE CURRENT UNIQUE SELECT CHARACTER. BEFORE THE STX IS SENT, THE TEST WILL TRY TO PRINT THE ERROR MESSAGE ON ALL TERMINALS. THEN THE STX WILL BE SENT AND A MESSAGE WILL BE PRINTED TO INDICATE THE UNIQUE SELECT CHARACTER USED. CHECK THAT THE CORRECT UNIQUE SELECT CHARACTER HAS ENABLED EACH TERMINAL. IF AN ERROR MESSAGE IS PRINTED BEFORE THE UNIQUE SELECT MESSAGE, THE TERMINAL WHERE THE MESSAGE WAS PRINTED WAS ENABLED BEFORE THE STX WAS RECEIVED. A MESSAGE WILL THEN BE PRINTED TELLING THE OPERATOR TO TYPE ANY PRINTABLE CHARACTER TO CHECK THAT THE KEYBOARD IS ENABLED. WHATEVER CHARACTER IS TYPED WILL BE ECHOED TO THE TERMINAL.

THE FINAL SECTION OF THE TEST WILL USE A DUMMY SELECT CHARACTER. THE ASCII CODE FOR THIS SELECT CHARACTER IS LOCATED BETWEEN THE TWO SELECT CHARACTER TABLE AT THE END OF THE TEST. THIS LOCATION SHOULD CONTAIN THE ASCII CODE OF ANY UNUSED SELECT CHARACTER. THE TEST WILL SEND AN EOT FOLLOWED BY THE DUMMY SELECT CHARACTER AND AN STX. THE ERROR MESSAGE WILL BE LOADED TO ALL TERMINALS AND SHOULD NOT BE PRINTED ON ANY TERMINALS SINCE ALL SHOULD BE DE-SELECTED. NEXT AN ETX (003) FOLLOWED BY THE CURRENT UNIQUE SELECT CHARACTER AND AN STX WILL BE SENT AND A PRINTED MESSAGE WILL INDICATE THE SELECT CHARACTER USED. ANOTHER EXT WILL BE SENT, FOLLOWED BY THE DUMMY SELECT CHARACTER AND AN STX THIS TIME. A MESSAGE WILL AGAIN BE PRINTED INDICATING THE CURRENT UNIQUE SELECT CHARACTER. ALL SELECTED TERMINALS SHOULD REMAIN SELECTED AND NO OTHER TERMINALS SHOULD GET SELECTED.

6.3.3 TEST 32 - ANSWER BACK OPTION

THIS TEST IS DESIGNED TO TEST THAT THE ANSWER BACK OPTION SENDS THE CORRECT MESSAGE UPON RECEIPT OF AN ENQ (005) OR UPON TYPING CONTROL-E OR THE HERE IS KEY ON THE KEYBOARD. THE TEST WILL SEND AN ENQ (005), READ THE MESSAGE, AND THEN PRINT OUT THE MESSAGE ON THE LA36. THE TEST WILL THEN ASK THE OPERATOR TO DEPRESS THE HERE IS KEY, READ THE MESSAGE, AND THEN PRINT OUT THE MESSAGE. FINALLY, THE TEST WILL TELL THE OPERATOR TO DEPRESS THE CONTROL-E KEY, READ THE MESSAGE, AND PRINT OUT THE MESSAGE. IF THE SELECTIVE ADDRESSING OPTION IS AVAILABLE, THE AUTO ANSWER BACK OPTION WILL NOT RESPOND TO ANOTHER ENQ AFTER THE FIRST ONE RECEIVED. THUS, YOU MAY HAVE TO DEPRESS THE RUBOUT KEY TO EXIT THE TEST.

6.3.4 TEST 33 - TOP OF FORM OPTION

THIS TEST IS DESIGNED TO TEST THE FORM FEED CAPABILITY OF THE TOP OF FORM OPTION. A SET OF INSTRUCTIONS IS PRINTED FOR THE OPERATOR TO REMIND HIM TO DEPRESS THE TOP OF FORM RESET SWITCH AFTER MAKING EACH SWITCH SETTING. UPON COMPLETION OF EACH SETTING, AFTER DEPRESSING THE RESET SWITCH, TYPE ANY CHARACTER (EXCEPT RUBOUT) ON THE KEYBOARD TO TEST THAT SWITCH SETTING. THE REFERENCE LINES PRINTED WILL INDICATE THE LENGTH FORM FEED JUST EXECUTED AND THE NEXT SWITCH SETTING TO MAKE. THE 3 INCH FORM FEED IS TESTED TWICE BEFORE TESTING THE REMAINING POSITIONS. THE FIRST TIME, 16 OR 17 LINE FEEDS ARE EXECUTED BEFORE DOING THE FORM FEED, DEPENDING ON HOW THE AUTO LINE FEED OPTION IS SET UP. THE DIAGNOSTIC WILL THEN TEST EACH POSITION IN SEQUENCE FROM 3 TO 14 INCHES. THE SINGLE STEP POSITION IS NOT CHECKED.

6.3.5 TEST 34 - HORIZONTAL TAB OPTION

THIS TEST CHECKS THE ABILITY TO SET A TAB IN EVERY COLUMN AND AT PREDETERMINED INTERVALS, AS WELL AS THE ABILITY TO CLEAR ALL TABS. THE PROGRAM SETS A TAB IN THE PREDETERMINED COLUMN, DOES A BACKSPACE, AND PRINTS AN "O". AFTER THE LINE IS PRINTED AND THE TABS ARE SET, A CARRIAGE RETURN IS SENT AND THEN THE PRINT HEAD IS POSITIONED USING TABS AND X'S ARE PRINTED OVER THE O'S. SINCE THE FIRST LINE OF THE TEST SETS A TAB IN EVERY COLUMN, THE PRINT HEAD IS TABED ACROSS THE PAGE TWICE TO TEST ALL TABS. THE FIRST PASS CHECKS THE EVEN NUMBERED COLUMNS WHILE THE SECOND PASS CHECKS THE ODD NUMBERED COLUMNS. THE TEST SETS TABS IN EVERY COLUMN, EVERY OTHER COLUMN, AND EVERY 4, 8, 16, 32, 64, 128, & 132 COLUMNS. ALL HORIZONTAL TABS WILL BE CLEARED AT THE END OF THE TEST IF THE TEST IS RUN TO COMPLETION. IF A RUBOUT IS USED TO EXIT THE TEST BEFORE COMPLETION, THE TABS WILL STILL BE SET.

EXAMPLE:

```

0000000000
 0 0 0 0 0
   0   0
    0

```

WHEN THE AUTO LINE FEED OPTION IS SET UP TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BEFORE EACH REFERENCE LINE OF O'S AND THE X'S WILL BE PRINTED ON THE NEXT LINE UNDER THE O'S. THE FIRST LINE OF O'S WILL HAVE 2 LINES OF X'S UNDER IT, THE FIRST HAVING X'S IN ALL EVEN NUMBERED COLUMNS AND THE SECOND HAVING X'S IN ALL ODD NUMBERED COLUMNS.

EXAMPLE:

```
000000000000  
X X X X X X  
X X X X X X  
  
0 0 0 0 0 0  
X X X X X X  
  
0 0 0  
X X X  
  
0  
X
```

6.3.6 TEST 35 - VERTICAL TAB OPTION

THIS TEST CHECKS THE VERTICAL TAB OPTION BY TESTING THE ABILITY TO SET TABS IN VARIOUS POSITIONS OF A 14 INCH FORM. AN INSTRUCTION IS PRINTED TELLING THE OPERATOR TO SET A 14 INCH FORM LENGTH AND DEPRESS THE TOP OF FORM RESET SWITCH. WHEN READY, TYPE ANY CHARACTER (EXCEPT RUBOUT) ON THE KEYBOARD TO CONTINUE. THE TEST WILL SEND LINE FEEDS, SET TABS, AND PRINT REFERENCE LINES WHEREVER A TAB IS SET. AT THE END OF THE FORM, A MESSAGE WILL INDICATE TO EITHER REMOVE THE REFERENCE PAGE (WITHOUT TOUCHING THE KEYBOARD) OR RESET THE FIRST REFERENCE LINE. TO RESET THE REFERENCE PAGE IN THE PRINTER, OPEN THE PAPER TRACTORS AND PLACE THE FIRST REFERENCE LINE IN FRONT OF THE PRINT HEAD. WHEN READY TO CONTINUE, TYPE ANY CHARACTER (EXCEPT RUBOUT) ON THE KEYBOARD. THE TEST WILL THEN REPRINT THE REFERENCE LINES, USING THE TABS INSTEAD OF LINE FEEDS TO ADVANCE THE PAPER. IF THE FIRST REFERENCE PAGE WAS REMOVED, HOLD IT AGAINST THE SECOND REFERENCE PAGE TO CHECK FOR PROPER PAPER ADVANCING USING TABS. IF THE REFERENCE PAGE WAS RESET IN THE PRINTER, THE SECOND SET OF REFERENCE LINES SHOULD HAVE PRINTED DIRECTLY OVER THE FIRST SET EXCEPT ON THE FIRST LINE WHERE THEY SHOULD BE SIDE-BY-SIDE. ALLOW FOR A SLIGHT VARIANCE IN PAPER POSITION WHEN CHECKING THAT THE REFERENCE LINES ARE CORRECT. LOOK FOR FULL LINE DIFFERENCES. THE TEST PRODUCES 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 BLANK LINES BETWEEN THE REFERNECES LINES. IN THAT ORDER.

00000001
00000002
00000004
00000010
00000020
00000040
00000100
00000200
00000400
00001000
00002000
00004000
00010000
00020000
00040000
00100000
00200000
00400000
10000000
00000000
04000000
00400000
005726
022626
000340
000200
000200
104000
104001
104002
104003
104004
104005
104006
104007
104008
104009
104010
104011
104012
104013
104014
104015
104016
104017
104018
104019
104020
104021
104022
104023
104024
104025

SYSTEM EQUATES

BIT0=1
BIT1=2
BIT2=4
BIT3=10
BIT4=20
BIT5=40
BIT6=100
BIT7=200
BIT8=400
BIT9=1000
BIT10=2000
BIT11=4000
BIT12=10000
BIT13=20000
BIT14=40000
BIT15=100000
OPEN=0
SCOPSW=BIT14
NITRSW=BIT11
POPSP=5726
POPSP2=22626
PRTY7=340
PRTY4=200
ACRLF=200

: SCOPE SWITCH
: TEST LOOP SWITCH
: POP STACK ONCE
: POP STACK TWICE
: PRIORITY LEVEL DEFINITIONS

PROGRAM TRAP EQUATES

TYPE=EMT+0
ERROR=EMT+1
EHALT=EMT+2
STRDRV=EMT+3
STPCHV=EMT+4
CHAIN=EMT+5
CHALT=EMT+6
TYPEM=EMT+7
DELAY=EMT+10
TTYCTL=EMT+11
CRLF=EMT+12
SCRLF=EMT+13
LF=EMT+14
PRINTC=EMT+15
PRTHDR=EMT+16
PRNT=EMT+17
READ=EMT+20
AREAD=EMT+21
CR=EMT+22
BTOASC=EMT+23
FORWD=EMT+24
READC=EMT+25

112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159

```

.SBTTL TRAP CATCHER & STARTING ADDRESSES
:
.ENABL ABS,AMA

      000000      . = 0
000000 000002      . + 2      ; UNASSIGNED TRAP
000002 000000      HALT
000004 000006      MACHER: . + 2      ; SP OVERFLOW, BUS ERROR TRAP
000006 000000      HALT
000010 000012      . + 2      ; RESERVED INSTRUCTION TRAP
000012 000000      HALT
000014 000016      . + 2      ; TRACE TRAP
000016 000000      HALT
000020 000022      . + 2      ; TRAP TO CALL IOX
000022 000000      HALT
000024 000026      . + 2      ; POWER FAIL TRAP
000026 000000      HALT
000030 003176      EMTINT      ; EMT TRAP
000032 000340      PRTY7

      000042      . = 42
000042 000000      0
      000046      . = 46
000046 011426      LOGICAL
      000052      . = 52
000052 010000      010000
      000174      . = 174
000174 000000      DISPREG: .WORD 0      ; SOFTWARE DISPLAY
000176 000000      SWREG:   .WORD 0      ; SOFTWARE SWITCH REGISTER

000200 000137 001304      JMP START      ; START UP WITH I/O TEST RUNNING
000204 000137 001232      JMP START1     ; START UP, SKIP ALL I/O TEST
000210 000137 001250      JMP START2     ; START UP TERMINAL CONTROL WITH I/O TEST
000214 000137 001266      JMP START3     ; START UP TERM CNTL WITHOUT I/O TESTS

      001100      . = 1100
001100 000000      SPBOT: 0      ; BOTTOM OF STACK

```

160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203

001102 177560
001104 000060
001106 176500

001110 177560
001112 177562
001114 177564
001116 177566
001120 000060
001122 000200
001124 000364
001126 000200
001130 000000
001132 000000
001134 000000
001136 000000
001140 000000
001142 000000
001144 000000
001146 000000
001150 000000
001152 000000
001154 000000
001156 000000
001160 000000
001162 000000
001164 000000
001166 000000
001170 000251
001172 000000
001174 000000
001176 000000
001200 000000
001202 000000
001204 000000
001206 000000
001210 000000
001212 177570

.SBTTL SYMBOL DEFINITIONS

CONADD: 177560
CONVEC: 60
CLADR: 176500

TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
TKVTR: 60
TKLVL: PRTY4
TPVTR: 64
TPLVL: PRTY4
FSTD: OPEN
CNTLSW: OPEN
RTNNG: OPEN
NXTST: OPEN
SCOPTA: OPEN
PRGID: OPEN
CRBUF: OPEN
CTRA: OPEN
WIDTH: OPEN
LEVEL: OPEN
CLCNT: OPEN
ICTR: OPEN
REPT: OPEN
BRCTR: OPEN
COUNT3: OPEN
XCSR: OPEN
TIMER: 251
SPCNT: OPEN
CURTST: OPEN
TEMPCH: OPEN
PARITY: OPEN
PCHAR: OPEN
LFCNT: OPEN
INCHK: OPEN
TEMP: OPEN
SR: 177570

: ADDR OF CONSOLE RECEIVER STATUS REG
: CONSOLE TERMINAL INTERRUPT VECTOR
: ADDRESS OF FIRST DL11, DEFAULT TO DL11-A,B
: SET TO 175610 FOR FIRST 16 (OF 31) OR
: SET TO 176000 FOR LAST 16 (OF 31)
: DL11-C,D,E
: CONSOLE RECEIVER STATUS REG
: CONSOLE RECEIVER BUFFER
: CONSOLE TRANSMITTER STATUS REG
: CONSOLE TRANSMITTER BUFFER
: C.T. RECEIVER INTERRUPT VECTOR
: C.T. RECEIVER PRIORITY LEVEL
: C.T. TRANSMITTER INTERRUPT VECTOR
: C.T. TRANSMITTER PRIORITY LEVEL
: ADDRESS OF FIRST ACTIVE DL11
: CONSOLE TERMINAL CONTROL SWITCH
: CONTAINS CURRENT TEST NUMBER
: CONTAINS ADDRESS OF NEXT TEST
: CONTAINS ADDRESS OF TEST SCOPE ENTRY
: CONTAINS TEST PROGRAM INDICATORS

: CURRENT PAPER WIDTH, BINARY
: LEVEL OF EXECUTION
: # OF MULTIPLE DL11S
: I/O TEST ITERATION COUNT
: TEMP STORAGE FOR TESTS E02!3E022
: COUNTER FOR ROUTINE "AREAD"
: COUNTER FOR ROUTINE "PRINTC"
: ADDRESS OF MULTIPLE DL11 STATUS
: 1 MSEC COUNTER FOR ROUTINE "DELAY"
: COUNTER FOR TEST ROUTINE "PT3"
: ADDRESS OF CURRENT TEST
: TEMP STOR FOR ECHO TESTS
: PARITY FLAG FOR RECEIVED CHAR
: CHAR CODE WITH PARITY BIT
: COUNTER FOR TEST ROUTINE "PT4"
: CHECK FOR INPUT FLAG
: TEMPORARY WORKING STORAGE
: SW REG ADDRESS

.SBTTL PROGRAM INITIALIZATION & CONTROL

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245

```

*****
;COMMON HALT---WHEN IN SWITCH REGISTER CONTROL THE CPU
;WILL BE ADVANCED TO THIS COMMON HALT WHERE
;A NEW TEST WILL BE EXPECTED TO BE STARTED
*****
001214 005737 001152  CHLT:  TST      LEVEL      ;TEST CURRENT LEVEL
001220 001403          BEQ      SELHLT    ;BRANCH IF 0, DO NOT HALT
001222 011600          MOV      QSP,RO   ;PUT ADDRESS OF CALLER INTO RO
001224 005740          TST      -(RO)
001226 000000          HALT
001230 000002  SELHLT: RTI      ;RETURN FROM INTERRUPT
001232 012737 177777 005522 START1: MOV      #177777,ATOX ;FORCE END OF I/O TESTS
001240 012737 104006 002154      MOV      #CHALT,WAITF ;FORCE SR CONTROL
001246 000424          BR       STARTX
001250 012737 104011 002154 START2: MOV      #TTYCTL,WAITF ;FORCE TERMINAL CONTROL
001256 012737 005552 005522      MOV      #AT1,ATOX   ;FORCE ALL I/O TESTS
001264 000415          BR       STARTX
001266 012737 104011 002154 START3: MOV      #TTYCTL,WAITF ;FOECE TERMINAL CONTROL
001274 012737 177777 005522      MOV      #177777,ATOX ;FORCE END OF I/O TESTS
001302 000406          BR       STARTX
001304 012737 005552 005522 START:  MOV      #AT1,ATOX   ;FORCE ALL I/O/TESTS
001312 012737 104006 002154      MOV      #CHALT,WAITF ;FORCE SR CONTROL
001320 012706 001100      STARTX: MOV      #SPBOT,SP ;SET STACK POINTER
001324 013746 000006      MOV      6,-(SP)   ;SAVE CURRENT VECTOR
001330 013746 000004      MOV      4,-(SP)
001334 012737 001350 000004      MOV      #10$,4   ;SET UP TIMEOUT VECTOR
001342 005777 177644      TST      QSR      ;TRY TO REFERENCE HARDWARE SW REG
001346 000404          BR       11$      ;BRANCH IF NO TIMEOUT TRAP OCCURS
001350 012737 000176 001212 10$:  MOV      #SWREG,SR  ;POINT TO SOFTWARE SWITCH REGISTER
001356 022626          CMP      (SP)+,(SP)+ ;RESTORE STACK
001360 012637 000004      11$:  MOV      (SP)+,4   ;RESTORE TIMEOUT VECTOR
001364 012637 000006      MOV      (SP)+,6
001370 005037 001206          CLR      INCHK     ;ALLOW INPUT CHECKING
001374 012737 000006 000004      MOV      #6,MACHER ;CLEAN UP
001402 005037 001142          CLR      PRGID    ;INITIALIZE PROGRAM FLAGS
001406 005037 001132          CLR      CNTLSW   ;INITIALIZE TERMINAL CONTROL SWITCH
001412 005037 001152          CLR      LEVEL    ;INITIALIZE LEVEL
001416 012737 003672 000024      MOV      #PFAIL,24 ;SET ADDR POWER FAIL ROUTINE
001424 004737 004122          JSR      PC,CONIT ;SET UP CONSOLE TERMINAL ADDRESS

```

```

246
247
248
249
250
251
252
253
254
255 001430 017701 177556
256 001434 042701 177400
257 001440 020127 000204
258 001444 003003
259 001446 020127 000035
260 001452 101002
261 001454 012701 000204
262 001460 010137 001150
263 001464 012700 016715
264 001470 012702 000003
265 001474 104023
266 001476 000401
267 001500 000405
268 001502 104007
269 001504 016505
270 001506 012737 000240 001476

```

```

:*****
:READ THE PAPER WIDTH, NUMBER OF COLUMNS,
:FROM SWITCH REGISTER POSITIONS 0-7. SAVE AND
:CONVERT TO 3 ASCII CHARACTERS. A WIDTH GT132
:OR LT30 COLUMNS (DECIMAL) WILL BE ASORTED TO 132.
:THE SWITCHES MAY BE CHANGED ONCE THE PROGRAM TITLE OR THE CL11 COUNT
:MESSAGE HAS STARTED TO PRINT.
:*****

```

```

MOV JSR,R1 ;PUT (SR) INT R1
BIC #177400,R1 ;SAVE ONLY BITS 0-7
CMP R1,#204 ;TEST NO. COLUMN GT132
SGT 2$ ;COLUMNS GT132, DEFAULT TO 132
1$: CMP R1,#35 ;CHECK IF NO. COLUMNS LT 30
BHI 3$ ;NOT LT 30 NOR GT 132
2$: MOV #204,R1 ;COLUMNS LT 30 OR GT 132, DEFAULT
3$: MOV R1,WIDTH ;SAVE NO. COLUMNS IN WIDTH
MOV #HORO,R0 ;ADDR TO STORE ASCII COLUMN VALUE
MOV #3,R2 ;DO A 3 CHAR. CONVERSION
BTOASC ;CONVERT NO. COLUMNS TO ASCII
4$: BR 5$
BR 6$
5$: TYPEN ;TYPE PROGRAM TITLE FIRST TIME RUN
STARTM
MOV #NOP,4$

```

```

271 ;*****
272 ;THIS NEXT PART CHECKS THE PRESENCE OF DL11-A OR DL11-C
273 ;STARTING AT 776500. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER
274 ;PRESENT. THE PRINTER DIAGNOSTIC WILL ADDRESS EACH OF
275 ;THE MULTIPLE DL11S IN THE SYSTEM IF SWITCH 13 IS DOWN (0)
276 ;*****
277
278 001514 012737 001574 000004 6$: MOV #END2A,MACHER ;INITIALIZE TIME OUT TRAP
279 001522 013700 001106 MOV DLADR,RO ;ADDRESS OF FIRST DL11 TO RO
280 001526 012701 000020 MOV #20,R1 ;SET DL CHECK COUNT
281 001532 005037 001154 CLR DLCNT ;INITIALIZE DLCNT
282 001536 005710 END3: TST (RO) ;IS DL PRESENT?
283 001540 012737 001606 000004 MOV #END2,MACHER ;YES, RESET TIME OUT TRAP
284 001546 010037 001130 MOV RO,FSTDL ;STORE ADDRESS OF FIRST DL11
285 001552 000401 BR 2$ ;CONTINUE
286 001554 005710 1$: TST (RO) ;IS DL11 PRESENT
287 001556 062700 000010 2$: ADD #10,RO ;POINTER AND DL11 ADDRESS
288 001562 005237 001154 INC DLCNT ;INCREMENT COUNT OF DL11'S
289 001566 005301 DEC R1 ;DECREMENT DL CHEC COUNT, DONE?
290 001570 001407 BEQ END4 ;BRANCH IF DONE
291 001572 000770 BR 1$ ;CHECK PRESENCE OF NEXT DL11
292 001574 005301 END2A: DEC R1 ;DONE DL CHECK?
293 001576 001404 BEQ END4 ;YES, EXIT
294 001600 062700 000010 ADD #10,RO ;NO, CHECK NEXT DL
295 001604 000754 BR END3 ;CONTINUE
296 001606 022626 END2: POPSP2 ;DL11 NOT PRESENT
297 001610 013701 001154 END4: MOV DLCNT,R1 ;GET # DL11'S
298 001614 012700 016654 MOV #DL11$1,RO ;ADR OF ASCII CHAR STORAGE
299 001620 012702 000002 MOV #2,R2 ;# OF ASCII CHARS
300 001624 104023 BTOASC ;CONVERT NUMBER
301 001626 104007 TYPEN ;TYPE MESSAGE
302 001630 011641 DL11S
303
304 ;*****
305 ;EXECUTE THE STRING OF CONSOLE TERMINAL I/O TESTS
306 ;THEN EITHER HALT AT LOCATION 1022 OR CONTINUE WITH
307 ;PRINTER TESTS AS A FLNCTION OF SR BIT 8.
308 ;*****
309
310 001632 005037 001134 CLR RTNNO ;SET ROUTINE NO = 0
311 001636 005037 001152 CLR LEVEL ;ST LEVEL = 0
312 001642 023727 005522 177777 CMP ATOX,#177777 ;SEE IF I/O IS TO BE SKIPED
313 001650 001515 BEQ SKIP
314 001652 012737 005520 001136 MOV #ATO,NXTST ;ADDRESS OF FIRST I/O TEST
315 001660 104024 FORWD ;SET UP TEST PARAMETERS
316 001662 000177 177306 JMP @CURTST ;GO TO I/O TEST ROUTINE

```


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

;CHAINN-- THIS PORTION IS THE COMMON RETURN
FOR ALL THREE CLASSES OF TESTS.
1--IF AN ERROR OCCURED DURING AN I/O TEST THE
OPERATOR CAN CAUSE THAT TEST TO BE LOOPED
WITHOUT ANY FURTHER ERROR HALTS BY
SETTING THE "SCOPE" BIT (#14) ON THE SR=1.
RESETTING SR BIT 14 TO 0 WILL ALLOW THE
ERROR HALT TO OCCUR AGAIN IF IT STILL EXISTS.
2--IF THE OPERATOR IS IN THE MAINTENANCE
MODE (BIT 8 SET = 1 AT START UP TIME), THE
SELECTED PROGRAM WILL LOOP CONTINUOUSLY
IF SR BIT 11 IS SET=1. IF BIT 11 IS = 0
THEN THE PROGRAM WILL BE ADVANCED TO
THE NEXT TEST IN IT'S CLASS IF BIT 8=0.
AS LONG AS BIT 11 AND
BIT 8 ARE 0, THE CLASS OF TESTS SELECTED
WILL BE CONTINUOUSLY SEQUENCED THROUGH.
IF BIT 11 IS 0 AND BIT 8=1, THEN THE CPU
WILL HALT AT LOCATION 1022 AND WAIT FOR THE
NEXT TEST NUMBER TO BE SET IN THE
SWITCH REGISTER.

343 001666 032737 000001 001132 CHAINN: BIT #1,CNTLSW ;CCKECK IF TERMINAL CONTROL
344 001674 001401 BEQ 1\$;BRANCH IF NOT
345 001676 104011 TTYCTL ;GO TO TERMINAL CONTROL
346 001700 005737 001142 1\$: TST PRGID ;TEST ERROR BIT IN PRGID
347 001704 100016 BPL 3\$;BRANCH IF ERROR BIT NOT SET
348 001706 032777 040003 177276 BIT #SCOPSW,2SR ;ERR, CHECK IF SCOPE OPTION ON
349 001714 001407 BEQ 2\$;BRANCH IF NO SCOPING
350 001716 022737 177777 001140 CMP #-1,SCOPTR ;YES, CHECK IF OK TO SCOPE THIS TEST
351 001724 001403 BEQ 2\$;BRANCH IF NOT OK
352 001726 017716 177206 MOV 2\$SCOPTR,2SP ;PUT ADDR OF SCOPE ENTRY INTO STACK
353 001732 000002 RTI ;GO TO SCOPE ENTRY IN TEST
354 001734 042737 100000 001142 2\$: BIC #BIT15,PRGID ;CLEAR ERROR IND. IN PRGID
355 001742 005737 001152 3\$: TST LEVEL ;CHECK LEVEL
356 001746 001405 BEQ 4\$;BRANCH IF LEVEL=0
357 001750 032777 004000 177234 BIT #NITRSW,2SR ;TEST LOOP SWITCH ON (=1)
358 001756 001405 BEQ 5\$;BRANCH IF NO LOOP TEST
359 001760 000002 RTI ;GO BACK TO TEST
360 001762 005337 001156 4\$: DEC ICTR ;DECREMENT TEST ITERATION COUNT
361 001766 001407 BEQ 6\$;BRANCH IF COUNT=0
362 001770 000002 RTI ;NOT ZERO, REPEAT TEST
363 001772 032777 000400 177212 5\$: BIT #BIT8,2SR ;TEST IF SEQUENCE TEST (BIT9)
364 002000 001402 BEQ 6\$;BRANCH TO NEXT TEST IF BIT9=0
365 002002 000137 002154 JMP WAITF ;GO WAIT FOR MORE INPUT
366 002006 022626 6\$: POPSP2 ;POP 2 OFF STACK
367 002010 000005 CHAINY: RESET ;RESET
368 002012 005777 177174 TST 2SR ;CHECK SR
369 002016 100003 BPL 1\$;BRANCH IF NO HALT WANTED
370 002020 113700 001134 MOVB RTNNO,RO ;CURRENT TEST NUMBER TO RO

```

371 002024 000000          HALT          ;HALT
372 002026 005737 001152 1$:  TST          LEVEL          ;TEST THE CURRENT LEVEL
373 002032 001420          BEQ          3$          ;BRANCH IF 0
374 002034 012737 000006 000004  MOV          #6,MACHER      ;CLEAN UP
375 002042 012706 001100          MOV          #SPBOT,SP     ;SET UP STACK POINTER
376 002046 104024          FORWD         ;SET UP VALUES FOR NEXT TEST
377 002050 022737 177777 001136  CMP          #-1,NXTST     ;END OF I/O TESTS (=-1)
378 002056 001004          BNE          2$          ;BRANCH IF NOT END
379 002060 012737 005520 001136  MOV          #ATO,NXTST    ;RESET NXTST TO FIRST I/O TEST
380 002066 104024          FORWD         ;SET UP VALUES FOR NEXT TEST
381 002070 000177 177100          JMP          @CURTST       ;GO TO TEST
382 002074 022737 177777 001136 3$:  CMP          #-1,NXTST     ;END OF I/O TESTS (=-1)
383 002102 001012          BNE          NEXT        ;BRANCH IF NOT
384 002104 032777 000400 177100  SKIP:  BIT          #BIT8,@SR ;TEST IF WANT TEST SELECTION RIGHT AWAY
385 002112 001016          BNE          NEXT1       ;BRANCH IF NOT
386 002114 052737 000200 001142  BIS          #BIT7,PRGID   ;BYPASS SCOPING
387 002122 012737 007242 001136  MOV          #PTO,NXTST    ;PROD TESTING, GO TO PRINTER TESTS
388 002130 012737 000006 000004  NEXT:  MOV          #6,MACHER ;CLEAN UP
389 002136 012706 001100          MOV          #SPBOT,SP     ;SET UP STACK POINTER
390 002142 104024          FORWD         ;SET UP NEXT TEST PARAMETERS
391 002144 000177 177024          JMP          @CURTST       ;GO TO ROUTINE
392 002150 005237 001152  NEXT1:  INC          LEVEL
393
394
395
396
397
398
399
400
401
402
403
404 002154 104006          WAITF:  CHALT         ;OR TTYCTL IF START WAS AT 206
405 002156 012737 000006 000004  MOV          #6,MACHER      ;CLEAN UP
406 002164 012706 001100          MOV          #SPBOT,SP     ;SET UP STACK POINTER
407 002170 017700 177016          MOV          @SR,RO        ;GET CURRENT SW REG
408 002174 042700 177700          BIC          #177700,RO    ;
409 002200 020027 000037          CMP          RO,#37        ;TEST IF PROG NO. IS I/O TEST
410 002204 101403          BLOS        1$          ;BRANCH IF EQ OR LT 37. AN ECHO OR PRINTER
411 002206 005037 001142          CLR          PRGID        ;I/O TEST, CLEAR PRGID
412 002212 000403          BR          2$          ;
413 002214 052737 000200 001142 1$:  BIS          #BIT7,PRGID   ;BYPASS SCOPING
414 002222 000241          CLC          ;CLEAR C BIT
415 002224 006100          ROL          RO          ;GET PROGRAM ADDRESS OUT OF
416 002226 016037 002776 001136  MOV          PRGTAB(RO),NXTST ;PROGRAM ADDRESS TABLE
417 002234 023727 001136 002154  CMP          NXTST,#WAITF  ;TEST IF LEGAL TEST NO.
418 002242 001744          BEQ          WAITF        ;BRANCH IF ILLEGAL
419 002244 104024          FORWD         ;SET UP TEST PARAMETERS
420 002246 000177 176722          JMP          @CURTST       ;GO TO TEST

```

```

*****
;WAIT FOR FURTHER INSTRUCTIONS:
;
;-LOAD PROGRAM NUMBER INTO BITS 0-5 OF THE SR
;-SET SR BIT 11=1 TO LOOP ON SELECTED TEST
;-SET SR BIT 11=0 AND BIT 8=0 TO LOOP THROUGH
; SEQUENCE OF SELECTED TESTS.
;-SET SR BIT 11=0 AND BIT 8=1 TO HALT AGAIN AFTER
; EXECUTING TEST ONCE
*****

```

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

002252 022626
002254 105777 176630
002260 100013
002262 017705 176624
002266 042705 177600
002272 020527 000177
002276 001004
002300 042737 004400 001132
002306 000413
002310 032737 004000 001132 1\$:
002316 001401
002320 000002
002322 032737 000400 001132 2\$:
002330 001402
002332 000137 002010
002336 012737 177777 001206 TTY1B:
002344 012700 000036
002350 104010
002352 104007
002354 017127
002356 005037 001206
002362 104020
002364 023727 001176 000040 1\$:
002372 001773
002374 012700 000036
002400 104010
002402 104017
002404 117777 176502 176504
002412 004737 002734
002416 000541
002420 010005
002422 006305
002424 006305
002426 006305
002430 104020
002432 023727 001176 000040 2\$:
002440 001773
002442 012700 000036
002446 104010
002450 104017
002452 117777 176434 176436

TTY1-- THIS SECTION IS USED WHEN THE DIAGNOSTIC IS UNDER CONTROL
THE CONSOLE TERMINAL. IT IS EFFECTIVE ONLY WHEN THE DIAGNOSTIC
STARTING ADDRESS IS 210 AND SR BIT 8 WAS SET AT START TIME.
THE RESPONSE TO THE MESSAGE "SELECT TEST NO." MUST BE THE 2
DIGIT OCTAL TEST NUMBER FOLLOWED BY ;
"L" TO LOOP ON TEST
"S" TO LOOP ON SEQUENCE
"." TO EXECUTE TEST ONCE
ALL SPACES WILL BE IGNORED. AN ILLEGAL INPUT WILL BE FLAGED BY A "?"
AND THE RETYPING OF THE ABOVE MESSAGE.

TTY1: POPSP2 ;POP 2 FROM STACK
TSTB @TKS ;TEST IF ANY INPUT
BPL 1\$;BRANCH IF NOT
MOV @TKB,R5 ;GET CHAR
BIC #177600,R5 ;MASK BITS
CMP R5,#177 ;CHECK IF RUBOUT
BNE 1\$;BRANCH IF NOT
BIC #4400,CNTLSW ;CLEAR LOOP BITS
BR TTY1B
BIT #NITRSW,CNTLSW ;CHECK IF LOOP ON TEST
BEQ 2\$;BRANCH IF N' LOOP ON TEST
RTI ;LOOP ON TEST
BIT #BIT8,CNTLSW ;TEST IF LOOP ON SEQUENCE
BEQ TTY1B ;BRANCH IF NO LOOP ON SEQUENCE
JMP CHAINY ;CHAIN TO NEXT TEST
TTY1B: MOV #-1,INCHK ;STOP INPUT CHECKING
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
TYPEM
MSG3 ;TYPE MESSAGE
CLR INCHK ;ALLOW INPUT CHECKING AGAIN
1\$: READ ;WAIT FOR INPUT
CMP TEMPCH,#40 ;TEST IF CHAR IS A SPACE
BEQ 1\$;BRANCH IF YES
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
PRNT ;READY?
MOVB @TKB,@TPB ;ECHO CHAR
JSR PC,TESTC ;CHECK IF CHAR IS OK
BR 8\$;NO, ERROR
MOV R0,R5 ;OK, PUT CHAR INTO R5
ASL R5 ;SHIFT INTO POSITION 5-3
ASL R5
2\$: READ ;WAIT FOR NEXT CHAR
CMP TEMPCH,#40 ;CHECK IF A SPACE
BEQ 2\$;BRANCH IF SPACE
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
PRNT ;READY?
MOVB @TKB,@TPB ;ECHO CHAR

475	002460	004737	002734			JSR	PC,TESTC		;CHECK IF CHAR IS OK
476	002464	000516				BR	9\$;ERROR IN CHAR
477	002466	060005				ADD	RO,R5		;OK,R5 NOW = OCTAL TEST NO.
478	002470	104020			3\$:	READ			;WAIT FOR TERMINATION CHARACTER
479	002472	023727	001176	000040		CMP	TEMPCH,#40		;CHECK IF SPACE
480	002500	001773				BEQ	3\$;BRANCH IF SPACE
481	002502	012700	000036			MOV	#30.,RO		;DELAY FOR HALF DUPLEX.
482	002506	104010				DELAY			
483	002510	104017				PRNT			;READY?
484	002512	117777	176374	176376		MOVB	@TKB,@TPB		;ECHO CHAR
485	002520	012737	004001	001132		MOV	#4001,CNTLSW		;SET BITS 11 & 0
486	002526	023727	001176	000114		CMP	TEMPCH,#114		;NO, IS IT AN "L" ?
487	002534	001427				SEQ	5\$;BRANCH IF YES
488	002536	023727	001176	000154		CMP	TEMPCH,#154		;CHECK LOWER CASE
489	002544	001423				BEQ	5\$		
490	002546	023727	001176	000123		CMP	TEMPCH,#123		;NO, IS IT AN "S"
491	002554	001414				BEQ	4\$;BRANCH IF YES
492	002556	023727	001176	000163		CMP	TEMPCH,#163		;CHECK LOWER CASE
493	002564	001410				BEQ	4\$		
494	002566	023727	001176	000056		CMP	TEMPCH,#56		;NO, IS IT A "." ?
495	002574	001052				BNE	8\$;NO, ERROR
496	002576	012737	000001	001132		MOV	#1,CNTLSW		;YES SET ONLY BIT 0 IN CONTROL WD
497	002604	000403				BR	5\$		
498	002606	012737	000401	001132	4\$:	MOV	#401,CNTLSW		;SET BITS 8 & 0
499	002614	012737	000006	000004	5\$:	MOV	#6,MACHER		;CLEAN UP
500	002622	012706	001100			MOV	#SPBOT,SP		;INIT SP
501	002626	020527	000040			CMP	R5,#40		;IS THIS AN I/O TEST
502	002632	103033				BHIS	8\$;BRANCH IF YES
503	002634	020527	000030			CMP	R5,#30		;IS THIS AN OPTION TEST?
504	002640	103007				BHIS	6\$;SKIP IF YES
505	002642	020527	000020			CMP	R5,#20		;IS THIS AN ECHO TEST
506	002646	103404				BLO	6\$;BRANCH IF NOT
507	002650	012737	000001	001132		MOV	#1,CNTLSW		;FORCE ECHO TEST TO A SINGLE RUN
508	002656	000402				BR	7\$;LEAVE THIS TERMINAL AS CONSOLE
509	002660	004737	004122		6\$:	JSR	PC,CONIT		;RESET CONSOLE TERMINAL ADDRESS
510	002664	052737	000200	001142	7\$:	BIS	#BIT7,PRGID		;BYPASS SCOPING
511	002672	000241				CLC			;CLEAR C BIT
512	002674	006105				ROL	R5		
513	002676	016537	002776	001136		MOV	PRGTAB(R5),NXTST		;ADDR OF TEST TO NXTST
514	002704	023727	001136	002154		CMP	NXTST,#WAITF		;CHECK IF TEST EXISTS
515	002712	001403				BEQ	8\$;BRANCH IF NOT
516	002714	104024				FORWD			;SET UP TEST PARAMETERS
517	002716	000177	176252			JMP	@CURTST		;GO TO TEST
518	002722	104017			8\$:	PRNT			;CHECK IF PRINTER IS READY
519	002724	112777	000077	176164		MOVB	#77,@TPB		;SEND A "?"
520	002732	000601				BR	TTYIB		;TRY AGAIN

003153
003154
003156
003160
003162
003164
003166
003170
003172
003174

WAITF
WAITF
WAITF
WAITF
WAITF
WAITF
WAITF
WAITF
WAITF
WAITF

: SPARE
: SPARE
: SPARE
: SPARE
: SPARE
: SPARE
: SPARE
: SPARE
: SPARE
: SPARE

:EMTINT AND TRPINT--SERVICE ROUTINES FOR TRAPS THROUGH
: LOC. 30 AND 34.

003176 011646
003200 162716 000002
003204 017616 000000
003210 121627 000025
003214 101402
003216 000000
003220 000775
003222 006116
003224 042716 177001
003230 062716 003252
003234 017516 000000
003240 005046
003242 012746 003250
003246 000002
003250 000136

EMTINT: MOV 2(SP),-(SP) ;GET SAVED PC
SUB #2,2SP
MOV 2(SP),2SP
CMPB 2SP,#25 ;CHECK THAT CALL IS WITHIN LIMITS
BLOS 25 ;BRANCH IF OK
1\$: HALT
BR 1\$
2\$: ROL 2SP ;EMT ARGUMENT *2
BIC #177001,2SP ;REMOVE 7 MSB
ADD #EMTTAB,2SP ;FORM EMT RTN ADDRESS
MOV 2(SP),2SP
CLR -(SP) ;CLEAR PSW
MOV #35,-(SP)
3\$: JMP 2(SP)+ ;GO TO EMT ROUTINE

003252 003336
003254 003532
003256 003560
003260 003570
003262 003620
003264 001666
003266 001214
003270 003404
003272 003650
003274 002252
003276 003434
003300 003362
003302 003436
003304 004536
003306 003456
003310 004526
003312 004324
003314 004052
003316 003446
003320 004220
003322 003774
003324 004416

EMTTAB: TYP ;MESSAGE OUTPUT ROUTINE
ERR ;I/O TEST ERROR ROUTINE
EHLT ;UNCONDITIONAL HALT
STLSRV ;KEYBOARD VECTOR/PRIORITY SETUP
STLSPV ;PRINTER VECTOR/PRIORITY SETUP
CHAINN ;COMMON TEST EXIT
CHLT ;SR BIT 15 HALT
TYPM ;MESSAGE OUTPUT ROUTINE, MULTI DEVICES
DLY ;DELAY ROUTINE
TTYI ;CONSOLE TERMINAL CONTROL
ICRLF ;CARRIAGE RETURN-LINE FEED TO ALL DLI's
SSCRLF ;CARRIAGE RETURN-LINE FEED TO CONSOLE
SLF ;LINE FEED ONLY (TO ALL)
SPRTC ;PRINT CHAR
SPRHDR ;PRINT TEST HEADER
SPRNT ;PRINTER READY
SREAD ;READ CHAR
SAREAD ;I/O TEST READ ROUTINE
SCR ;CARRIAGE RETURN ONLY (TO ALL)
SBTASC ;BINARY TO ASCII CONVERSION
SFRWD ;FORWARD ROUTINE (BETWEEN TESTS)
SREADC ;READ CONSOLE KYBD ONLY

.SBTTL COMMON ROUTINES USED BY LA36 TESTS

THIS SECTION CONTAINS MOST ROUTINES CALLED BY
THE VARIOUS TESTS EITHER BY TRAPPING THROUGH LOCATIONS
30 AND 34 OR BY SUBROUTINE CALLS (JSR PC,***)

TYPE-- A COMMON ROUTINE USED TO TYPE MESSAGES ON THE
CONSOLE TERMINAL ONLY. THE NULL CHARACTER TERMINATES
THE MESSAGE. CALLED THROUGH AN EMT TRAP.
CALLING SEQUENCE
TYPE
MSG ;ADDRESS OF MESSAGE

TYP: MOV (SP),R1 ;GET POINTER TO ADDR. OF MSG.
ADD #2,R1
MOV (R1),R1 ;ADDR. OF MSG TO R1
1\$: MOVB (R1)+,R0 ;GET CHAR
BMI 2\$;BRANCH IF WANT AUTO CR-LF
BNE 3\$;PRINT CHAR IF NOT NULL
RTI ;EXIT IF NULL CHAR
2\$: SCRLF ;YES, SEND CR-LF
BR 1\$;GET NEXT CHAR
3\$: PRNT ;PRINTER READY?
MOVB R0,RTPB ;LOAD PRINTER BUFFER WITH CHAR
BR 1\$;GO GET NEXT CHAR
SSCRLF: PRNT ;PRINTER READY?
MOVB #15,RTPB ;SEND CR
PRNT ;PRINTER READY?
MOVB #12,RTPB ;SEND LF
RTI ;RETURN TO CALLER

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

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

003404 011601
003406 362716 000002
003412 011101
003414 112100
003416 100402
003420 001003
003422 000002
003424 104012
003426 000772
003430 104015
003432 000770

003434 104022
003436 012700 000012
003442 104015
003444 000002

003446 012700 000015
003452 104015
003454 000002

003456 104007
003460 016703
003462 013700 001134
003466 006200
003470 006200
003472 006200
003474 042700 177770
003500 062700 000060
003504 104015
003506 013700 001134
003512 042700 177770
003516 062700 000060
003522 104015
003524 104012
003526 104014
003530 000002

```
:XXXXXXXXXX
:
: TYPM---MULTI TYPE-A COMMON ROUTINE TO OUTPUT
: A MESSAGE ON ALL DL115 IF THE MULTI TEST
: SWITCH (BIT 13) IS SET. THIS ROUTINE IS USED BY
: THE PRINTER TESTS TO TYPE HEADINGS. IF A UNIT
: IS NOT READY, THE CHARACTER WILL NOT BE TYPED.
:
:XXXXXXXXXX
:
TYPM:  MOV      (SP),R1      ;GET POINTER TO ADDR OF MESSG
      ADD      #2,R1
      MOV      (R1),R1      ;ADDR OF MESSG TO R1
15:    MOVE     (R1)+,R0     ;GET CHAR
      BMI     25            ;BRANCH IF WANT AUTO CR-LF
      BNE     35            ;CONTINUE IF NOT NULL
      RTI
25:    CRLF
      BR      15           ;YES, SEND CR-LF
      BR      35           ;NEXT CHAR
35:    PRINTC
      BR      15           ;PRINT CHAR
      BR      15           ;GO GET NEXT CHAR.

$CRLF: CR
$LF:   MOV      #12,R0     ;SEND CR
      PRINTC
      RTI                ;SET LF CHAR
                        ;SEND IT
                        ;RETURN TO CALLER

$CR:   MOV      #15,R0     ;SET CR CHAR
      PRINTC
      RTI                ;SEND IT
                        ;RETURN

:*****
:ROUTINE TO PRINT TEST HEADER
:*****

$PRHDR: TYPM
      HDRMSG                ;PRINT MESSAGE
      MOV      RTNNO,R0     ;GET TEST NUMBER
      ASR     R0            ;GET FIRST DIGIT
      ASR     R0
      ASR     R0
      BIC     #177770,R0    ;MASK FIRST DIGIT
      ADD     #60,R0        ;MAKE ASCII
      PRINTC                ;PRINT DIGIT
      MOV     RTNNO,R0     ;GET TEST NUMBER AGAIN
      BIC     #177770,R0    ;MASK LAST DIGIT
      ADD     #60,R0        ;MAKE ASCII
      PRINTC                ;PRINT DIGIT
      CRLF                 ;CR-LF
      LF
      RTI                 ;BLANK LINE
                        ;RETURN
```

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

```

:*****
:ERRA-- COMMON ERROR RETURN FROM I/O TESTS. HALTS
: WITH ADDRESS OF ERROR IN RD. TO CONTINUE
: ON SAME TEST BUT NOT HALTING ON ERROR.
: SET THE SCOPE BIT (14) = 1 AND PRESS CONTINUE
:*****
003532 032777 040C00 17545E ERR: BIT #SCOPSW,ASR ;CHECK SCOPE SWITCH
003540 001404 BEQ 15 ;BRANCH IF NO SCOPE
003542 005737 001142 TST PRGID ;SCOPING WANTED, FIRST ERROR?
003546 100001 SPL 15 ;BRANCH AND HALT ON FIRST ERROR
003550 000002 RTI ;SCOPE EXIT
003552 052737 100000 001142 15: BIS #BIT15,PRGID ;SET ERROR INDICATOR
003560 011600 EHLT: MOV JSP,RO ;ADDRESS OF CALL INTO RD
003562 005740 TST -(RO)
003564 000000 HALT
003566 000002 ERRHLT: RTI ;RETURN TO TEST FOLLOWING CALL
:*****
:STLSRV--- THIS ROUTINE SETS UP KEYBOARD INTERRUPT
: VECTOR AND PRIORITY. CALLING SEQUENCE
:
: STDRV
: AT20C ;LOCATION OF NEW INTERRUPT VECTOR
:*****
003570 017637 000000 003610 STLSRV: MOV 2(SP),STPRA+2 ;SET RETURN ADR AND VECTOR
003576 062716 000002 ADD #2,JSP
003602 013701 001120 MOV TKVTR,R1
003606 012721 000000 STPRA: MOV #0,(R1)+
003612 013721 001122 MOV TKLVL,(R1)+
003616 000002 RTI
:*****
:STLSPV-- THIS ROUTINE SETS UP PRINTER INTERRUPT
: VECTOR AND PRIORITY CALLING SEQUENCE
:
: STPCHV
: AT35E ;LOCATION OF NEW INTERRUPT VECTOR
:*****
003620 017637 000000 003640 STLSPV: MOV 2(SP),STPPA+2 ;SET RETURN ADR AND VECTOR
003626 062716 000002 ADD #2,JSP
003632 013701 001124 MOV TPVTR,R1
003636 012721 000000 STPPA: MOV #0,(R1)+
003642 013721 001126 MOV TPLVL,(R1)+
003646 000002 RTI ;RETURN TO CALLER

```

804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819

003650 010146
003652 013701 001170
003656 005301
003660 001376
003662 005300
003664 001372
003666 012601
003670 000002

DELAY--A COMMON ROUTINE TO DELAY PROCESSING
A GIVEN NUMBER OF MSEC.
CALLING SEQUENCE:
MOV #5,R0 ;R0 CONTAINS THE NUMBER OF MSEC DELAY DESIRED
DELAY

THE DELAY IS EFFECTED BY THE EXECUTION OF THE LOOP;
IS: DEC R1
BNE IS

SINCE THE EXECUTION TIMES OF THE PDP11 LINE DOES VARY FROM
MACHINE TO MACHINE, THE VALUE AT SYMBOLIC LOCATION
"TIMER" MUST BE CHANGED TO THE APPROPRIATE VALUE AS SHOWN BELOW
BEFORE STARTING THE DIAGNOSTIC. "TIMER" IS INITIALIZED
FOR AN 11/05,11/10(=251).

MACHINE	05810	35840	15820	BIPOLAR	11/45 & 11/70 MOS	CORE	
LOOP: DEC R1	3.4	.99	2.3	.30	.51	.90	
BNE LOOP	2.5	1.76	2.6	.60	.98	1.13	
TIME=	5.9USEC	2.75	4.9	.90USEC	1.49USEC		2.03LSEC
SET TIMER	251	554	314	2127	1237	755	

XXXXXXXXXX

```

CLY:  MOV R1, -(SP) ;SAVE R1
IS:   MOV TIMER, R1 ;MOV 1 MSEC LOOP CNT TO R1
2S:  DEC R1 ;DECREMENT COUNT
      BNE 2S ;BRANCH IF NOT ZERO
      DEC R0 ;DEC NO. OF MSEC DELAY
      BNE IS ;DELAY AGAIN IF NOT ZERO
      MOV (SP)+, R1 ;ALL DONE RESTORE R1
      RTI

```

```

820          ;*****
821          ;
822          ;PFAIL--POWER FAIL ROUTINE
823          ;   SAVE ALL REGISTERS AND SET RESTART ADDRESS
824          ;   INTO LOCATION 24
825          ;
826          ;RESTART--POWER FAIL RECOVERY
827          ;   RESTORE ALL REGISTERS AND GO TO START
828          ;*****
829
830
831 003672 010046          PFAIL:  MOV    R0,-(SP)
832 003674 010146          MOV    R1,-(SP)
833 003676 010246          MOV    R2,-(SP)
834 003700 010346          MOV    R3,-(SP)
835 003702 010446          MOV    R4,-(SP)
836 003704 010546          MOV    R5,-(SP)
837 003706 010637 000024  MOV    R6,-(SP)
838 003712 010637 003726  MOV    SP,SAVR6          ;SAVE STACK POSITION
839 003716 012737 003730 000024  MOV    *RESTR,24        ;STORE RESTART ADDRESS
840 003724 000000          HALT
841 003726 000000          SAVR6: .WORD 0
842 003730 104007          RESTR:  TYPEM
843 003732 003764          IS
844 003734 013706 003726          MOV    SAVR6,SP          ;RESTORE STACK POINTER
845 003740 012637 000024          MOV    (SP)+,24        ;RESTORE PFAIL ADDRESS
846 003744 012605          MOV    (SP)+,R5
847 003746 012604          MOV    (SP)+,R4
848 003750 012603          MOV    (SP)+,R3
849 003752 012602          MOV    (SP)+,R2
850 003754 012601          MOV    (SP)+,R1
851 003756 012600          MOV    (SP)+,R0
852 003760 000137 001304          JMP    START
853
854 003764 050200 053517 051105 15:  .ASCIZ <ACRLF>/POWER/<ACRLF>
855 003772 000200
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

:FORWARD--THIS ROUTINE TRANSFERS THE 2 OR 4 ARGUMENTS
:FROM THE TEST ROUTINE. THEY ARE;
:
: 1- ROUTINE NUMBER
: 2- ADDRESS OF NEXT TEST
: 3- ITERATION COUNT (I/O TESTS ONLY)
: 4- SCOPE ENTRY ADDRESS (I/O TESTS ONLY)
:*****

003774 013705 001136
004000 012537 001134
004004 012537 001136
004010 105737 001142
004014 100407
004016 012537 001156
004022 012537 001140
004026 010537 001174
004032 000002
004034 012737 177777 001140
004042 012737 000001 001156
004050 000766

\$FORWD: MOV NXTST,R5 ;ADDR OF NEXT TEST TO R5
MOV (R5)+,RTNNO ;GET NUMBER OF NEXT TEST
MOVL (R5)+,NXTST ;GET ADDR OF FOLLOWING TEST
TSTB PRGID ;CHECK IF I/O TEST
BMI FORWDB ;SKIP THE FETCH OF ITER CNT AND SCOPE
MOV (R5)+,ICTR ;GET ITERATION COUNT
MOV (R5)+,SCOPTR ;GET SCOPE ENTRY POINT
FORWDA: MOV R5,CURTST ;ENTRY POINT TO TEST IN CUR TST
RTI ;EXIT
FORWDB: MOV #-1,SCOPTR ;FORCE NO SCOPE
MOV #1,ICTR ;FORCE INTERATION COUNT OF 1
BR FORWDA

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

:*****

:AREAD--A ROUTINE WHICH, THROUGH THE FACILITY OF
THE MAINTENANCE BIT, OUTPUTS TO THE
PRINTER BUFFER AND READS THE KEYBOARD
STATUS DONE. IF THE DONE IS NOT SET
WITHIN 200 MSEC, THE CPU WILL HALT WITH
THE LOCATION OF THE ERROR IN RO. PRESS
CONTINUE TO CONTINUE WITH TESTS.

:*****

004052	012737	000200	001162	\$AREAD: MOV	#200, BRCTR	;SET UP 200 MSEC DELAY
004060	052777	000004	175026	BIS	#4, ATPS	;SET MAINTENANCE BIT
004066	005077	175024		CLR	ATPB	;LOAD PRINTER BUFFER
004072	105777	175012		1\$: TSTB	ATKS	;CHECK DONE BIT
004076	100410			BMI	2\$;BRANCH IF DONE
004100	012700	000001		MOV	#1, RO	;ONE TO RO
004104	104010			DELAY		;DELAY 1 MSEC.
004106	005337	001162		DEC	BRCTR	;200 MSEC OVER
004112	001367			BNE	1\$;BRANCH IF NO
004114	104002			EHALT		
004116	000755			BR	\$AREAD	;TRY AGAIN
004120	000002			2\$: RTI		;RETURN TO TEST

:*****

:CONIT--THIS ROUTINE SETS UP THE DEVICE ADDRESSES
AND INTERRUPT VECTORS FOR THE CONSOLE
TERMINAL.

:*****

004122	013700	001102		CONIT: MOV	CONADD, RO	;CONSOLE KEYBOARD STATUS ADDR TO RO
004126	010037	001110		CONSET: MOV	RO, TKS	;KEYBOARD STATUS ADDRESS (777560) TO TKS
004132	005720			TST	(RO)+	;INCREMENT RO BY TWO
004134	010037	001112		MOV	RO, TKB	;KEYBOARD DATA ADDR (777562) TO TKB
004140	005720			TST	(RO)+	;INCREMENT RO BY TWO
004142	013737	001114	004214	MOV	TPS, TPSS	;SAVE TPS OF LAST TERMINAL
004150	010037	001114		MOV	RO, TPS	;PRINTER STATUS ADDR(777564) TO TPS
004154	005720			TST	(RO)+	;INCREMENT RO BY TWO
004156	013737	001116	004216	MOV	TPB, TPBS	;SAVE TPB OF LAST TERMINAL
004164	010037	001116		MOV	RO, TPB	;PRINTER DATA ADDR (777566) TO TPB
004170	013737	001104	001120	MOV	CONVEC, TKVTR	;KEYBOARD INTERRUPT VECTOR (60) TO TKVTR
004176	013737	001104	001124	MOV	CONVEC, TPVTR	
004204	062737	000004	001124	ADD	#4, TPVTR	;PRINTER INTERRUPT VECTOR (64) TO TPVTR
004212	000207			RTS	PC	
004214	000000			TPSS: .WORD	0	;LAST TERM STATUS REG ADR
004216	000000			TPBS: .WORD	0	;LAST TERM BUFFER REG ADR

```

931          ;*****
932          ;
933          ; BINARY TO ASCII CONVERSION (1 TO 5 ASCII CHARACTERS)
934          ; CALLING SEQUENCE
935          ;     MOV     ADDRESS OF LOC TO STORE FIRST ASCII CHAR. INTO R0
936          ;     MOV     BINARY NUMBER TO BE CONVERTED INTO R1
937          ;     MOV     NUMBER TO BE CONVERTED AS A POWER OF TEN INTO R2
938          ;     BTOASC
939          ;
940          ;*****
941          ;
942 004220 010237 004304 $BTASC: MOV     R2,CNVCTR      ;SAVE TEN POWER
943 004224 006302          ASL     R2              ;R2*2
944 004226 062702 004312          ADD     #ADTENP,R2      ;CALCULATE ADDRESS OF
945          ;          ; STARTING TEN POWER
946 004232 014237 004310 1$:     MOV     -(R2),TENPWR ;POWER OF TEN VALUE TO TEN PWR
947 004236 005037 004306          CLR     DIGIT          ;CLEAR CURPENT DIGIT
948 004242 163701 004310 2$:     SUB     TENPWR,R1      ;SUBTRACT TEN POWER FROM BINARY VALUE
949 004246 103403          BCS     3$            ;BRANCH IF END
950 004250 005237 004306          INC     DIGIT
951 004254 000772          BR      2$
952 004256 063701 004310 3$:     ADD     TENPWR,R1      ;RESTORE SUBTRACTED VALUE
953 004262 062707 000060 004306          ADD     #60,DIGIT      ;CONVERT (DIGIT) TO ASCII
954 004270 113720 004306          MOVB   DIGIT,(R0)+    ;PUT ASCII CHAR INTO USER BUFFER
955 004274 005337 004304          DEC     CNVCTR        ;FINISHED ALL CHARS. CALLED FOR
956 004300 001354          BNE    1$            ;BRANCH IF NOT FINISHED
957 004302 000002          RTI                    ;YES, EXIT
958 004304 000000          CNVCTR: .WORD 0        ;CONVERSION CHARACTER COUNT
959 004306 000000          DIGIT:  .WORD 0        ;CONVERTED CHARACTER
960 004310 000000          TENPWR: .WORD 0       ;CURRENT TEN POWER
961 004312 000001 000012 000144          ADTENP: .WORD 1..10..100..1000.,10000.
962 004320 001750 023420

```

```

963          :XXXXXXXXXX
964
965          :READ-- A COMMON ROUTINE WHICH CHECKS THE KEYBOARD
966          :          DONE FLAG & SETS A FLAG INDICATING CHAR PARITY
967          :
968          :XXXXXXXXXX
969
970 004324 004737 004122 $READ: JSR    PC,CONIT      ;RESET CONSOLE ADR AND VECTORS
971 004330 005737 001154      TST    DL,CONT      ;CHECK IF MULTI DL11'S AVAILPBLE
972 004334 001430      BEQ    $READC      ;NONE, WAIT FOR CONSOLE INPUT
973 004336 013737 001154 001164 1$:  MOV    DL,CONT3      ;SET DL11 COUNT
974 004344 013737 001130 001166      MOV    FSTDL,XCSR      ;ADDRESS OF FIRST DL11 INTO XCSR
975 004352 105777 174610      2$:  TSTB   @XCSR        ;TEST IF ANY INPUT
976 004356 100005      BPL    3$              ;CONTINUE IF NO INPUT
977 004360 013700 001166      MOV    XCSR,RO        ;SET THIS DL11 AS CONSOLE
978 004364 004737 004126      JSR    PC,CONSET
979 004370 000415      BR     READ1          ;READ CHAR AND RETURN
980 004372 005337 001164      3$:  DEC    COUNT3        ;DECREMENT DL11 COUNT
981 004376 001404      BEQ    4$              ;TEST CONSOLE WHEN DONE DL11'S
982 004400 062737 000010 001166      ADD    #10,XCSR      ;NEXT DL11 ADDRESS
983 004406 000761      BR     2$              ;CONTINUE
984 004410 105777 174474      4$:  TSTB   @TKS         ;CHECK CONSOLE
985 004414 100350      BPL    1$              ;WAIT, NO INPUT
986 004416 105777 174466      $READC: TSTB   @TKS         ;CHECK KEYBOARD DONE FLAG
987 004422 100375      BPL    $READC        ;BRANCH IF NOT SET
988 004424 117737 174462 001176 READ1: MOV    @TKB,TEMPCH    ;SAVE CHARACTER
989 004432 113737 001176 001202      MOV    TEMPCH,PCHAR  ;SAVE CODE WITH PARITY BIT
990 004440 042737 177400 001202      BIC    #177400,PCHAR  ;MASK UNWANTED BITS
991 004446 113737 001176 001201      MOV    TEMPCH,PARITY+1 ;SAVE CHAR WITH PARITY BIT
992 004454 042737 177600 001176      BIC    #177600,TEMPCH ;MAKE IT 7 BIT ASCII
993 004462 023727 001176 000004      CMP    TEMPCH,#4     ;DISREGARD EOT
994 004470 001715      BEQ    $READ
995 004472 012700 000011      MOV    #11,RO        ;SET SHIFT COUNT
996 004476 042737 000377 001200      BIC    #377,PARITY   ;CLEAR PARITY FLAG
997 004504 005300      1$:  DEC    RO            ;DECREMENT SHIFT COUNT
998 004506 001406      BEQ    2$              ;EXIT IF DONE
999 004510 106337 001201      ASLB   PARITY+1      ;SHIFT CODE
1000 004514 103373      BCC    1$              ;CONTINUE IF BIT WAS ZERO
1001 004516 105137 001200      COMB   PARITY        ;CHANGE PARITY FLAG IF BIT WAS ONE
1002 004522 000770      BR     1$              ;CONTINUE
1003 004524 000002      2$:  RTI              ;SET, RET. TO CALLER
1004
1005          :XXXXXXXXXX
1006
1007          :PRINT-- A COMMON ROUTINE TO CHECK THE PRINTER READY FLAG
1008          :
1009          :XXXXXXXXXX
1010
1011 004526 105777 174362 $PRNT: TSTB   @TPS        ;CHECK PRINTER READY FLAG
1012 004532 100375      BPL    $PRNT         ;BRANCH IF NOT SET
1013 004534 000002      RTI              ;SET, RETURN
    
```



```

1014 :*****
1015 :
1016 :PRINTC--SENDS A CHARACTER AT A TIME FIRST TO THE
1017 :CONSOLE DL11 THEN TO ALL MULTIPL DL11S IF
1018 :SR BIT 13 IS = 0. IF THE REFERENCED PRINTER
1019 :READY BIT IS NOT SET. THE CHARACTER WILL NOT BE
1020 :SENT TO THAT PRINTER. ENTER WITH CHARACTER IN RO
1021 :CALL: PRINTC
1022 :
1023 :*****
1024 :
1025 004536 013737 001102 001210 $PRTC: MOV CONADD,TEMP ;SET CONSOLE ADR
1026 004544 062737 000004 001210 ADD #4,TEMP
1027 004552 105777 174432 1$: TSTB @TEMP
1028 004556 100375 BPL 1$ ;WAIT FOR CONSOLE READY
1029 004560 062737 000002 001210 ADD #2,TEMP ;SET ADR
1030 004566 010077 174416 MOV RO,@TEMP ;LOAD CONSOLE PRINT BUFFER
1031 004572 032777 020000 174412 BIT #BIT13,@SR ;CHECK SW 13
1032 004600 001003 BNE 2$ ;SEND ALL TERMS IF SW13 DOWN
1033 004602 005737 001154 TST DLCNT ;CHECK IF MULTIPLE DL11'S
1034 004606 001002 BNE 3$ ;CHECK FOR INPUT IF THERE
1035 004610 000137 005256 2$: JMP 18$
1036 004614 013737 001154 001164 3$: MOV DLCNT,COUNT3 ;PUT NO. DL11'S INTO COUNT3
1037 004622 013737 001130 001166 MOV FSTDL,XCSR ;ADDR OF FIRST DL INTO XCSR
1038 004630 023727 001134 000032 4$: CMP RTNNO,#32 ;CHECK IF TEST 32
1039 004636 001543 BEQ 13$ ;DON'T CHECK FOR INPUT IF TEST 32
1040 004640 005737 001206 TST INCHK ;CHECK FOR INPUT?
1041 004644 001140 BNE 13$
1042 004646 023727 001134 000020 CMP RTNNO,#20 ;PRINTING TEST?
1043 004654 002004 BGE 5$ ;BRANCH IF NOT
1044 004656 022737 104011 002154 CMP #TTYCTL,WAITF ;KEYBOARD CONTROL?
1045 004664 001130 BNE 13$ ;SKIP INPUT CHECK IF NOT
1046 004666 105777 174274 5$: TSTB @XCSR ;TEST IF ANY INPUT
1047 004672 100125 BPL 13$ ;CONTINUE IF NO INPUT
1048 004674 062737 000002 001166 ADD #2,XCSR ;SET BUFFER ADDRESS
1049 004702 017737 174260 001176 MOV @XCSR,TEMPCH
1050 004710 042737 177600 001176 BIC #177600,TEMPCH
1051 004716 023727 001176 000003 CMP TEMPCH,#3 ;CHECK IF CONTROL-C
1052 004724 001006 BNE 6$ ;CONTINUE IF NOT
1053 004726 023727 001134 000024 CMP RTNNO,#24 ;CHECK IF TEST 24
1054 004734 001002 BNE 6$ ;CONTINUE IF NOT CONTROL-C
1055 004736 000137 005372 JMP 20$
1056 004742 023727 001176 000177 6$: CMP TEMPCH,#177 ;CHECK IF RUBOUT
1057 004750 001427 BEQ 9$ ;YES, CHECK TEST NUMBER
1058 004752 023727 001134 000017 CMP RTNNO,#17 ;TEST 17?
1059 004760 001003 BNE 7$ ;BRANCH IF NOT
1060 004762 013703 001176 MOV TEMPCH,R3 ;SAVE CHAR
1061 004766 000461 BR 12$ ;CONTINUE
1062 004770 023727 001134 000021 7$: CMP RTNNO,#21 ;TEST 21?
1063 004776 001004 BNE 8$ ;BRANCH IF NOT
1064 005000 013737 001176 001160 MOV TEMPCH,REPT ;SAVE CHAR
1065 005006 000451 BR 12$ ;CONTINUE
1066 005010 023727 001134 000022 8$: CMP RTNNO,#22 ;TEST 22?
1067 005016 001056 BNE 14$ ;CONTINUE IF NOT

```

005020	013737	001176	001160	MOV	TEMPCH,REPT	:SAVE CHAR	
005026	000441			BR	12\$:CONTINUE	
005030	023727	001134	000021	3\$:	CMP	RTNNO,#21	:CHECK IF TEST 21
005036	001011			BNE	10\$:NO, CHECK IF TEST 22	
005040	022626			POPSP2		:ADJUST STACK	
005042	012700	000236		MOV	#30.,RO	:DELAY FOR HALF DUPLEX.	
005046	104010			DELAY			
005050	104007			TYPEN		:YES, TEST 21	
005052	017060			ECCEND		:PRINT TERMINATION MESSAGE	
005054	104005			CHAIN		:CHAIN TO NEXT TEST	
005056	000137	011664		JMP	E021A	:REPEAT TEST IF LOOP ON TEST SW SET	
005062	023727	001134	000022	10\$:	CMP	RTNNO,#22	:CHECK IF TEST 22
005070	001011			BNE	11\$:NO, CHECK IF TEST 24	
005072	022626			POPSP2		:ADJUST STACK	
005074	012700	000036		MOV	#30.,RO	:DELAY FOR HALF DUPLEX	
005102	104010			DELAY			
005102	104007			TYPEN		:YES, PRINT TERMINATION MESSAGE	
005104	017060			ECCEND			
005106	104005			CHAIN		:CHAIN TO NEXT TEST	
005110	000137	011722		JMP	E022A	:REPEAT TEST IF LOOP ON TEST SW SET	
005114	023727	001134	000024	11\$:	CMP	RTNNO,#24	:TEST 24?
005122	001137			BNE	22\$:WAIT FOR NEXT TEST IF NOT, TEST 24	
005124	022626			POPSP2		:RESET STACK	
005126	000137	012644		JMP	TERM	:TERMINATE TEST	
005132	012700	000036		12\$:	MOV	#30.,RO	:DELAY FOR HALF DUPLEX
005136	104010			DELAY			
005140	013700	001176		MOV	TEMPCH,RO	:SET NEW CHAR	
005144	000402			BR	14\$:CONTINUE	
005146	062737	000002	001166	13\$:	ADD	#2,XCSR	:SET STATUS ADDRESS IN XCSR
005154	062737	000002	001166	14\$:	ADD	#2,XCSR	
005162	013737	001102	001210		MOV	CONADD,TEMP	:CHECK IF CONSOLE TERM
005170	062737	000004	001210		ADD	#4,TEMP	:IS THIS DL
005176	023737	001210	001166		CMP	TEMP,XCSR	
005204	001420			BEG	17\$		
005206	105777	173754		15\$:	TSTB	2XCSR	:TEST PRINTER READY
005212	100375			BPL	15\$:WAIT FOR READY
005214	062737	000002	001166		ADD	#2,XCSR	:SET XCSR TO PRINTER BUFFER
005222	010377	173740			MOV	RO,2XCSR	:LOAD CHARACTER INTO BUFFER
005226	005337	001164		16\$:	DEC	COUNT3	:DECREASE COUNT OF DL1:'S
005232	00,411			BEG	18\$:ALL DONE,EXIT
005234	062737	000002	001166		ADD	#2,XCSR	:SET XCSR TO NEXT DL11 PRINTER STATUS
005242	000137	004630			JMP	4\$:GO TEST NEXT DL11 READY FLAG
005246	062737	000002	001166	17\$:	ADD	#2,XCSR	:SET XCSR TO PRINTER BUFFER
005254	000764			BR	15\$:DO NOT LOAD BUFFER
005256	023727	001134	000032	18\$:	CMP	RTNNO,#32	:CHECK IF TEST 32
005264	001514			BEG	26\$:DON'T CHECK FOR INPUT IF TEST 32
005266	005737	001236		TST	INCHK		:WANT INPUT CHECK?
005272	001111			BNE	26\$:NO, BRANCH
005274	023727	001134	000020		CMP	RTNNO,#20	:PRINTING TEST?
005302	002004			BGE	19\$:BRANCH IF NOT
005304	022737	104011	002154		CMP	@TTYCTL,WAITF	:KEYBOARD CONTROL?
005312	001101			BNE	26\$:SKIP INPUT CHECK IF NOT
005314	105777	173562		19\$:	TSTB	2CONADD	:TEST IF ANY INPUT
005320	100076			BPL	26\$:BRANCH IF NONE

```

113 005322 013737 001102 001210 MOV CONADD,TEMP ;SET ADR
1130 005330 062737 000002 001210 ADD #2,TEMP
1131 005336 117737 173646 001176 MOV# JTEMP,TEMPCH
1132 005344 042737 177600 001176 BIC #177600,TEMPCH ;MASK UNWANTED BITS
1133 005352 023727 001176 000003 CMP TEMPCH,#3 ;CHAR = CONTROL-C?
1134 005360 001013 BNE 21$ ;CONTINUE IF NOT
1135 005368 023727 001134 000024 CMP RTNNO,#24 ;TEST 24?
1136 005376 001007 BNE 21$ ;CONTINUE IF NOT
1137 005384 012700 000036 23$: MOV #30.,RO ;DELAY FOR HALF DUPLEX
1138 005392 104010 DELAY
1139 005400 104012 CRLF ;SEND CR-LF
1140 005408 022626 POPSP2 ;RESET STACK
1141 005416 000137 012516 JMP E0248 ;RETURN TO TEST
1142 005424 023727 001176 000177 21$: CMP TEMPCH,#177 ;CHECK IF RUBOUT
1143 005432 001006 BNE 23$ ;BRANCH IF NO
1144 005440 000503 BR 9$
1145 005448 012737 000001 001132 22$: MOV #1,CNTLSW ;CLEAR LOOP AND SEQUENCE BITS
1146 005456 000137 002336 JMP TTY18 ;GO WAIT FOR NEXT TEST
1147 005464 010046 23$: MOV RO,-(SP) ;SAVE RO
1148 005472 012700 000036 MOV #30.,RO ;DELAY FOR HALF DUPLEX
1149 005480 104010 DELAY
1150 005488 012620 MOV (SP)+,RO ;RESTORE RO
1151 005496 023727 001134 000017 CMP RTNNO,#17 ;CHECK IF TEST 17
1152 005504 001022 BNE 24$ ;BRANCH IF NOT TEST 17
1153 005512 013703 001176 MOV TEMPCH,R3 ;STORE INPUTTED CHAR
1154 005520 023727 001134 000021 24$: CMP RTNNO,#21 ;CHECK IF TEST 21
1155 005528 001003 BNE 25$ ;BRANCH IF NOT TEST 21
1156 005536 013737 001176 001160 MOV TEMPCH,REPT ;STORE INPUTTED CHAR
1157 005544 023727 001134 000022 25$: CMP RTNNO,#22 ;CHECK IF TEST 22
1158 005552 001003 BNE 26$ ;BRANCH IF NOT TEST 22
1159 005560 013737 001176 001160 MOV TEMPCH,REPT ;STORE INPUTTED CHAR
1160 005568 000002 26$: RTI ;RETURN TO TEST

```

```

1155 .SBTTL I/O LOGIC TESTS
1156
1157 *****
1158
1159 : ONLY THE CONSOLE TERMINAL IS TESTED.
1160 : UPON COMPLETION, THE CPU WILL EITHER HALT IF SR
1161 : BITS IS = 1 AND AWAIT FUTHER INSTRUCTIONS OR CONTINUE
1162 : AND EXECUTE THE PRINTER TESTS CONTINUOUSLY
1163 : IF AN I/O TEST FAILS, THE CPU WILL HALT AT ERRHLT
1164 : WITH THE ADDRESS OF THE ERROR IN RO (LOC 777700). PRESSING
1165 : THE CONTINUE SWITCH WILL CAUSE THE I/O TEST TO
1166 : CONTINUE WITH THE NEXT TEST. HOWEVER IF SWITCH 14
1167 : WERE SET, OR IS SET BEFORE THE CONTINUE SWITCH IS
1168 : PRESSED, THE FAILED TEST WILL LOOP ON ITSELF
1169 : WITHOUT FUTHER HALTS
1170
1171 *****
1172 : ATO-- TEST #40--TESTS THE ABILITY TO REFERENCE THE
1173 : RECEIVER STATUS WORD (RKS) WITHOUT TRAPPING.
1174
1175 *****
1176 ATO: 40 ;TEST NUMBER
1177 ATOX: AT1 ;NEXT TEST
1178 ;ITERATION COUNT

```


1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232

005604 000042
005606 005636
005610 000012
005612 005622
005614 012737
005622 005777
005636 104005
005630 000774
005632 104001
005634 000774

005636 000043
005640 005670
005642 000012
005644 005654
005646 012737
005654 005777
005660 104005
005662 000774
005664 104001
005666 000774

005632 000004
173265

005664 000004
173236

```
*****  
:AT2--TEST #42--TESTS THE ABILITY TO REFERENCE THE  
:TRANSMITTER STATUS WORD (TPS) WITHOUT TRAPPING.  
*****  
AT2: 42 ;TEST NUMBER  
AT3 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
MOV #3 MACHER ;SET UP MACHINE ERROR TRAP  
1$: TST 0TPS ;REFERENCE TRANSMITTER STATUS  
2: CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST  
3$: ERROR ;TRAPPED WHEN REFERENCING  
BR 2$ ;TRANSMITTER STATUS WORD  
  
*****  
:AT3-- TEST #43--TESTS THE ABILITY TO REFERENCE THE  
:TRANSMITTER BUFFER (TPB) WITHOUT TRAPPING.  
*****  
AT3: 43 ;TEST NUMBER  
AT4 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
MOV #3$ MACHER ;SET UP ERROR TRAP  
1$: TST 0TPB ;REFERENCE TRANSMITTER BUFFER  
2$: CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST  
3$: ERROR ;TRAPPED WHEN REFERENCING  
BR 2$ ;TRANSMITTER BUFFER.
```

```

1233 :*****
1234 :AT4-- TEST #44--TESTS THE ABILITY TO SET AND CLEAR THE
1235 : RECIIVER INTERRUPT ENABLE BIT.
1236 :*****
1237
1238 005670 000044 AT4: 44 ;TEST NUMBER
1239 005672 005760 AT5: AT6 ;NEXT TEST
1240 005674 000012 10. ;ITERATION COUNT
1241 005676 005712 1$ ;SCOPE ENTRY
1242 005700 012746 000340 MOV #PTY7,-(SP) ;SET PRIORITY 7
1243 005704 012746 005712 MOV #1$,-(SP)
1244 005710 000002 RTI
1245 005712 052777 000100 173170 1$: BIS #BIT6,@TKS ;SET INTERRUPT ENABLE BIT
1246 005720 032777 000100 173162 BIT #BIT6,@TKS ;CHECK IS BIT IS SET
1247 005726 001002 BNE 3$ ;BRANCH IF SET
1248 005730 104001 2$: ERROR ;NOT SET, ERROR
1249 005732 000410 BR 5$ ;TRY AGAIN
1250 005734 042777 000100 173146 3$: BIC #BIT6,@TKS ;CLEAR INTERRUPT ENABLE BIT
1251 005742 032777 000100 173140 BIT #BIT6,@TKS ;CHECK IF BIT IS CLEARED
1252 005750 001401 BEQ 5$ ;BRANCH IF CLEARED
1253 005752 104001 4$: ERROR ;NOT CLEARED, ERROR
1254 005754 104005 5$: CHAIN ;CHAIN TO NEXT TEST
1255 005756 000755 BR 1$ ;DO TEST AGAIN
1256
1257 :*****
1258 :AT5-- TEST #45--CHECKS THAT THE RECEIVER INTERRUPT
1259 : ENABLE BIT CAN BE CLEARED WITH RESET INSTRUCTION.
1260 :*****
1261
1262 005760 000045 AT5: 45 ;TEST NUMBER
1263 005762 006030 AT6: AT6 ;NEXT TEST
1264 005764 000012 10. ;ITERATION COUNT
1265 005766 006002 1$ ;SCOPE ENTRY
1266 005770 012746 000340 MOV #PTY7,-(SP) ;SET PRIORITY TO 7
1267 005774 012746 006002 MOV #1$,-(SP)
1268 006000 000002 RTI
1269 006002 052777 000100 173100 1$: BIS #BIT6,@TKS ;SET INTERRUPT ENABLE BIT
1270 006010 000005 RESET ;RESET
1271 006012 032777 000100 173070 BIT #BIT6,@TKS ;TEST INTERRUPT ENABLE BIT
1272 006020 001401 BEQ 2$ ;BRANCH IF CLEARED
1273 006022 104001 ERROR ;STILL SET,ERROR
1274 006024 104005 2$: CHAIN ;CHAIN TO NEXT ROUTINE
1275 006026 000755 BR 1$ ;REPEAT TEST

```

1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318

006030 000046
006032 006120
006034 000012
006036 006052
006040 012746 000340
006044 012746 006052
006050 000002
006052 052777 000100 173034
006060 032777 000100 173026
006066 001002
006070 104001
006072 000410
006074 042777 000100 173012
006102 032777 000100 173004
006110 001401
006112 104001
006114 104005
006116 000755

```
*****
:AT6-- TEST#46--TESTS THE ABILITY TO SET AND CLEAR
:          TRANSMITTER INTERRUPT ENABLE BIT.
*****
AT6:      46          ;TEST NUMBER
          AT7        ;NEXT TEST
          10.        ;ITERATION COUNT
          1$         ;SCOPE ENTRY
          MOV        #PRTY7, -(SP) ;SET PRIORITY TO 7
          MOV        #1$, -(SP)
          RTI
          1$:      BIS        #BIT6, @TPS ;SET INTERRUPT ENABLE BIT
          BIT        #BIT6, @TPS ;CHECK THAT BIT IS SET
          BNE        2$         ;BRANCH IF SET
          ERROR      ;NOT SET, ERROR
          BR         3$         ;TRY AGAIN
          2:      BIC        #BIT6, @TPS ;CLEAR INTERRUPT ENABLE BIT
          BIT        #BIT6, @TPS ;CHECK IF BIT IS CLEARED
          BEQ        3$         ;BRANCH IF CLEARED
          ERROR      ;NOT CLEARED, ERROR
          3$:     CHAIN      ;CHAIN TO NEXT TEST
          BR         1$         ;DO AGAIN
```

```
*****
:AT7-- TEST #47--TESTS THE ABILITY TO CLEAR TRANSMITTER
:          INTERRUPT ENABLE BIT WITH RESET INSTRUCTION.
*****
```

```
AT7:      47          ;TEST NUMBER
          AT10       ;NEXT TEST
          10.        ;ITERATION COUNT
          1$         ;SCOPE ENTRY
          MOV        #PRTY7, -(SP) ;SET PRIORITY TO 7
          MOV        #1$, -(SP)
          RTI
          1$:      BIS        #BIT6, @TPS ;SET INTERRUPT BIT
          RESET      ;RESET
          BIT        #BIT6, @TPS ;CHECK IF BIT IS CLEARED
          BEQ        2$         ;BRANCH IF CLEARED
          ERROR      ;ERROR, RESET DID NOT CLEAR BIT
          2$:     CHAIN      ;CHAIN TO NEXT ROUTINE
          BR         1$         ;REPEAT TEST
```

```

1319
1320
1321
1322
1323
1324 006170 000050
1325 006172 006216
1326 006174 000012
1327 006176 006200
1328 006200 000005
1329 006202 105777 172706
1330 006206 100401
1331 006210 104001
1332 006212 104005
1333 006214 000771

```

```

:*****
:AT10-- TEST #50--CHECKS THAT RESET SETS THE TRANSMITTER
:       READY BIT AND THAT THE READY BIT CAN BE READ RELIABLY.
:*****
AT10:  50           ;TEST NUMBER
       AT11        ;NEXT TEST
       10.         ;ITERATION COUNT
       1$          ;SCOPE ENTRY
1$:    RESET       ;RESET
       TSTB        ;CHECK TRANSMIT READY BIT
       @TPS        ;BRANCH IF SET
       2$          ;ERROR, RESET DID NOT SET READY BIT
       BMI         ;CHAIN TO NEXT TEST
2$:    CHAIN       ;DO AGAIN
       BR          1$

```

```

1334
1335
1336
1337
1338
1339
1340 006216 000051
1341 006220 006256
1342 006222 000012
1343 006224 006226
1344 006226 012700 000226
1345 006232 104010
1346 006234 000005
1347 006236 005077 172654
1348 006242 105777 172646
1349 006246 100001
1350 006250 104001
1351 006252 104005
1352 006254 000764

```

```

:*****
:AT11-- TEST #51--TESTS THAT THE TRANSMITTER READY RESETS
:       BY LOADING THE TRANSMITTER BUFFER.
:*****
AT11:  51           ;TEST NUMBER
       AT12        ;NEXT TEST
       10.         ;ITERATION COUNT
       1$          ;SCOPE ENTRY
1 :    MOV         ;DELAY 150 MSEC.
       #226,RO    ;RESET
       DELAY      ;LOAD TRANSMITTER BUFFER
       1$         ;CHECK TRANSMIT READY BIT
       CLR        ;BRANCH IF CLEARED
       @TPB       ;NOT CLEARED, ERROR
       @TPS       ;CHAIN TO NEXT TEST
2$:    BPL         ;REPEAT TEST
       2$         ;
       BPL        2$
       BR          1$

```



```

1353
1354
1355
1356
1357
1358 006256 000052
1359 006260 006330
1360 006262 000012
1361 006264 006272
1362 006266 104004
1363 006270 006324
1364 006272 005077 172616
1365 006276 005046
1366 006300 012746 006306
1367 006304 000002
1368 006306 052777 000100 172600
1369 006314 000240
1370 006316 104001
1371 006320 104005
1372 006322 000763
1373 006324 022626
1374 006326 000774
1375
1376
1377
1378
1379
1380
1381 006330 000035
1382 006332 006410
1383 006334 000012
1384 006336 005344
1385 006340 104004
1386 006342 006402
1387 006344 013746 001126
1388 006350 012746 006356
1389 006354 000002
1390 006356 005077 172532
1391 006362 052777 000100 172524
1392 006370 000240
1393 006372 005077 172516
1394 006376 104005
1395 006400 000761
1396 006402 022626
1397 006404 104001
1398 006406 000771

:*****
:AT12-- TEST #52--CHECKS THAT THE TRANSMIT READY BIT CAN
:      CAUSE AN INTERRUPT
:*****
AT12:  52          ;TEST NUMBER
        AT13      ;NEXT TEST
        10.       ;ITERATION COUNT
        1$        ;SCOPE ENTRY
        STPCHV    ;SET UP TRANSMITTER INTERRUPT VECTOR
        4$        ;TO 3$
1 :    CLR        @TPS ;DISABLE TRANSMIT INTERRUPT
        CLR        -(SP) ;SET PRIORITY TO ZERO
        MOV        #2$,-(SP)
        RTI
2$:    BIS        #BIT6,@TPS ;ENABLE TRANSMIT INTERRUPT
        NOP
3$:    ERROR     ;TRANSMIT READY DID NOT CAUSE INTERRUPT
        CHAIN    ;CHAN TO NEXT TEST
        BR        1$ ;REPEAT TEST
4$:    POPSP2    ;INTERRUPT OCCURED, CLEAN STACK
        BR        3$ ;CHAIN TO NEXT TEST

:*****
:AT13-- TEST#35--TESTS THAT THE TRANSMIT READY DOES NOT CAUSE AN
:      INTERRUPT WHEN THE PROCESSOR IS AT THE SAME
:*****
AT13:  35          ;TEST NUMBER
        AT14      ;NEXT TEST
        10.       ;ITERATION COUNT
        1$        ;SCOPE ENTRY
        STPCHV    ;SET UP TRANSMIT INTERRUPT
        4$        ;VECTOR TO 3$
1 :    MOV        TPLVL,-(SP) ;SET PROCESSOR TO SAME LEVEL AS XMITTER
        MOV        #2$,-(SP)
        RTI
2$:    CLR        @TPS ;DISABLE TRANSMITTER INTERRUPTS
        BIS        #BIT6,@TPS ;ENABLE TRANSMITTER INTERRUPTS
        NOP
3$:    CLR        @TPS ;OK, NO INTERRUPT OCCURED
        CHAIN    ;CHAIN TO NEXT TEST
        BR        1$ ;REPEAT TEST
4$:    POPSP2    ;INTERRUPT OCCURED,ERROR,CLEAN
        ERROR    ;UP STACK
        BR        3$ ;CHAIN TO NEXT TEST

```

```

1399
1400
1401
1402
1403
1404
1405 006410 000054
1406 006412 006474
1407 006414 000012
1408 006416 006424
1409 006420 104004
1410 006422 006462
1411 006424 005077 172464
1412 006430 013746 001126
1413 006434 162716 00C040
1414 006440 012746 006446
1415 006444 000002
1416 006446 052777 000100 172440
1417 006454 000240
1418 006456 104001
1419 006460 000401
1420 006462 022626
1421 006464 005077 172424
1422 006470 104005
1423 006472 000754

```

```

:*****
:AT14-- TEST#54--TESTS THAT THE TRANSMIT READY DOES CAUSE AN
:          INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY LEVEL
:          ONE LOWER THAN THE TRANSMIT INTERRUPT REQUEST LEVEL
:*****
AT14: 54          ;TEST NUMBER
      AT15       ;NEXT TEST
      10.        ;ITERATION COUNT
      1$        ;SCOPE ENTRY
      STPCHV     ;SET UP TRANSMIT INTERRUPT
      3$        ;VECTOR TO 2$
1$:   CHR        @TPS ;DISABLE TRANSMIT INTERRUPTS
      MOV        TPLVL, -(SP) ;SET PROCESSOR PRIORITY ONE
      SUB        #40, (SP)    ;LEVEL LOWER THAN TRANSMITTER
      MOV        #2$, -(SP)
      RTI
2$:   BIS        #BIT6, @TPS ;ENABLE TRANSMITTER INTERRUPTS
      NOP
      ERRJR
      BR         4$
3$:   POPSP2
4$:   CLR        @TPS
      CHAIN
      BR         1$
;NO INTERRUPT, ERROR
;CHAIN TO NEXT TEST
;INTERRUPT OCCURED, OK, CLEAN STACK
;DISABLE TRANSMITTER INTERRUPTS
;CHAIN TO NEXT TEST
;REPEAT TEST

```

```

1424
1425
1426
1427
1428
1429
1430 006474 000055
1431 006476 006574
1432 006500 000012
1433 006502 006504
1434 006504 104004
1435 006506 006546
1436 006510 005077 172400
1437 006514 005046
1438 006516 012746 006524
1439 006522 000002
1440 006524 052777 000100 172362
1441 006532 000240
1442 006534 104001
1443 006536 005077 172352
1444 006542 104005
1445 006544 000757
1446 006546 012777 006566 172350
1447 006554 012716 006562
1448 006560 000002
1449 006562 000240
1450 006564 000764
1451 006566 022626
1452 006570 104001
1453 006572 000761
1454
1455
1456
1457
1458
1459 006574 000056
1460 006576 006632
1461 006600 000012
1462 006602 006604
1463 006604 012700 000226
1464 006610 104010
1465 006612 104021
1466 006614 000005
1467 006616 105777 172266
1468 006622 100001
1469 006624 104001
1470 006626 104005
1471 006630 000770

;*****
;AT15-- TEST#55--TESTS THAT THE TRANSMIT READY DOES NOT
;REINTERRUPT AFTER AN RTI WHEN THE READY BIT HAS
;NOT BEEN RESET.
;*****
AT15: 55 ;TEST NUMBER
      AT16 ;NEXT TEST
      10. ;ITERATION COUNT
      1$ ;SCOPE ENTRY
1$: STPCHV ;SET TRANSMIT INTERRUPT VECTOR
   4$ ;TO 3$
   CLR @TPS ;DISABLE TRANSMITTER INTERRUPTS
   CLR -(SP) ;SET PROCESSOR PRIORITY TO ZERO
   MOV #2$,-(SP)
RTI ;
2$: BIS #BIT6,@TPS ;ENABLE TRANSMITTER INTERRUPTS
   NOP
   ERROR ;ERROR1, TRANSMITTER FAILED TO INTERRUPT
3$: CLR @TPS ;DISABLE TRANSMITTER INTERRUPTS
   CHAIN ;CHAIN TO NEXT TEST
   BR 1$ ;REPEAT TEST
4$: MOV #6$,@TPVTR ;INTERRUPT OCCURED, CHANGE INTERRUPT
   MOV #5$,@SP ;VECTOR TO 5$ AND RETURN TO 4$
RTI ;RETURN FROM INTERRUPT
5$: NOP
   BR 3$ ;CHAIN TO NEXT TEST
6 : POPSP2 ;ERROR2, TRANSMITTER REINTERRUPTED
   ERROR ;AFTER RTI WITH READY BIT LEFT ON.
   BR 3$ ;CLEAN STACK, CHAIN TO NEXT TEST.

;*****
;AT16--TEST#56--CHECKS THAT RESET CLEARS THE RECEIVER DONE BIT
;*****
AT16: 56 ;TEST NUMBER
      AT17 ;NEXT TEST
      10. ;ITERATION COUNT
      1$ ;SCOPE ENTRY
1$: MOV #226,R0 ;DELAY 150 MSEC.
   DELAY ;ENABLE RECEIVER
2$: AREAD ;RESET
   RESET ;TEST DONE BIT
   TSTB @TKS ;BRANCH IF DONE IS CLEARED
   BPL 3$ ;NOT CLEARED, ERROR
3$: CHAIN ;CHAIN TO NEXT TEST
   BR 2$ ;REPEAT TEST

```

```

1472
1473
1474
1475
1476
1477 006632 000057
1478 006634 006672
1479 006636 000012
1480 006640 006542
1481 006642 012700 000226
1482 006646 104010
1483 006650 104021
1484 006652 105777 172234
1485 006656 105777 172226
1486 006662 100001
1487 006664 104001
1488 006666 104005
1489 006670 000767
1490
1491
1492
1493
1494
1495
1496 006672 000060
1497 006674 006754
1498 006676 000012
1499 006700 006716
1500 006702 104003
1501 006704 006746
1502 006706 012700 000226
1503 006712 104010
1504 006714 104021
1505 006716 005077 172166
1506 006722 005046
1507 006724 012746 006732
1508 006730 000002
1509 006732 052777 000100 172150
1510 006740 000240
1511 006742 104001
1512 006744 000401
1513 006746 022626
1514 006750 104005
1515 006752 000761

;*****
;AT17-- TEST#57--CHECKS THAT REFERENCING THE RECEIVER BUFFER
;          CLEARS THE DONE BIT.
;*****
AT17:    57          ;TEST NUMBER
          AT20       ;NEXT TEST
          10.        ;ITERATION COUNT
          1$         ;SCOPE ENTRY
1$:      MOV        #226,R0
          DELAY      ;DELAY 150 MSEC.
          2$:      AREAD      ;ENABLE RECEIVER
          TSTB      @TKB    ;REFERENCE RECEIVER BUFFER
          TSTB      @TKS    ;TEST DONE BIT
          BPL       3$      ;BRANCH IF NOT SET
          ERROR     ;DONE BIT IS SET, ERROR
          3$:      CHAIN     ;CHAIN TO NEXT TEST
          BR        2$      ;REPEAT TEST

;*****
;AT20-- TEST#60--CHECK THAT THE RECEIVER DONE BIT IS ABLE TO
;          CAUSE AN INTERRUPT.
;*****
AT20:    60          ;TEST NUMBER
          AT21       ;NEXT TEST
          10.        ;ITERATION COUNT
          2$         ;SCOPE ENTRY
          STRDRV     ;SET UP RECEIVER INTERRUPT
          4$         ;VECTOR TO 3$
1$:      MOV        #226,R0
          DELAY      ;DELAY 150 MSEC
          AREAD     ;ENABLE RECEIVER
          2$:      CLR        @TKS
          CLR        -(SP)  ;DISABLE RECEIVER INTERRUPTS
          MOV        #3$,-(SP) ;SET PROCESS STATUS TO ZERO
          RTI
          3$:      BJS        #BIT6,@TKS ;ENABLE RECEIVER INTERRUPT
          NOP
          ERROR     ;ERR RECEIVER FAILED TO INTERRUPT
          BR        5$      ;CHAIN TO NEXT TEST
          4$:      POPSP2
          5$:      CHAIN     ;OK, CLEAN STACK
          BR        2$      ;CHAIN TO NEXT TEST
          ;REPEAT TEST

```

```

1516                                     ;*****
1517                                     ;AT21-- TEST#61--TESTS THAT THE RECEIVER DONE DOES NOT CAUSE AN
1518                                     ;      INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL AS
1519                                     ;      THE RECEIVER'S INTERRUPT REQUEST LEVEL.
1520                                     ;*****
1521
1522 006754 000061 AT21: 51 ;TEST NUMBER
1523 006756 007944          AT22 ;NEXT TEST
1524 006760 000012          10. ;ITERATION COUNT
1525 006762 007000          2$ ;SCOPE ENTRY
1526 006764 104003          STRDRV ;SET RECEIVER VECTOR TO 4$
1527 006766 007036          5$
1528 006770 012700 C00226 1$: MOV #226,R0
1529 006774 104010          DELAY ;DELAY 150 MSEC
1530 006776 104021          AREAD ;ENABLE RECEIVER
1531 007000 005077 172104 2$: CLR @TKS ;DISABLE RECEIVER INTERRUPTS
1532 007004 013746 001122     MOV TKLVL,-(SP) ;SET PROCESSOR TO SAME LEVEL AS RECEIVER
1533 007010 012746 007016     MOV #3$,-(SP)
1534 007014 000002          RTI
1535 007016 052777 000100 172064 3$: BIS #BIT6,@TKS ;ENABLE RECEIVER INTERRUPTS
1536 007024 000240          NOP
1537 007026 005077 172056 4$: CLR @TKS ;OK, NO INTERRUPT OCCURED
1538 007032 104005          CHAIN ;CHAIN TO NEXT TEST
1539 007034 000770          BR 3$ ;REPEAT TEST
1540 007036 022626          POPSP2 ;ERROR, RECEIVER INTERRUPTED. CLEAN STACK
1541 007040 104001          ERROR
1542 007042 000771          BR 4$ ;BRANCH 3$

```

```

1543
1544
1545
1546
1547
1548
1549
1550 007044 000062
1551 007046 007142
1552 007050 000012
1553 007052 007070
1554 007054 104003
1555 007056 007130
1556 007060 012700 000226
1557 007064 104010
1558 007066 104021
1559 007070 005077 172014
1560 007074 013746 001122
1561 007100 012746 007106
1562 007104 000002
1563 007106 162737 000040 177776
1564 007114 052777 000100 171766
1565 007122 000240
1566 007124 104001
1567 007126 000401
1568 007130 022626
1569 007132 005077 171752
1570 007136 104005
1571 007140 000753

```

```

:*****
:AT22-- TEST#62--TESTS THAT THE RECEIVER DONE DOES CAUSE AN
:          INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY ONE
:          LEVEL LOWER THAN THE RECEIVER'S INTERRUPT
:          REQUEST LEVEL
:*****
AT22: 62 ;TEST NUMBER
      AT23 ;NEXT TEST
      10. ;ITERATION COUNT
      1$ ;SCOPE ENTRY
      STRDRV ;SET RECEIVER INTERRUPT
      3$ ;VECTOR TO 2$
      MOV #226,R0
      DELAY ;DELAY 150 MSEC
      AREAD ;ENABLE RECEIVER
1$: CLR @TKS ;DISABLE READER INTERRUPTS
   MOV TKLVL,-(SP) ;SET PROCESSOR ONE LEVEL
   MOV #2$,-(SP)
RTI
2$: SUB #40,PSW ;LOWER THAN READERS
   BIS #BIT6,@TKS ;ENABLE INTERRUPTS
NOP
ERROR ;FAILED TO INTERRUPT
BR 4$ ;CHAIN TO NEXT TEST
3$: POPSP2 ;OK, CLEAN STACK
4$: CLR @TKS ;DISABLE RECEIVER INTERRUPTS
   CHAIN ;CHAIN TO NEXT TEST
   BR 1$ ;REPEAT TEST

```



```

1646
1647
1648
1649
1650
1651
1652 007316 000001
1653 007320 007440
1654 007322 104016
1655 007324 012701 000040
1656 007330 012702 000100
1657 007334 012703 000140
1658 007340 110100
1659 007342 004737 007410
1660 007346 110200
1661 007350 004737 007410
1662 007354 012704 000003
1663 007360 110300
1664 007362 104015
1665 007364 005304
1666 007366 001375
1667 007370 104012
1668 007372 122122
1669 007374 105723
1670 007376 020327 000200
1671 007402 103756
1672 007404 104005
1673 007406 000746
1674 007410 012704 000003
1675 007414 104015
1676 007416 005304
1677 007420 001375
1678 007422 012700 000040
1679 007426 104015
1680 007430 012700 000040
1681 007434 104015
1682 007436 000207

```

```

:XXXXXXXXXX
:PT1 -- PRINTER CHARACTER TEST --- PRINTS ALL PRINTABLE CHARACTERS
:XXXXXXXXXX

PT1: 1 :TEST NUMBER
      PT2 :NEXT TEST
      PRTHDR
1$: MOV #40,R1 :SPACE TO R1
      MOV #100,R2 :@ TO R2
      MOV #140,R3 :\ TO R3
2$: MOVB R1,R0 :CHAR TO R0
      JSR PC,SPSP :SEND TWO SPACES
      MOVB R2,R0 :NEXT CHAR TO R0
      JSR PC,SPSP :SEND TWO SPACES
3$: MOV #3,R4 :PRINT COUNT TO R4
      MOVB R3,R0 :THIRD CHAR TO R0
      PRINTC :PRINT THE CHAR
      DEC R4 :THREE TIMES ?
      BNE 3$ :BRANCH IF NOT
      CRLF :CARRIAGE RETURN LINE FEED
      CMPB (R1)+,(R2)+ :NEXT CHARACTERS
      TSTB (R3)+
      CMP R3,#200 :CHECK IF ALL DONE
      BLO 2$ :BRANCH IF NOT
      CHAIN :EXIT TO NEXT TEST
      BR 1$ :REPEAT TEST
SPSP: MOV #3,R4 :PRINT COUNT TO R4
1: PRINTC :PRINT CHAR
      DEC R4 :THREE TIMES?
      BNE 1$ :BRANCH IF NOT
SP2: MOV #40,R0 :SPACE TO R0
      PRINTC :SEND A SPACE
SPC: MOV #40,R0 :SPACE TO R0
      PRINTC :SEND ANOTHER
      RTS PC :RETURN

```

```

1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696 007440 000002
1697 007442 010070
1698 007444 104016
1699 007446 005003
1700 007450 012701 007626
1701 007454 012702 000003
1702 007460 012704 000010
1703 007464 112100
1704 007466 104015
1705 007470 005304
1706 007472 001374
1707 007474 022703 000002
1708 007500 001420
1709 007502 022703 000004
1710 007506 001414
1711 007510 022703 000033
1712 007514 001412
1713 007516 022703 000007
1714 007522 001002
1715 007524 012703 000020
1716 007530 022703 000040
1717 007534 001003
1718 007536 000421
1719 007540 005203
1720 007542 005203
1721 007544 010300
1722 007546 012704 000003
1723 007552 104015
1724 007554 005304
1725 007556 001375
1726 007560 005203
1727 007562 005302
1728 007564 001404
1729 007566 004737 007422
1730 007572 104015
1731 007574 000731
1732 007576 104012
1733 007600 000725
1734 007602 012704 000003
1735 007606 012700 000177
1736 007612 104015

```

```

:XXXXXXXXXX
PT2 -- NON-PRINTING CHARACTER TEST. THIS TEST
PRINTS THE OCTAL CODE FOLLOWED BY THE MNEMONIC
OF ALL NON-PRINTING CHARACTERS. FOLLOWING EACH
MNEMONIC, THE PRINTER IS DRIVEN BY THE NON-PRINTING
CODE (000 THROUGH 037 PLUS 177)
ALL CONTROL CHARACTERS (INCLUDING THOSE FOR OPTIONS
WILL BE SKIPPED. REFER TO THE DOCUMENT FOR A LIST OF THOSE
PRINTED.
:XXXXXXXXXX
PT2:      2      :TEST NUMBER
          PT3    :NEXT TEST
          PRTHDR
15:      CHR     R3      :NON-PRINTABLE CHAR..NULL IS FIRST
          MOV     #LINE2,R1 :ADDR OF CHAR STRING TO R1
25:      MOV     #3,R2     :NO. OF CHAR GROUPS PER LINE
35:      MOV     #10,R4    :NO. OF CHARS PER GROUP
45:      MOV8    (R1)+,R0  :CHAR INTO R0
          PRINTC :PRINT CHAR
          DEC     R4      :8 CHARS. PRINTED?
          BNE    45      :BRANCH IF NOT
          CMP     #2,R3   :CHAR = STX?
          BEQ    75      :YES, SET NEXT CHAR
          CMP     #4,R3   :CHAR = EOT?
          BEQ    65      :YES, SET NEXT CHAR
          CMP     #32,R3  :CHAR = ESC?
          BEQ    75      :SET NEXT CHAR
          CMP     #7,R3   :YES, IS NON-PRINTABLE CHAR A BELL?
          BNE    55      :BRANCH IF NOT A BELL
          MOV     #20,R3  :SET NEXT CHAR
55:      CMP     #40,R3   :NO, IS IT THE LAST
          BNE    85      :BRANCH IF NO
          BR     115     :YES, OUTPUT LAST CHAR (177)
65:      INC     R3      :SKIP CHAR
75:      INC     R3      :SKIP CHAR
85:      MOV     R3,R0   :NON-PRINTABLE CHAR TO R0
          MOV     #3,R4  :A COUNT OF 3 TO R4
95:      PRINTC :DRIVE PRINTER WITH NON-PRINTABLE CHAR
          DEC     R4      :DECREMENT COUNTER
          BNE    95      :BRANCH IF NOT ZERO (3 TIMES)
          INC     R3      :INCREMENT CHAR CODE
          DEC     R2      :DEC. GROUPS PER LINE COUNTER (3)
          BEQ    105     :BRANCH IF ZERO
          JSR    PC,SP2  :SEND 3 SPACES
          BR     35
105:     CALF    :CONTINUE
          BR     25      :SEND A CR LF
          BR     25      :GO DO NEXT LINE
115:     MOV     #3,R4   :A 3 COUNT TO R4
          MOV     #177,R0 :A DEL TO R0
125:     PRINTC :PRINT CHAR

```

1737	007614	005304	DEC	R4	: DECREMENT COUNTER
1738	007616	001375	BNE	125	: BRANCH IF NOT ZERO
1739	007620	104012	CRLF		: SEND A CR, LF
1740	007622	040005	CHAIN		: CHAIN TO NEXT TEST
1741	007624	000710	BR	15	: REPEAT TEST

1742					
1743					
1744					
1745	007626	030060	020060	047040	LINE2: .ASCII /000 NUL001 SOHC02 STX/
1746	007634	046125	030060	020061	
1747	007642	051440	044117	030060	
1748	007650	020062	051440	054124	
1749	007656	030060	020066	040440	*ASCII /006 ACK020 DLE021 DC1/
1750	007664	045503	031060	020060	
1751	007672	042040	042514	031060	
1752	007700	020061	042040	030503	
1753	007706	031060	020062	042040	.ASCII /022 DC2023 DC3024 DC4/
1754	007714	031103	031060	020063	
1755	007722	042040	031503	031060	
1756	007730	020064	042040	032103	
1757	007736	031060	020065	047040	.ASCII /025 NAK026 SYN027 ETB/
1758	007744	045501	031060	020066	
1759	007752	051440	047131	031060	
1760	007760	020067	042440	041124	
1761	007766	031460	020060	041440	.ASCII /030 CAN031 EM 032 SUB/
1762	007774	047101	031460	020061	
1763	010002	042440	020115	031460	
1764	010010	020062	051440	041125	
1765	010016	031460	020064	043040	.ASCII /034 FS 035 GS 036 RS /
1766	010024	020123	031460	020065	
1767	010032	043440	020123	031460	
1768	010040	020066	051040	020123	
1769	010046	031460	020067	052440	*ASCII /037 US 177 DEL /
1770	010054	020123	033461	020067	
1771	010062	042040	046105	040	
1772	010070				.EVEN

1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813

010070 000003
010072 010210
010074 104016
010076 005037 001172
010102 013701 001150
010106 012700 000117
010112 104015
010114 005301
010116 001404
010120 004737 007430
010124 005301
010126 001367
010130 104022
010132 012737 000001 001172
010140 013701 001172
010144 004737 007430
010150 005301
010152 001374
010154 012700 000130
010160 104015
010162 104022
010164 062737 000002 001172
010172 023737 001172 001150
010200 103757
010202 104014
010204 104005
010206 000733

:XXXXXXXXXX
PT3 -- CARRIAGE RETURN TEST
THE LINE CONSISTS OF A STRING OF O'S AND X'S. FIRST, THE O'S ARE PRINTED OUT TO THE LAST COLUMN WITH A SPACE SEPARATING EACH. THEN THE CARRIAGE IS SPACED TO THE FIRST BLANK SPACE, AND X PRINTED AND THE RETURNED TO THE MARGIN. THIS PROCESS IS CONTINUE UNTIL ALL SPACES BETWEEN THE ZEROES HAVE BEEN FILLED.

:XXXXXXXXXX
PT3: 3 ;TEST NUMBER
PT4 ;NEXT TEST
PRTHDR ;TYPE HEADER
1\$: CLR SPCNT ;CLEAR SPACE COUNTER
MOV WIDTH,R1 ;POSITION COUNTER TO R1
2\$: MOV #117,R0 ;"0" TO R0
PRINTC ;PRINT THE "0"
DEC R1 ;DECREMENT POSITION COUNTER
BEQ 3\$;BRANCH IF 0
JSR PC,SPC ;SEND SPACE
DEC R1 ;DECREMENT POSITION COUNTER
BNE 2\$;BRANCH IF NOT ZERO
3\$: CR ;SEND A CR
MOV #1,SPCNT ;SPACE COUNTER SET TO 1
4\$: MOV SPCNT,R1 ;NO. OF SPACES TO R1
5\$: JSR PC,SPC ;SEND SPACE
DEC R1 ;DECREMENT SPACE COUNTER
BNE 5\$;BRANCH IF NOT ZERO
MOV #130,R0 ;"X" INTO R0
PRINTC ;PRINT "X"
CR ;PRINT CR
ADD #2,SPCNT ;INCREMENT SPACE COUNT BY 2
CMP SPCNT,WIDTH ;COMPARE POSITION COUNTER WITH COLM. COUNT
BLO 4\$;BRANCH IF LOWER
LF ;SEND LF
CHAIN ;CHAIN TO NEXT TEST
BR 1\$;REPEAT TEST

```

1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824 010210 000004
1825 010212 010366
1826 010214 104016
1827 010216 012737 000001 001204
1828 010224 013701 001150
1829 010230 012702 010350
1830 010234 004737 010320
1831 010240 013701 001204
1832 010244 104014
1833 010246 005301
1834 010250 001375
1835 010252 006337 001204
1836 010256 022737 000100 001204
1837 010264 001406
1838 010266 112200
1839 010270 104015
1840 010272 112200
1841 010274 104015
1842 010276 104022
1843 010300 000757
1844 010302 013701 001150
1845 010306 004737 010320
1846 010312 104014
1847 010314 104005
1848 010316 000737
1849 010320 112200
1850 010322 104015
1851 010324 112200
1852 010326 104015
1853 010330 005741
1854 010332 012700 000137
1855 010336 104015
1856 010340 005301
1857 010342 001375
1858 010344 104022
1859 010346 000207
1860
1861 010350 030460 031060 032060
1862 010356 034060 033061 031063
1863 010364 030060

```

```

:XXXXXXXXXX
PT4 -- MULTIPLE LINE FEED TEST -- 63 LINE FEEDS ARE
SENT WITH A REFERENCE LINE AT THE START AND END.
A NUMBER IS PRINTED WHICH INDICATES THE NUMBER OF LINE
FEEDS THAT WILL BE ISSUED BEFORE THE NEXT
NUMBER OR REFERENCE LINE IS PRINTED.
:XXXXXXXXXX

```

```

PT4: 4 ; TEST NUMBER
PTS ; NEXT TEST
PRTHDR ; TYPE HEADER
1S: MOV #1,LFCNT ; LINE FEED COUNT TO 1
MOV WIDTH,R1 ; COLUMN COUNT TO R1
MOV #LINE3,R2 ; ADDR OF NUMBER FIELD TO R2
JSR PC,REF ; PRINT REFERENCE LINE
2S: MOV LFCNT,R1 ; LINE FEED COUNT TO R1
3S: LF ; SEND LF
DEC R1 ; DECREMENT COUNTER
BNE 3S ; BRANCH IF NOT YET 0
ASL LFCNT ; DOUBLE LINE FEED COUNT
CMP #BIT6,LFCNT ; TEST IF COUNT IS 32
BEQ 4S ; BRANCH IF =32, END
MOVB (R2)+,R0 ; NUMBER TO R0
PRINTC ; PRINT IT
MOVB (R2)+,R0 ; NUMBER TO R0
PRINTC ; PRINT IT
CR ; PRINT CR
BR 2S ; DRIVE THE LINEFEEDS
4: MOV WIDTH,R1 ; COLUMN COUNT TO R1
JSR PC,REF ; SEND END REFERENCE LINE
LF ; ADVANCE PAPER
CHAIN
BR 1S ; REPEAT TEST
REF: MOVB (R2)+,R0 ; NUMBER TO R0
PRINTC ; PRINT IT
MOVB (R2)+,R0 ; NUMBER TO R0
PRINTC ; PRINT IT
TST -(R1) ; DECREASE COUNTER BY 2
MOV #137,R0 ; DASH (-) TO R0
1S: PRINTC ; PRINT IT
DEC R1 ; DECREMENT COLUMN COUNTER
BNE 1S ; BRANCH IF NO ZERO
CR ; PRINT CR
RTS PC ; RETURN

```

```

LINE3: .ASCII /01020408163200/

```

```

1864 :XXXXXXXXXX
1865 ;PTS-- SINGLE LINE FEED TEST -- TESTS THE LINE FEED
1866 : CAPABILITY FROM ALL COLUMNS.
1867 ;XXXXXXXXXX
1868
1869 010366 000005 PTS: 5 :TEST NUMBER
1870 010370 010572 PT6 :NEXT TEST
1871 010372 104016 PRTHDR :TYPE HEADER
1872 010374 013701 001150 1$: MOV WIDTH,R1 :COLUMN COUNT TO R1
1873 010400 005741 TST -(R1) :DECREASE BY 2
1874 010402 012700 000060 MOV #60,R0 :'0' TO R0
1875 010406 104015 2$: PRINTC :SEND 0
1876 010410 005301 DEC R1 :DECREMENT COLUMN COUNTER
1877 010412 001375 BNE 2$ :BRANCH IF NOT ZERO
1878 010414 012700 000062 MOV #62,R0 :SEND A 2
1879 010420 104015 PRINTC
1880 010422 104015 PRINTC :SEND A SECOND TWO
1881 010424 023727 001150 000204 CMP WIDTH,#132. :COMPARE COLUMN COUNT
1882 010432 001404 BEQ 3$ :BRANCH IF EQ 132
1883 010434 012700 003410 MOV #3410,R0 :DELAY 1.8 SEC
1884 010440 104010 DELAY
1885 010442 000407 BR 5$
1886 010444 012700 000063 3$: MOV #63,R0 :3'S TO R0
1887 010450 012701 000100 MOV #100,R1 :64 TO COUNTER
1888 010454 104015 4$: PRINTC :SEND CHARACTER
1889 010456 005301 DEC R1 :DECREMENT COUNT
1890 010460 001375 BNE 4$ :BRANCH IF NOT ZERO
1891 010462 104012 5$: CRLF :SEND A CR,LF
1892 010464 013701 001150 MOV WIDTH,R1 :NO. COLUMNS TO R1
1893 010470 012700 000134 6$: MOV #134,R0 :BACKSLASH TO R0
1894 010474 104015 PRINTC :SEND IT
1895 010476 104014 LF :PRINT LF
1896 010500 005301 DEC R1 :DECREMENT COUNTER
1897 010502 001372 BNE 6$ :BRANCH IF NOT ZERO.
1898 010504 104022 CR :SEND CR
1899 010506 004737 010534 JSR PC,PTSAL :SEND REF LINE #1
1900 010512 104012 CRLF :SEND A CR,LF
1901 010514 012700 001750 MOV #1750,R0 :DELAY 1 SEC
1902 010520 104010 DELAY
1903 010522 004737 010534 JSR PC,PTSAL :SEND A SECOND REF. LINE
1904 010526 104012 CRLF :SEND A CR,LF
1905 010530 104005 CHAIN :CHAIN TO NEXT TEST
1906 010532 000720 BR 1$ :REPEAT TEST
1907 010534 013701 001150 PTSAL: MOV WIDTH,R1 :COLUMN COUNT TO R1
1908 010540 012700 000061 MOV #61,R0 :'1' TO R0
1909 010544 104015 1$: PRINTC :PRINT R0
1910 010546 005301 DEC R1 :DECREMENT COUNTER
1911 010550 001407 BEQ 2$ :BRANCH IF=0
1912 010552 005200 INC R0 :INCREMENT CHARACTER
1913 010554 020027 000071 CMP R0,#71 :COMP CHAR TO "9"
1914 010560 101771 BLOS 1$ :BRANCH IF LOWER OR SAME
1915 010562 012700 000060 MOV #60,R0 :RESET CHAR TO "0"
1916 010566 000766 BR 1$ :CONTINUE
1917 010570 000207 2$: RTS PC :FINISHED, RETURN TO CALLER

```

```

1918
1919
1920
1921
1922
1923
1924
1925
1926
1927 010572 000006
1928 010574 010760
1929 010576 104016
1930 010600 104007
1931 010602 016715
1932 010604 013701 001150
1933 010610 005741
1934 010612 012700 000060
1935 010616 104015
1936 010620 005301
1937 010622 001375
1938 010624 012700 000062
1939 010630 104015
1940 010632 104015
1941 010634 023727 001150 000204
1942 010642 001404
1943 010644 012700 003410
1944 010650 104010
1945 010652 000407
1946 010654 012700 000063
1947 010660 012701 000100
1948 010664 104015
1949 010666 005301
1950 010670 001375
1951 010672 104012
1952 010674 013701 001150
1953 010700 012700 000134
1954 010704 104015
1955 010706 012700 000010
1956 010712 104015
1957 010714 012700 000057
1958 010720 104015
1959 010722 005301
1960 010724 001365
1961 010726 104014
1962 010730 104022
1963 010732 004737 010534
1964 010736 104012
1965 010740 012700 001750
1966 010744 104010
1967 010746 004737 010534
1968 010752 104012
1969 010754 104005
1970 010756 000712

```

```

:XXXXXXXXXX
PT6-- BACKSPACE TEST -- A REFERENCE LINE SUCH AS IN
:          TEST PT5 IS PRINTED.  THE SECOND LINE CONSISTS
:          OF PRINTING A BACKSLASH, BACKSPACE AND FORWARD
:          SLASH COMBINATION OUT TO THE GIVEN COLUMN WIDTH.
:          THIS LINE IS THEN FOLLOWED BY THE SAME TWO REFERENCE
:          LINES AS PRINTED IN TEST PT5.
:XXXXXXXXXX

PT6:  6          ;TEST NUMBER
      PT7        ;NEXT TEST
      PRTHDR     ;PRINT HEADER
      TYPEN      ;PRINT COLUMN # MSG
      HDRO
1$:   MOV        WIDTH,R1 ;COLUMN COUNT TO R1
      TST        -(R1)    ;DECREMENT BY 2
      MOV        #60,R0   ;"0" TO R0
2$:   PRINTC     ;SEND 0
      DEC        R1       ;DECREMENT COLUMN COUNTER
      BNE        2$       ;BRANCH IF NOT ZERO
      MOV        #62,R0   ;"2" TO R0
      PRINTC     ;SEND A "2"
      PRINTC     ;SEND A SECOND "2"
      CMP        WIDTH,#132. ;COMPARE COLUMN COUNT
      BEQ        3$
      MOV        #3410,R0 ;DELAY 1.8 SEC
3$:   BR         5$
      MOV        #63,R0   ;3'S TO R0
      MOV        #100,R1  ;64 TO CHAR COUNT
4$:   PRINTC     ;SEND CHAR
      DEC        R1       ;DECREMENT CHAR COUNT
      BNE        4$       ;CONTINUE IF NOT DONE
5$:   CRLF      ;SEND A CR,LF
      MOV        WIDTH,R1 ;COLUMN COUNT TO R1
6:   MOV        #130,R0  ;BACKSLASH TO R0
      PRINTC     ;SEND IT
      MOV        #10,R0   ;BACKSPACE TO R0
      PRINTC     ;SEND IT
      MOV        #57,R0   ;FORWARD SLASH TO R0
      PRINTC     ;SEND IT
      DEC        R1       ;END OF PAPER
      BNE        6$       ;BRANCH IF NO
      LF         ;SEND LF
      CR         ;SEND CR
      JSR        PC,PT5AL ;SEND REF LINE #1
      CRLF      ;SEND A CR,LF
      MOV        #1750,R0 ;DELAY 1 SEC
      DELAY     ;
      JSR        PC,PT5AL ;SEND SECOND REF LINE
      CRLF      ;SEND A CR,LF
      CHAIN     ;CHAIN TO NEXT TEST
      BR        1$       ;REPEAT TEST

```

```

1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982 010760 000007
1983 010762 011172
1984 010764 104016
1985 010766 012703 000002
1986 010772 013701 001150
1987 010776 012700 000115
1988 011002 104015
1989 011004 005301
1990 011006 001404
1991 011010 004737 007430
1992 011014 005301
1993 011016 001367
1994 011020 022703 000002
1995 011024 001003
1996 011026 104022
1997 011030 005303
1998 011032 000757
1999 011034 005703
2000 011036 001373
2001 011040 104012
2002 011042 005723
2003 011044 013701 001150
2004 011050 004737 007430
2005 011054 005301
2006 011056 001405
2007 011060 012700 000100
2008 011064 104015
2009 011066 005301
2010 011070 001367
2011 011072 022703 000002
2012 011076 001003
2013 011100 104022
2014 011102 005303
2015 011104 000757
2016 011106 005703
2017 011110 001373
2018 011112 104012
2019 011114 005723
2020 011116 013701 001150
2021 011122 012700 000046
2022 011126 104015
2023 011130 005301
2024 011132 001404

```

```

:XXXXXXXXXX
:PT7-- OVERPRINT TEST-- A ROW OF ALTERNATING M'S AND
: SPACES ARE PRINTED, OUT TO THE LAST COLUMN AND OVERPRINT TWICE.
: A SECOND LINE OF ALTERNATING SPACES AND "Q'S" IS THEN
: SENT 3 TIMES AS THE FIRST LINE. THIS IS FOLLOWED
: BY A THIRD AND FINAL LINE OF ALTERNATING '8'
: AND SPACES.
:XXXXXXXXXX

```

```

PT7: 7 ;TEST NUMBER
PT10 ;NEXT TEST
PRTHDR ;PRINT MESSAGE
1$: MOV #2,R3 ;2 COUNT TO R3
2$: MOV WIDTH,R1 ;NO. OF COLUMNS TO R1
3$: MOV #115,R0 ;"M" TO R0
PRINTC ;SEND IF
DEC R1 ;END OF LINE
BEQ 4$ ;BRANCH IF YES
JSR PC,SPC ;SEND SPACE
DEC R1 ;END OF LINE?
BNE 3$ ;BRANCH IF NO
4$: CMP #2,R3 ;TEST R3
BNE 6$ ;BRANCH IF NOT FIRST TIME
5$: CR ;SEND CR
DEC R3 ;DECREASE LINE COUNTER
BR 2$ ;REPEAT LINE
6$: TST R3 ;THIRD TIME?
BNE 5$ ;BRANCH IF NOT
CRLF ;NEXT LINE
TST (R3)+ ;REPEAT COUNTER TO R3
7$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
8$: JSR PC,SPC ;SEND SPACE
DEC R1 ;DECREASE COLUMN COUNT
BEQ 9$ ;BRANCH IF 0, END OF LINE
MOV #100,R0 ;"Q" TO R0
PRINTC ;SEND IT
DEC R1 ;DECREASE COLUMN COUNT
BNE 8$ ;BRANCH IF NOT 0 (NOT END)
9$: CMP #2,R3 ;END OF LINE, FIRST TIME?
BNE 11$ ;BRANCH IF NOT
10$: CR ;SEND CR
DEC R3 ;DECREASE LINE COUNTER
BR 7$ ;REPEAT LINE
11$: TST R3 ;TEST IF THIRD REPEAT
BNE 10$ ;BRANCH IF NOT
CRLF ;DO NEXT LINE
TST (R3)+ ;LINE REPEAT COUNTER TO R3
12$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
13$: MOV #46,R0 ;"8" TO R0
PRINTC ;SEND IT
DEC R1 ;DECREASE COLUMN COUNT
BEQ 14$ ;BRANCH IF END

```


2025	011134	004737	007430		JSR	PC,SPC	:SEND SPACE
2026	011140	005301			DEC	R1	:DECREASE COLUMN COUNT
2027	011142	001367			BNE	13\$:BRANCH IF NOT END
2028	011144	022703	000002	14\$:	CMP	#2,R3	:TEST IF FIRST TIME
2029	011150	001003			BNE	16\$:BRANCH IF =2, FIRST TIME
2030	011152	104022		15 .	CR		:SENT CR
2031	011154	005303			DEC	R3	:DECREASE REPEAT COUNTER
2032	011156	000757			BR	12\$:PRINT LINE AGAIN
2033	011160	005703		16\$:	TST	R3	:TEST IF END, R3=0
2034	011162	001373			BNE	15\$:BRANCH IF NOT END
2035	011164	104012			CRLF		:SEND CR,LF
2036	011166	104005			CHAIN		:CHAIN TO NEXT TEST
2037	011170	000676			BR	1\$:REPEAT TEST

2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086

011172 000010
011174 011330
011176 104016
011200 012701 000036
011204 012702 000170
011210 012737 000036 011226
011216 012700 000110
011222 104015
011224 012700 000036
011230 104010
011232 005301
011234 001426
011236 005302
011240 001430
011242 013704 011226
011246 006204
011250 006204
011252 006204
011254 006204
011256 010405
011260 006204
011262 006204
011264 006204
011266 060405
011270 022702 000074
011274 003403
011276 160537 011226
011302 000745
011304 060537 011226
011310 000742
011312 104012
011314 012701 000036
011320 000746
011322 104012
011324 104005
011326 000724

:XXXXXXXXXX
PT10-- PRINTING FREQUENCY TEST-- 120 H'S ARE PRINTED ON 4 LINES
30 PER LINE. THE TEST IS SUCH THAT BETWEEN THE FIRST AND SECOND
"H" A 30 MSEC DELAY IS INTRODUCED. THIS DELAY IS THEN INCREASED
BETWEEN CHARACTERS OUT TO 60 CHARACTERS IN AN EXPONENTIAL
MANNER. THE DELAY IS THEN DECREASED IN THE SAME MANNER OUT TO THE
120TH CHARACTER. THIS DELAY IS CALCULATED AS FOLLOWS;

NEW DELAY = OLD DELAY [+ OR -] (OLD DELAY/16 + OLD DELAY/128)
:XXXXXXXXXX

PT10: 10 ;TEST NUMBER
PT11 ;NEXT TEST
PRTHDR ;TYPE MESSAGE
1\$: MOV #36,R1 ;SET R1=30
MOV #120,R2 ;SET CHAR COUNT = 120
MOV #30,3\$+2 ;SET UP DELAY VALUE
2\$: MOV #110,R0 ;"H" TO R0
PRINTC ;SEND IT
3\$: MOV #30.,R0 ;DELAY
DELAY ;DEC. COUNT OF CHARS PER LINE
DEC R1 ;BRANCH IF 0, END OF LINE
4\$: BEQ 6\$;DECREMENT CHAR COUNTER
DEC R2 ;BRANCH IF END
BEQ 7\$;GET OLD DELAY
MOV 3\$+2,R4 ;CAL 1/16 OF OLD DELAY
ASR R4
ASR R4
ASR R4
ASR R4
MOV R4,R5 ;SAVE 1/16 IN R5
ASR R4 ;CAL 1/128 OF OLD DELAY
ASR R4
ASR R4
5\$: ADD R4,R5 ;1/16 +1/128 TO R5
CMP #60.,R2 ;TEST WHICH HALF OF THE 120 CHARS.
BLE 5\$;BRANCH IF LT OR EQ 60
SUB R5,3\$+2 ;GT 51(DECREASE DELAY BY 34 MEC.
BR 2\$;GO PRINT AGAIN
5\$: ADD R5,3\$+2 ;LT HALF WAY, ADD DELAY OF 34 MEC.
BR 2\$;GO PRINT AGAIN
6\$: CRLF ;SEND CRLF
MOV #36,R1 ;SET R1=30
BR 4\$
7\$: CRLF ;SEND CR,LF
CHAIN ;CHAIN TO NEXT TEST
BR 1\$;REPEAT TEST

```

2087
2088
2089
2090
2091
2092
2093
2094
2095
2096 011330 000011
2097 011332 011362
2098 011334 104016
2099 011336 012701 000030
2100 011342 012700 000130
2101 011346 104015
2102 011350 104012
2103 011352 005301
2104 011354 001372
2105 011356 104005
2106 011360 000766
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119 011362 000012
2120 011364 007242
2121 011366 104016
2122 011370 012701 000010
2123 011374 012700 000007
2124 011400 104015
2125 011402 005301
2126 011404 001375
2127 011406 104014
2128 011410 012700 003720
2129 011414 104010
2130 011416 013700 000042
2131 011422 001405
2132 011424 000005
2133 011426 004710
2134 011430 000240
2135 011432 000240
2136 011434 000240
2137 011436 104005
2138 011440 000753

```

```

:XXXXXXXXXX
:PT11)- RIBBON FEED TEST-- THIS TEST PRINTS A SINGLE COLUMN OF X'S
:          (24 LINES) DOWN THE LEFT MARGIN OF THE PAGE.
:          VISUALLY CHECK THE RIBBON FEED MECHANISM FOR PROPER OPERATION.
:XXXXXXXXXX

```

```

PT11: 11 ;TEST NUMBER
      PT12 ;NEXT TEST
      PRTHDR ;TYPE MESSAGE
1$: MOV #30,R1 ;SET R1=24(10), LINE COUNT
2 : MOV #130,R0 ;SET CHAR = X
      PRINTC ;PRINT X
      CRLF ;SEND CR-LF
      DEC R1 ;DECREMENT LINE COUNT
      BNE 2$ ;CONTINUE IF NOT DONE TEST
      CHAIN ;CHAIN TO NEXT TEST
      BR 1$ ;REPEAT TEST

```

```

:XXXXXXXXXX
:PT12-- PRINTER BELL TEST-- THE LAST TEST IN THE
:          PRINTER TEST SEQUENCE. THIS TEST OUTPUTS
:          EIGHT BELL SIGNALS TO THE PRINTER
:XXXXXXXXXX

```

```

PT12: 12 ;THIS TEST
      PTO ;NEXT TEST
      PRTHDR ;TYPE HEADER
PT12A: MOV #10,R1 ;COUNTER TO R1
      MOV #7,R0 ;BELL TO R0
1$: PRINTC ;SEND IT
      DEC R1 ;DECREMENT COUNT
      BNE 1$ ;BRANCH IF NOT ZERO
      LF
      MOV #3720,R0 ;DELAY 2 SEC BEFORE RESTARTING
      DELAY
      MOV @#42,R0 ;CHECK IF UNDER ACT11 OR XXDP
      BEQ HERE ;CONTINUE TEST SEQUENCE
      RESET
LOGICAL: JSR PC,(R0)
      NOP
      NOP
      NOP
HERE: CHAIN ;CHAIN TO NEXT TEST
      BR PT12A ;REPEAT TEST

```


.SBTTL LA36 ECHO TESTS

:XXXXXXXXXX

:E020-- CHARACTER ECHO TEST-- ALL PRINTABLE AND
NON-PRINTING CHARACTERS TYPED ON THE KEYBOARD
ARE USED TO DRIVE THE PRINTER, ONE CHARACTER AT
A TIME. A "RUBOUT" WILL CAUSE THE TEST TO BE
TERMINATED.

:XXXXXXXXXX

011606 000020
011610 011656
011612 104016
011614 104020
011616 012700 000036
011622 104010
011624 022737 000177 001176
011632 001405
011634 104017
011636 117777 167250 167252
011644 000763
011646 104027
011650 017060
011652 104035
011654 000757

E020: 20 :TEST NUMBER
E021 :NEXT TEST
PRTHDR :TYPE HEADER
15: READ :GO WAIT FOR KEYBOARD INPUT
MOV #30,R0 :DELAY FOR HALF DUPLEX
DELAY
CMP #177,TEMPCH :CHECK IF RUBOUT
BEQ 25 :BRANCH IF YES
PRNT :NO, CHECK PRINTER READY
MOV# 2TKB,2TPB :READY, ECHO CHARACTER
BR 15
25: TYPEM :PRINT TERMINATION MESSAGE
E0END
CHAIN :CHAIN TO NEXT TEST
BR 15 :REPEAT TEST

:XXXXXXXXXX

:E021-- LINE ECHO TEST, FAST RATE-- THIS TEST WILL
CAUSE THE CONTINUAL PRINTING OF "O" AT THE MAXIMUM
RATE UNTIL EITHER ANOTHER CHARACTER IS SELECTED
BY PRESSING A KEY ON THE KEYBOARD OR TERMINATION BY THE
RUBOUT.

:XXXXXXXXXX

011656 000021
011660 011714
011662 104016
011664 012737 000060 001160
011672 013702 001150
011676 013700 001160
011702 104015
011704 005302
011706 003373
011710 104015
011712 000767

E021: 21 :TEST NUMBER
E022 :NEXT TEST
PRTHDR :TYPE HEADER
E021A: MOV #60,REPT :CHARACTER TO BE REPEATED (O)
1: MOV WIDTH,R2 :SET COLUMN COUNT
25: MOV REPT,R0 :GET CHAR
PRINTC :PRINT CHAR
DEC R2 :DEC COLUMN COUNT
BGT 25 :FINISH LINE
CALF :SEND A CR AND LF
BR 15

011752
011756
011760
011764
011768
011772
011776
012002
012006
012010
012014
012018
012022
012026
012030
012034
012038
012042
012046
012050
012054
012058
012062
012066
012070
012074
012078
012082
012086
012090
012094
012098
012102
012106
012110
012114
012118
012122
012126
012130
012134
012138
012142
012146
012150
012154
012158
012162

052516 020114
047523 020110
052123 020130
052105 020130
047505 020124
047105 020121
041501 020113
042502 020114
051502 020040
052110 020040
043114 020040
052126 020040
043106 020040
051103 020040
047523 020040
044523 020040
046104 020105
041504 020051
041504 020062
041504 020063
041504 020064
040516 020113
054523 020116
052105 020102
040503 020116
046505 020040
052523 020102
051505 020103
051506 020040
051507 020040
051522 020040
051525 020040
050123 020040

: THIS FOLLOWING TABLE IS USED BY TFST EC23
: *****

MONIC: .ASCII /NUL /
.ASCII /SOH /
.ASCII /STX /
.ASCII /ETX /
.ASCII /EOT /
.ASCII /ENQ /
.ASCII /ACK /
.ASCII /BEL /
.ASCII /BS /
.ASCII /HT /
.ASCII /LF /
.ASCII /VT /
.ASCII /FF /
.ASCII /CR /
.ASCII /SO /
.ASCII /SI /
.ASCII /DLE /
.ASCII /DC1 /
.ASCII /DC2 /
.ASCII /DC3 /
.ASCII /DC4 /
.ASCII /NAK /
.ASCII /SYN /
.ASCII /ETB /
.ASCII /CAN /
.ASCII /EM /
.ASCII /SUB /
.ASCII /ESC /
.ASCII /FS /
.ASCII /GS /
.ASCII /RS /
.ASCII /US /
.ASCII /SP /

.EVEN

```

2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315 012166 000023
2316 012170 012510
2317 012172 104016
2318 012174 104020
2319 012176 012700 000036
2320 012202 104010
2321 012204 023727 001176 000041
2322 012212 103015
2323 012214 004737 012350
2324 012220 113700 001176
2325 012224 006300
2326 012226 006300
2327 012230 062700 011762
2328 012234 004737 012426
2329 012240 104000
2330 012242 017107
2331 012244 000753
2332 012246 023727 001176 000177
2333 012254 001421
2334 012256 012701 012500
2335 012262 113721 001176
2336 012266 112721 000040
2337 012272 112721 000040
2338 012276 112721 000040
2339 012302 004737 012350
2340 012306 012700 012500
2341 012312 004737 012426
2342 012316 000750
2343 012320 004737 012350
2344 012324 012700 012504
2345 012330 004737 012426
2306 012334 104000
2347 012336 017107
2349 012340 104007
2349 012342 017060
2350 012344 104005
2351 012346 000712
2352 012350 012702 000003
2353 012354 012701 017111
2354 012360 062701 000003
2355 012364 013700 001202
2356 012370 042700 177770
2357 012374 062700 000060

```

```

:XXXXXXXXXX
E023-- CHARACTER CODE TEST-- ANY CHARACTER SELECTED
WILL BE ECHOED ALONG WITH ITS OCTAL CODE.
A MNEMONIC WILL BE PRINTED INSTEAD OF THE CHARACTER
IF IT IS A NON-PRINTING CHARACTER.
THE PARITY OF THE RECEIVED CODE WILL ALSO BE
INDICATED AS EITHER EVEN OR ODD.
:XXXXXXXXXX
E023: 23 ;TEST NUMBER
E024 ;NEXT TEST
PRTHDR ;TYPE HEADER
1$: READ ;GO WAIT FOR CHARACTER
MOV #30,RO ;DELAY FOR HALF DUPLEX
DELAY
CMP TEMPCH,#41 ;TEST IF CHAR IS PRINTABLE
BHS 3$ ;BRANCH IF IT IS
JSR PC,STRLN ;STORE CODE INTO MESSAGE
MOVB TEMPCH,RO ;GET CODE AGAIN
ASL RO ;MULT BY 2
ASL RO ;MULT BY 4
ADD #MONIC,RO ;ADD ADDR OF MNEMONIC TABLE
JSR PC,MOVNUM ;MOV MNEMONIC TO MESSAGE
2$: TYPE ;TYPE CODE AND MNEMONIC
E023M ;ADDRESS OF MESSAGE
BR 1$ ;GO WAIT FOR NEXT CHARACTER
3$: CMP TEMPCH,#177 ;TEST IF CHAR IS A RUBOUT
BEQ 4$ ;BRANCH IF RUBOUT
MOV #MG24,R1
MOVB TEMPCH,(R1)+
MOVB #40,(R1)+
MOVB #40,(R1)+
MOVB #40,(R1)+
JSR PC,STRLN ;STORE CODE INTO MESSAGE
MOV #MG24,RO ;ADDR OF CHAR INTO RO
JSR PC,MOVNUM ;MOVE CHAR INTO MESSAGE
4$: BR 2$ ;TYPE MESSAGE
JSR PC,STRLN ;RUBOUT, CONVERT AND STOR CODE
MOV #MG25,RO ;ADDR. OF DEL INTO RO
JSR PC,MOVNUM ;MOVE DEL INTO MESSAGE
TYPE ;TYPE MESSAGE
E023M ;ADDR OF MESSAGE
TYPEN
E0END
CHAIN ;CHAIN TO NEXT TEST
BR 1$ ;REPEAT TEST
STPLN: MOV #3,R2 ;COUNT OF 3 TO R2
MOV #LINES,R1 ;ADDR OF MESSG TO R1
ADD #3,R1 ;POINT TO LAST SPACE IN MESSG
1: MOV PCHAR,RO ;MOVE OCTAL CODE TO RO
BIC #177770,RO ;SAVE LS OCTAL CHAR.
ADD #60,RO ;IAKE ASCII

```


2358	012400	110041		MOVW	RO,-(R1)	: MOVE INTO MESH
2359	012402	005302		DEC	R2	: DECREMENT CHAR COUNTER
2360	012404	001407		BFG	2\$: BRANCH IF 3 MOVED
2361	012406	006237	001202	ASR	PCHAR	: NOT THREE, SHIFT NEXT OCTAL
2362	012412	006237	001202	ASR	PCHAR	: CHARACTER TO THE RIGHT
2363	012416	006237	001202	ASR	PCHAR	
2364	012422	000760		SR	1\$: CONVERT AND STORE NEXT CHAR.
2365	012424	000207		RTS	PC	: RETURN TO CALLER
2366	012426	012701	017115	MOVNUM: MOV	#LINESA,R1	: ADDR OF LINES IN R1
2367	012432	012702	000004	MOV	#4,R2	: COUNT OF 4 TO R2
2368	012436	112021		1\$: MOVW	(R0)+,(R1)+	: MOV 4 CHARS TO MESH AREA
2369	012440	005302		DEC	R2	: DECREMENT COUNTER
2370	012442	001375		SNE	1\$: BRANCH IF NOT ALL DONE
2371	012444	105737	001200	TSTB	PARITY	: TEST PARITY FLAG
2372	012450	001003		BNE	2\$: BRANCH IF ODD PARITY
2373	012452	012700	017157	MOV	#EVEN,R0	: SET ADDRESS FOR EVEN PARITY MESH
2374	012456	000402		BR	3\$: CONTINUE
2375	012460	012700	017163	2\$: MOV	#ODD,R0	: SET ADDRESS FOR ODD PARITY MESH
2376	012464	012702	000004	3\$: MOV	#4,R2	: COUNT OF 4 TO R2
2377	012470	112021		4\$: MOVW	(R0)+,(R1)+	: MOVE 4 CHARS TO MESH AREA
2378	012472	005302		DEC	R2	: DECREMENT COUNTER
2379	012474	001375		BNE	4\$: BRANCH IF NOT DONE
2380	012476	000207		RTS	PC	: RETURN
2381						
2382	012500	020040	020040	MG24: .ASCII	/ /	: SAVE CHARACTER CODE
2383						
2384					*EVEN	
2385						
2386	012504	042504	020114	MG25: *ASCII	/DEL /	: MNEMONIC FOR RUBOUT
2387						
2388					.EVEN	

```

2399
2390
2391
2392
2393
2394
2395
2396
2397
2399
2399
2400
2401
2402 012510 000024
2403 012512 013256
2404 012514 104016
2405 012516 005001
2406 012520 012702 012654
2407 012524 104020
2409 012526 012700 000036
2409 012532 104010
2410 012534 022737 000177 001176
2411 012542 001440
2412 012544 022737 000003 001176
2413 012552 001413
2414 012554 020127 000400
2415 012560 103361
2416 012562 113722 001176
2417 012566 005201
2418 012570 104017
2419 012572 113777 001176 166316
2420 012600 000751
2421
2422
2423
2424 012602 020227 012654
2425 012606 001403
2426 012610 113722 001176
2427 012614 104013
2428 012616 012702 012654
2429 012622 021227 000003
2430 012626 001733
2431 012630 112200
2432 012632 020027 000003
2433 012636 001767
2434 012640 104015
2435 012642 000772
2436 012644 104007
2437 012646 017060
2438 012650 104005
2439 012652 000721
2440 012654 000003
2441 012656 000400

```

```

:XXXXXXXXXX
E024-- SELECTED PATTERN ECHO TEST-- SELECT 1 TO 256
CHARACTERS. EACH WILL BE ECHOED
AND STORED UNTIL THE CNTL/C IS SELECTED.
AT THAT TIME ALL CHARACTERS WILL BE PRINTED AS
A CONTINUOUS STRING UNTIL EITHER THE RUBOUT IS
SELECTED TO TERMINATE OR THE CNTL/C IS SELECTED
AGAIN. A TERMINATING CNTL/C FOLLOWED BY ANOTHER
CNTL/C WILL ALWAYS CAUSE THE LAST INPUTTED STRING TO
BE PRINTED. A TERMINATING CNTL/C FOLLOWED BY A CHARACTER OTHER THAN A
RUBOUT WILL CAUSE A NEW STRING TO BE INPUTTED.
:XXXXXXXXXX

```

```

E024: 24 :TEST NUMBER
E025 :NEXT TEST
PRTHDR :TYPE TEST HEADER
E024B: CLR R1 :CLEAR CHARACTER COUNT
MOV #BUFR,R2 :ADDRESS OF BUFFER TO R2
1$: READ :WAIT FOR INPUT
MOV #30.,R0 :DELAY FOR HALF DUPLEX
DELAY
CMP #177,TEMPCH :TEST IF RUBOUT
BEQ TERM :BRANCH IF RUBOUT
CMP #3,TEMPCH :TEST IF CNTL-C
BEQ OUTPUT :BRANCH IF CNTL-C
CMP R1,#256. :YES, CHECK IF CHAR CNT IS EQ. GT 256
BHS 1$ :BRANCH IF YES, IGNORE CHAR
MOVB TEMPCH,(R2)+ :STORE CHAR INTO BUFFER
INC R1 :INCREMENT CHARACTER COUNT
PRINT :CHECK IF PRINTER READY
MOVB TEMPCH,PTPB :ECHO CHAR
BR 1$ :GO WAIT FOR NEXT CHAR

:SECTION TO OUTPUT CONTINUOUS STRING
OUTPUT: CMP R2,#BUFR :CHECK IF POINTER IS AT START OF TABLE
BEQ 1$ :YES, BRANCH
MOVB TEMPCH,(R2)+ :NO, STORE TC IN TABLE
SCRLF :SEND A CR LF
1$: MOV #BUFR,R2 :BUFFER ADDRESS TO R2
CMP (R2),#3 :CHECK IF FIRST CHAR IS TC
BEQ E024B :YES, LOOK FOR INPUT AGAIN
2$: MOVB (R2)+,R0 :GET CHARACTER
CMP R0,#3 :DONE STRING?
BEQ 1$ :YES, RESTART STRING
PRINTC :PRINT CHAR
BR 2$ :CONTINUE
TERM: TYPEM :OUTPUT TERMINATION MESSAGE
ECCEND
CHAIN :CHAIN TO NEXT TEST
BR E024B :REPEAT TEST
BUFR: 3 :INITIALIZE FIRST CHAR AS CNTL-C IN TABLE
.BLKB 256. :256 CHARACTER BUFFER

```

2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479

013256 000025
013250 011606
013262 104016
013264 023727 001150 000101
013272 103427
013274 104007
013276 016733
013300 000402
013302 104000
013304 016733
013306 104020
013310 012700 000036
013314 104010
013316 023727 001176 000040
013324 103770
013326 022737 000177 001176
013334 001410
013336 104017
013340 113777 001176 165550
013346 104013
013350 000754
013352 104007
013354 017033
013356 104007
013360 017060
013362 104005
012364 000737

:XXXXXXXXXXXX

:E025-- BELL ECHO TEST-- A MESSAGE IS PRINTED AND
THE TEST WAITS FOR SOME PRINTABLE CHARACTER
TO BE SELECTED ON THE KEYBOARD (GTO40). THIS
TEST IS VALID ONLY IF THE PAPER WIDTH IS GT 64
COLUMNS. IF LT64 COLUMNS AN ILLEGAL BELL TEST
MESSAGE IS PRINTED.

:XXXXXXXXXXXX

E025: 25 ;TEST NUMBER
E020 ;NEXT TEST HEADER
PRTHDR ;PRINT HEADER
1\$: CMP WIDTH,#101 ;TEST IF COLUMN COUNT IS EQ.GT 64
BLO 4\$;BRANCH IF NOT
TYPEM ;TYPE TEST MSG
E025MA
BR 3\$;WAIT FOR CHAR
2\$: TYPE ;TYPE TEST MSG ON TERM CHAR RCVD ON
E025MA
3\$: READ ;WAIT FOR OPERATOR RESPONSE
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
CMP TEMPCH,#40 ;TEST IF PRINTABLE
BLO 3\$;BRANCH IF NON-PRINTABLE
CMP #177(TEMPCH ;CHECK IF CHAR IS RUBOUT
BEQ 5\$;BRANCH IF YES
PRNT ;CHECK IF PRINTER IS READY
MOV# TEMPCH,#TPB ;PRINT CHAR. (BELL SHOULD SOUND)
SCRLF ;SEND A CRLF
BR 2\$;REPEAT
4\$: TYPEM ;TYPE ERROR MESSAGE
E025MB
5\$: TYPEM ;PRINT TERMINATION
E0END
CHAIN ;EXIT TO NEXT TEST
BR 1\$;REPEAT TEST

```

2480          .SBTTL  LA36 OPTION TESTS
2481
2482          ;XXXXXXXXXXXX
2483          ;TEST30 -    SECONDARY CHARACTER SET OPTION
2484          ;XXXXXXXXXXXX
2485
2486          013366  000030
2487          013370  013374
2488          013372  104016
2489          013374  012704  000010
2490          013400  104007
2491          013402  013464
2492          013404  012702  000177
2493          013410  004737  013476
2494          013414  104007
2495          013416  013471
2496          013420  013702  013546
2497          013424  020227  000377
2498          013430  001403
2499          013432  012700  000016
2500          013436  104015
2501          013440  004737  013476
2502          013444  104012
2503          013446  005304
2504          013450  001353
2505          013452  012700  000017
2506          013456  104015
2507          013460  104005
2508          013462  000744
2509
2510          013464  021417  036461  000  10$:  *ASCIZ  <17>/#1=/
2511          013471  017  031043  000075  20$:  .ASCIZ  <17>/#2=/
2512
2513          013476  010201
2514          013500  042701  177537
2515          013504  013703  001150
2516          013510  162703  000003
2517          013514  010100
2518          013516  104015
2519          013520  005201
2520          013522  020102
2521          013524  001406
2522          013526  005303
2523          013530  001371
2524          013532  104012
2525          013534  013703  001150
2526          013540  000765
2527          013542  104012
2528          013544  000207
2529
2530          013546  000177
2531
          T30SC:  .WORD  177

```

```

1$:  PRTHDR          ;PRINT TEST HEADER
      MOV  #8.,R4    ;SET PASS COUNT
2$:  TYPEM          ;INDICATE PRIMARY SET & SEND "SI"
      10$
      MOV  #177,R2   ;SET END CHAR
      JSR  PC,30$    ;PRINT CHAR SET
      TYPEM          ;INDICATE SECONDARY CHAR SET
      20$
      MOV  T30SC,R2  ;SET CHAR SET LIMIT
      CMP  R2,#377   ;USING 8 BITS INSTEAD OF SI?
      BEQ  3$        ;BRANCH IF YES
      MOV  #16,R0    ;SET SO CHAR
      PRINTC        ;SEND IT
3$:  JSR  PC,30$    ;PRINT CHAR SET
      CRLF          ;BLANK LINE
      DEC  R4        ;DEC PASS COUNT
      BNE  2$        ;FINISH TEST
      MOV  #17,R0    ;SET SO CHAR AGAIN
      PRINTC        ;MAKE SURE ON PRIMARY CHAR SET
      CHAIN        ;NEXT TEST SELECTION OR LOOP
      BR  1$        ;LOOP ON TEST

30$:  MOV  R2,R1     ;GET LIMIT CHAR
      BIC  #177537,R1 ;GET START CHAR
      MOV  WIDTH,R3  ;GET COLUMN COUNT
      SUB  #3,R3     ;SUBTRACT 3
31$:  MOV  R1,R0     ;GET CHAR
      PRINTC        ;PRINT IT
      INC  R1        ;NEXT CHAR
      CMP  R1,R2    ;DONE CHAR SET?
      BEQ  32$      ;EXIT IF DONE
      DEC  R3        ;DEC COLUMN COUNT
      BNE  31$      ;FINISH LINE
      CRLF          ;CR-LF WHEN DONE LINE
      MOV  WIDTH,R3 ;RESET COLUMN COUNT
      BR  31$      ;CONTINUE
32$:  CRLF          ;SEND CR-LF
      RTS  PC       ;RETURN

T30SC: .WORD  177 ;CHAR SET LIMIT
          ;CHANGE TO 377 WHEN USING 9 BIT CHAR SELECTION

```

```

2532          :XXXXXXXXXXXXX
2533          :TEST31 -      SELECTIVE ADDRESSING OPTION
2534          :XXXXXXXXXXXXX
2535
2536 013550 000031 TEST31: 31
2537 013552 013550          TEST31
2538 013554 104012          CRLF
2539 013556 013700 014356 7$:      MOV      34$,RO      ;SEND CR-LF FIRST
2540 013562 104015          PRINTC      ;SET EOT CHAR
2541 013564 104007          TYPEN      ;SEND IT
2542 013566 014206          ICS      ;TRY PRINTING ERROR MMSG
2543 013570 012700 000007          MOV      #7,RO      ;SET BEL CHAR
2544 013574 104015          PRINTC      ;SEND IT
2545 013576 012700 000002          MOV      #2,RO      ;SET STX CHAR
2546 013602 104015          PRINTC      ;SEND IT
2547 013604 104016 014356          PRTHDR      ;PRINT TEST HEADER ON ALL TERMINALS
2548 013606 013700          MOV      34$,RO      ;SET EOT CODE
2549 013612 104015          PRINTC      ;SEND IT
2550 013614 012700 000002          MOV      #2,RO      ;SET STX CODE
2551 013620 104015          PRINTC      ;SEND IT
2552 013622 104007          TYPEN      ;TRY PRINT ERROR MMSG
2553 013624 014206          ICS
2554 013626 012703 014334          MOV      #30$,R3      ;SET TABLE POINTER
2555 013632 005713          1 :      TST      (R3)      ;CHECK TABLE ENTRY
2556 013634 001416          BEQ      2$      ;NEXT PORTION OB TEST IF DONE
2557 013636 013700 014356          MOV      34$,RO      ;SEND EOT CHAR
2558 013642 104015          PRINTC
2559 013644 011300          MOV      (R3),RO      ;SEND GROUP SELECT CHAR
2560 013646 104015          PRINTC
2561 013657 012700 000002          MOV      #2,RO      ;SEND STX CHAR
2562 013654 104015          PRINTC
2563 013656 104007          TYPEN      ;TYPE MMSG
2564 013660 014252          14$
2565 013662 012300          MOV      (R3)+,RO      ;TYPE SELECT CHAR FOR MMSG
2566 013664 104015          PRINTC
2567 013666 104012          CRLF      ;CR-LF
2568 013670 000760          BR      1$      ;CONTINUE
2569 013672 012703 014362          2$:      MOV      #40$,R3      ;SET TABLE ADR
2570 013676 005713          3$:      TST      (R3)      ;CHECK SELECT CHAR
2571 013700 001526          BEQ      4$      ;CONTINUE TEST
2572 013702 013700 014356          MOV      34$,RO      ;SET EOT CODE
2573 013706 104015          PRINTC      ;SEND IT - ALL TERMINALS OFF
2574 013710 011300          MOV      (R3),RO      ;GET UNIQUE SELECT CHAR
2575 013712 104015          PRINTC      ;SEND CHAR
2576 013714 104007          TYPEN      ;TRY PRINTING FRROR MMSG
2577 013716 014206          10$
2578 013720 012700 000002          MOV      #2,RO      ;SET STX CHAR
2579 013724 104015          PRINTC      ;SEND IT
2580 013726 104007          TYPEN
2581 013730 014260          15$
2582 013732 011300          MOV      (R3),RO      ;TYPE SELECT CHAR MMSG
2583 013734 104015          PRINTC      ;PRINT SELECT CHAR FOR MMSG
2584 013736 104012          CRLF
2585 013740 104007          TYPEN      ;PRINT INSTRUCTIONS

```

2586	013742	014277			20\$		
2587	013744	104020			READ		; READ CHAR
2588	013746	012700	000036		MOV	#30.,RC	; DELAY FOR HALF DUPLEX
2599	013752	104010			DELAY		
2590	013754	022737	000177	001176	CMP	#177,TEMPCH	; CHECK CHAR
2591	013762	001021			BNE	5\$; CONTINUE IF NOT RUBOUT
2592	013764	013700	014356		MOV	34\$,RO	; ENABLE ALL LINES
2593	013770	104015			PRINTC		
2594	013772	012700	000007		MOV	#7,RO	
2595	013776	104015			PRINTC		
2596	014000	012700	000002		MOV	#2,RO	
2597	014004	104015			PRINTC		
2598	014006	032737	000001	001132	BIT	#1,CNTLSW	; WHICH CONTROL?
2599	014014	001402			BEQ	6\$	
2600	014016	000137	002336		JMP	TTY1B	; GO TO KYBD CONTROL
2601	014022	000137	002154		JMP	WAITF	; GO TO SW REG CONTROL
2602	014026	104017			PRNT		; READY?
2603	014030	013777	001176	165060	MOV	TEMPCH,@TPB	; ECHO CHAR
2604	014036	104012			CRLF		; SEND CR-LF
2605	014040	013700	014356		MOV	34\$,RO	; SEND EOT
2606	014044	104015			PRINTC		
2607	014046	013700	014360		MOV	35\$,RO	; SEND DUMMY SELECT CHAR
2608	014052	104015			PRINTC		
2609	014054	012700	000002		MOV	#2,RO	; SEND STX CHAR
2610	014060	104015			PRINTC		
2611	014062	104007			TYPEN		; TRY PRINTING ERROR MESSG
2612	014064	014206			10\$		
2613	014066	012700	000003		MOV	#3,RO	; SEND ETX
2614	014072	104015			PRINTC		
2615	014074	011300			MOV	(R3),RO	; SEND UNIQUE SELECT CHAR
2616	014076	104015			PRINTC		
2617	014100	012700	000002		MOV	#2,RO	; SEND STX
2618	014104	104015			PRINTC		
2619	014106	104007			TYPEN		; PRINT MESSG ON SELECTED TERMINAL
2620	014110	014260			15\$		
2621	014112	011300			MOV	(R3),RO	; PRINT SELECT CHAR
2622	014114	104015			PRINTC		
2623	014116	104012			CRLF		; SEND CR-LF
2624	014120	012700	000003		MOV	#3,RO	; SEND ETX
2625	014124	104015			PRINTC		
2626	014126	013700	014360		MOV	35\$,RO	; SEND DUMMY SELECT CHAR
2627	014132	104015			PRINTC		
2628	014134	012700	000002		MOV	#2,RO	; SEND STX
2629	014140	104015			PRINTC		
2630	014142	104007			TYPEN		; PRINT MESSG ON SELECTED TERMINALS
2631	014144	014260			15\$		
2632	014146	012300			MOV	(R3)+,RO	; PRINT SELECT CHAR
2633	014150	104015			PRINTC		
2634	014152	104012			CRLF		; SEND CR-LF
2635	014154	000650			BR	3\$; CONTINUE
2636	014156	013700	014356		MOV	34\$,RO	; ENABLE ALL LINES
2637	014162	104015			PRINTC		; BEFORE EXITING TEST
2638	014164	012700	000007		MOV	#7,RO	
2639	014170	104015			PRINTC		

6\$:
5\$:

4\$:

```

2640 014172 012700 000002      MOV #2,RO
2641 014176 104015      PRINTC
2642 014200 104005      CHAIN
2643 014202 000137 013554      JMP 7$ ;NEXT TEST SELECTION OR LOOP
2644                                     ;LOOP ON TEST
2645 014206 051105 047522 026122 10$: .ASCIZ /ERROR. ALL TERMINALS SHOULD BE OFF/<ACRLF>
2646 014214 040440 046114 052040
2647 014222 051105 044515 040516
2648 014230 051514 051440 047510
2649 014236 046125 020104 042502
2650 014244 047440 043106 000200
2651 014252 051107 052517 020120 14$: .ASCII /GROUP /
2652 014260 042523 042514 052103 15$: .ASCIZ /SELECT CHAR = /
2653 014266 041440 040510 020122
2654 014274 020075 000
2655 014277 124 050131 020105 20$: .ASCIZ /TYPE ANY PRINTABLE CHAR ... /
2656 014304 047101 020131 051120
2657 014312 047111 040524 046102
2658 014320 020105 044103 051101
2659 014326 027040 027056 000040
2660
2661 014334 000107 30$: .EVEN
2662 014336 000000 107 ;GROUP SELECT CHAR TABLE
2663 014340 000000 ;FIRST ZERO = END OF TABLE
2664 014342 000000
2665 014344 000000
2666 014346 000000
2667 014350 000000
2668 014352 000000
2669 014354 000000
2670
2671 014356 000004 34$: 004
2672 014360 000045 35$: 45 ;Deselect CHAR = "EOT"
2673                                     ;DUMMY SELECT CODE = %
2674                                     ;IF % IS USED AS A GROUP OR UNIQUE
2675                                     ;SELECT CODE, REPLACE WITH ASCII
2676                                     ;CODE OF ANY UNUSED CHARACTER.
2677
2677 014362 000125 40$: 125
2678 014364 000000 ;UNIQUE SELECT CHAR TABLE
2679 014366 000000 ;FIRST ZERO - END OF TABLE
2680 014370 000000
2681 014372 000000
2682 014374 000000
2683 014376 000000
2684 014400 000000
2685 014402 000000
2686 014404 000000
2687 014406 000000
2688 014410 000000
2689 014412 000000
2690 014414 000000
2691 014416 000000
2692 014420 000000
2693 014422 000000

```

```

2694 ;XXXXXXXXXXXXX
2695 ;TEST32 - AUTO ANSWER BACK OPTION
2696 ;XXXXXXXXXXXXX
2697
2698 014424 000032 TEST32: 32
2699 014426 014432 1$
2700 014430 104016 PRTHDR ;PRINT TEST HEADER
2701 014432 105777 167556 1$: TSTB @TPSS ;SEND ENQ TO SELECTING TERMINAL
2702 014436 100375 BPL 1$ ;OR CONSOLE TERMINAL
2703 014440 012777 000005 167550 MOV #5,@TPBS
2704 014446 004737 014476 JSR PC,10$ ;READ AND PRINT MMSG
2705 014452 104007 TYPEM ;TYPE INSTRUCTION
2706 014454 014604 20$
2707 014456 004737 014476 JSR PC,10$ ;READ AND PRINT MMSG
2708 014462 104007 TYPEM ;TYPE INSTRUCTION
2709 014464 014631 30$
2710 014466 004737 014476 JSR PC,10$ ;READ & ECHO MMSG
2711 014472 104005 2$: CHAIN ;CHAIN TO NEXT TEST
2712 014474 000756 BR 1$ ;LOOP ON TEST
2713
2714 014476 012702 014655 10$: MOV #STORE,R2 ;SET TABLE ADR
2715 014502 104020 READ ;READ FIRST CHAR
2716 014504 023727 001176 000177 16$: CMP TEMPCH,#177 ;CHAR = RUBOUT?
2717 014512 001010 BNE 11$ ;CONTINUE IF NOT RUBOUT
2718 014514 032737 000001 001132 BIT #1,CNTLSW ;WHICH CONTROL?
2719 014522 001402 BEQ 17$
2720 014524 000137 002336 JMP TTY1B ;GO TO KYBD CONTROL
2721 014530 000137 002154 17$: JMP WAITF ;GO TO SW REG CONTROL
2722 014534 113722 001176 11$: MOVB TEMPCH,(R2)+ ;STORE CHAR
2723 014540 012700 00031C MOV #200,R0 ;SET DELAY COUNTS FOR 200 MSEC.
2724 014544 013701 001170 12$: MOV TIMER,R1
2725 014550 105777 164334 13$: TSTB @TKS ;ANY INPUT?
2726 014554 100411 BMI 14$ ;READ CHAR IF INPUT
2727 014556 005301 DEC R1 ;WAIT FOR CHAR
2728 014560 001373 BNE 13$
2729 014562 005300 DEC R0
2730 014564 001367 BNE 12$
2731 014566 105012 15$: CLRB (R2) ;SET NULL AS TERMINATOR IN TABLE
2732 014570 104000 TYPE ;TYPE MMSG ON TERMINAL RECEIVED ON
2733 014572 014654 STORE-1
2734 014574 104013 SCRLF
2735 014576 000207 RTS PC ;RETURN TO TEST
2736 014600 104025 14$: READC ;READ CHAR
2737 014602 000740 BR 15$ ;CHECK CHAR & CONTINUE
2738
2739 014604 042504 051120 051505 20$: .ASCIZ /DEPRESS HERE IS KEY/<ACRLF>
2740 014612 020123 042510 042522
2741 014620 044440 020123 042513
2742 014626 100131 000
2743 014631 104 050105 042522 30$: .ASCIZ /DEPRESS CONTROL-E/<ACRLF>
2744 014636 051523 041440 047117
2745 014644 051124 046117 042455
2746 014652 000200
2747 014654 200 .ASCII <ACRLF>

```


2748 014655 J00025
2749

STORE: .BLKB 21.
.EVEN

:20 CHAR + TERMINATOR BUFFER

```

XXXXXXXXXXXXX
TEST33 -
XXXXXXXXXXXXX

TEST33: 33
15:      PRTHDR
        MOV      #505,R5
        MOV      #605,R4
        TYPEN
        405
        MOV      #16.,R1
        READ
        MOV      #30.,R0
        DELAY
        CMP      #177,TEMPCH
        BNE      95
        BIT      #1,CNTLSW
        BEQ      115
        JMP      TTY18
        JMP      WAITF
115:     CR
        LF
        DEC      R1
        BNE      25
        MOV      #14,R0
        PRINTC
        CLR      R0
        MOVB     605,R1
        PRINTC
        DEC      R1
        BNE      35
        TYPEN
        455
        MOV      R5,65
        READ
        MOV      #30.,R0
        DELAY
        CMP      #177,TEMPCH
        BNE      95
        BIT      #1,CNTLSW
        BEQ      125
        JMP      TTY18
        JMP      WAITF
12:     CR
        MOV      #14,R0
        PRINTC
        MOVB     (R4)+,R1
        CLR      R0
        PRINTC
        DEC      R1
        BNE      55
        TYPEN
        155

```

```

:XXXXXXXXXXXXX
:TEST33 -
:XXXXXXXXXXXXX

:PRINT TEST HEADER
:SET TABLE POINTER
:SET TABLE POINTER
:PRINT INSTR

:SET LF COUNT TO 16
:WAIT FOR KYBD FLAG
:DELAY FOR HALF DUPLEX

:CHECK FOR RUBOUT
:EXIT IF RUBOUT
:WHICH CONTROL?

:GO TO KYBD CONTROL
:SW REG CONTROL
:SEND CR
:SEND LF
:DEC COUNT
:CONTINUE
:SET FF
:SENT IT
:SET NULL
:SET FILL COUNT
:SEND FILL
:DEC FILL COUNT
:CONTINUE
:TYPE MESSAGE

:SET MESS
:WAIT FOR KYBD FLAG
:DELAY FOR HALF DUPLEX

:CHECK CHAR
:EXIT IF RUBOUT
:WHICH CONTROL?

:SW REG CONTROL
:SEND CR
:SEND FF

:GET FILL COUNT
:SET NULL
:SEND FILL
:DEC COUNT
:CONTINUE
:PRINT MESS

```


015166	026455	026455	000055	15\$:	.ASCIZ	/-----/
015174	020042	043106	027440	20\$:	.ASCIZ	"/ FF (\57)/ /
015202	000040					
015204	020042	042516	052130	30\$:	.ASCIZ	"/ NEXT /
015212	000040					
015214	051120	051505	020123	40\$:	.ASCII	/PRESS TOP OF FORM RESET SWITCH/<ACRLF>/AFTER EACH SWITCH SETTING/<ACRLF
015222	047524	020120	043117			
015230	043040	051117	020115			
015236	042522	042523	020124			
015244	053523	052111	044103			
015252	040600	052106	051105			
015250	042440	041501	020110			
015266	053523	052111	044103			
015274	051440	052105	044524			
015302	043516	200				
015305	124	050131	020105		.ASCII	/TYPE SPACE WHEN READY/<ACRLF>
015312	050123	041501	020105			
015320	044127	047105	051040			
015326	040505	054504	200			
015332	055	026455	020055	45\$:	.ASCIZ	/----- SET 3 INCH FORMFEED -----/
015340	042523	020124	031440			
015346	044440	041516	020110			
015354	047506	046522	042506			
015362	042105	025440	026455			
015370	000055					
015372	000			60\$:	.BYTE	2
015373	000	010	021	.BYTE		5..8..17..20..26..32..35..50..56..58.
015376	024	032	040			
015401	043	062	070			
015404	104					
015406					.EVEN	
015406	020040	000063		50\$:	.ASCIZ	/ 3 /
015412	027063	000065		.ASCIZ		/ 3.5 /
015416	020040	000064		.ASCIZ		/ 4 /
015422	027065	000065		.ASCIZ		/ 5.5 /
015426	020040	000066		.ASCIZ		/ 5 /
015432	020040	000067		.ASCIZ		/ 7 /
015436	020040	000070		.ASCIZ		/ 8 /
015442	027070	000065		.ASCIZ		/ 8.5 /
015446	030440	000061		.ASCIZ		/ 11 /
015452	030440	000062		.ASCIZ		/ 12 /
015456	030440	000064		.ASCIZ		/ 14 /
015462	020040	000040		55\$:	.ASCIZ	/

```

2866 :XXXXXXXXXXXX
2867 :TEST34 - HORIZONTAL TAB OPTION
2868 :XXXXXXXXXXXX
2869
2870 015466 000034 TEST34: 34
2871 015470 015474 1$
2872 015472 104016 PRTHDR ;PRINT TEST HEADER
2873 015474 005004 1$: CLR R4 ;SET TABLE POINTER
2874 015476 012737 015672 015670 2$: MOV #11$,R4+2 ;RESET JUMP INSTR FOR FIRST TIME THRU
2875 015504 013703 001150 2$: MOV WIDTH,R3 ;SET COLUMN COUNT
2876 015510 012700 000033 MOV #33,R0 ;CLEAR OLD TABS
2877 015514 104015 PRINTC
2878 015516 012700 000062 MOV #62,R0
2879 015522 104015 PRINTC
2880 015524 104022 CR ;DO A CARRIAGE RETURN
2881 015526 016401 015760 3$: MOV 20$(R4),R1 ;GET SPACE COUNT FOR TAB
2882 015532 000405 BR 5$ ;SUBTRACT 1 FOR TAB SET
2883 015534 012700 000040 4$: MOV #40,R0 ;GET SPACE
2884 015540 104015 PRINTC ;SEND IT
2885 015542 005303 DEC R3 ;DEC COLUMN COUNT
2886 015544 001420 BEQ 6$ ;CR IF DONE LINE
2887 015546 005301 5$: DEC R1 ;DEC SPACE COUNT
2888 015550 001371 BNE 4$ ;CONTINUE IF NOT DONE
2889 015552 012700 000033 MOV #33,R0 ;SEND IT
2890 015556 104015 PRINTC
2891 015560 012700 000061 MOV #61,R0 ;SEND 1, -SET TAB
2892 015564 104015 PRINTC
2893 015566 012700 000010 MOV #10,R0 ;SET BACKSPACE
2894 015572 104015 PRINTC ;SEND IT
2895 015574 012700 000117 MOV #0,R0 ;PRINT 0
2896 015600 104015 PRINTC
2897 015602 005303 DEC R3 ;DEC COLUMN COUNT
2898 015604 001350 BNE 3$ ;CONTINUE IF NOT DONE LINE
2899 015606 104022 6$: CR ;SEND CR
2900 015610 013703 001150 MOV WIDTH,R3 ;RESET COLUMN COUNT
2901 015614 016401 015760 7$: MOV 20$(R4),R1 ;RESET COLUMN COUNT FOR TAB
2902 015620 020127 000001 CMP R1,#1 ;ADD 1 IF FIRST LINE
2903 015624 001001 BNE 8$
2904 015626 005201 INC R1
2905 015630 016402 016004 8$: MOV 30$(R4),R2 ;SET FILL CHAR COUNT
2906 015634 160103 SUB R1,R3 ;SUBTRACT TAB FROM COLUMN COUNT
2907 015636 002413 BLT 10$ ;BRANCH IF TOO MANY COLUMNS
2908 015640 012700 000011 MOV #11,R0 ;SET TAB
2909 015644 104015 PRINTC ;SEND IT
2910 015646 005000 CLR R0 ;YES, SET NULL
2911 015650 104015 9$: PRINTC ;SEND FILL CHAR
2912 015652 005302 DEC R2 ;DEC FILL COUNT
2913 015654 001375 BNE 9$ ;CONTINUE FILLS
2914 015656 012700 000130 MOV #'X,R0 ;SET X CHARACTER
2915 015662 104015 PRINTC ;SEND IT
2916 015664 000761 BR 8$ ;CONTINUE
2917 015666 000137 015672 10$: JMP J#11$ ;SKIP FOLLOWING AFTER FIRST TIME
2918 015672 012737 015720 015670 11$: MOV #12$,R0+2
2919 015700 104022 CR ;SEND CR

```

```

000000 015702 012700 000130      MOV      #'X,R0      ;PRINT X
000001 015706 104015      PRINTC
000002 015710 013703 001150      MOV      WIDTH,R3   ;RESET COLUMN COUNT
000003 015714 005303      DEC      R3         ;SUBTRACT ONE FOR X
000004 015716 000736      BR       7$        ;CONTINUE
000005 015720 104012      12$:  CRLF        ;SEND CR-LF
000006 015722 005724      TST      (R4)+      ;INC TABLE POINTER
000007 015724 016401 015760      MOV      20$(R4),R1 ;GET COLUMN COUNT FOR TAB
000008 015730 001403      BEQ     13$        ;EXIT IF DONE TABLE (0)
000009 015732 020137 001150      CMP     R1,WIDTH   ;CHECK IF TOO LARGE
000010 015736 101662      BLOS   2$         ;CONTINUE TEST, OK
000011 015740 012700 000033      13$:  MOV      #33,R0 ;CLEAR HORIZONTAL TABS
000012 015744 104015      PRINTC
000013 015746 012700 000062      MOV      #62,R0
000014 015752 104015      PRINTC
000015 015754 104005      CHAIN
000016 015756 000646      BR       1$        ;SELECT TEST OR LOOP
000017 015760 000001 000002 000004 20$:  .WORD 1,2,4,8,16,32,64,128,132,0
000018 015766 000010 000020 000040
000019 015774 000100 000200 000204
000020 016002 000000
000021 016004 000001 000002 000003 30$:  .WORD 1,2,3,5,9,18,36,71,73,0
000022 016012 000005 000011 000022
000023 016020 000044 000107 000111
000024 016026 000000

```

```

2946 ;XXXXXXXXXXXXXXXXXX
2947 ;TEST35 - VERTICAL TAB OPTION
2948 ;XXXXXXXXXXXXXXXXXX
2949
2950 016030 000035 TEST35: 35
2951 016032 016034 15
2952 016034 104016 15: PRTHDR ;PRINT TEST HEADER
2953 016036 104007 TYPEM ;TYPE INSTR
2954 016040 016365 205
2955 016042 104020 READ ;WAIT FOR KYBD FLAG
2956 016044 012700 000036 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
2957 016050 104010 DELAY
2958 016052 022737 000177 001176 CMP #177,TEMPCH ;CHECK CHAR
2959 016060 001505 BEQ B5 ;EXIT IF RUBOUT
2960 016062 005004 CLR R4 ;SET LINE COUNT
2961 016064 012700 000033 MOV #33,R0 ;CLEAR VERTICAL TABS
2962 016070 104015 PRINTC
2963 016072 012700 000064 MOV #64,R0
2964 016076 104015 PRINTC
2965 016100 104007 25: TYPEM ;TYPE REF LINE
2966 016102 016343 155
2967 016104 005204 INC R4 ;INC LINE COUNT
2968 016106 020427 000013 CMP R4,#13 ;CHECK IT
2969 016112 003013 BGT 45 ;BRANCH IF DONE REF.
2970 016114 010401 MOV R4,R1 ;GET LF COUNT
2971 016116 104014 35: LF ;SEND LF
2972 016120 005301 DEC R1 ;DEC COUNT
2973 016122 001375 BNE 35 ;CONTINUE
2974 016124 012700 000033 MOV #33,R0 ;SET TAB FOR THIS LINE
2975 016130 104015 PRINTC
2976 016132 012700 000063 MOV #63,R0
2977 016136 104015 PRINTC
2978 016140 000757 BR 25 ;CONTINUE
2979 016142 012700 000014 45: MOV #14,R0 ;SEND FF
2980 016146 104015 PRINTC
2981 016150 104007 TYPEM ;TYPE MESSG
2982 016152 016457 305
2983 016154 104020 READ ;WAIT FOR KYBD FLAG
2984 016156 012700 000036 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
2985 016162 104010 DELAY
2986 016164 022737 000177 001176 CMP #177,TEMPCH ;CHAR = RUBOUT?
2987 016172 001440 BEQ B5 ;EXIT IF RUBOUT
2988 016174 005004 CLR R4 ;RESET LF COUNT
2989 016176 104007 55: TYPEM ;TYPE REF LINE
2990 016200 016342 105
2991 016202 005204 INC R4 ;INC LINE COUNT
2992 016204 020427 000013 CMP R4,#13 ;CHECK IT
2993 016210 003014 BGT 75 ;BRANCH IF DONE
2994 016212 012700 000013 MOV #13,R0 ;SEND TAB
2995 016216 104015 PRINTC
2996 016220 010401 MOV R4,R1 ;SET FILL COUNT
2997 016222 162701 000020 SUB #16.,R1 ;SUBTRACT 16
2998 016226 003763 BLE 55 ;SKIP NULLS IF COUNT < 0
2999 016230 005000 CLR R0 ;SET NULL CHAR

```



```

3047 .SBTTL MISC. DIAGNOSTIC MESSAGES
3048
3049 016505 200 046417 044501 STARTM: .ASCII <ACRLF><17>/MAINDEC-11-DZLAC-C/<ACRLF>
3050 016512 042116 041505 030455
3051 016520 026461 055104 040514
3052 016526 026503 100103
3053 016532 040514 033063 052040 .ASCII /LA36 TERMINAL DIAGNOSTIC/<ACRLF>
3054 016540 051105 044515 040516
3055 016546 020114 044504 043501
3056 016554 047516 052123 041511
3057 016562 200
3058 016563 104 030514 020061 .ASCIZ /DL11 & KL11 INTERFACE/<ACRLF><12>
3059 016570 020046 046113 030461
3060 016576 044440 052116 051105
3061 016604 040506 042503 005200
3062 016612 000
3063 016613 200 042412 042116 ENDPAS: .ASCII <ACRLF><12>/END OF PASS /
3064 016620 047440 020106 040520
3065 016626 051523 020040
3066 016632 030060 030060 005200 PASMES: .ASCIZ /0000/<ACRLF><12>
3067 016640 000
3068 016641 200 047503 051516 DL11S: .ASCII <ACRLF>/CONSOLE & /
3069 016646 046117 020105 020046
3070 016654 030060 042040 030510 DL11S1: *ASCIZ /00 DL11'S UNDER TEST/<ACRLF><12>
3071 016662 023461 020123 047125
3072 016670 042504 020122 042524
3073 016676 052123 005200 000
3074 016703 200 005017 042524 HDRMSG: .ASCIZ <ACRLF><17><12>+TEST #/
3075 016710 052123 021440 000
3076 016715 060 030060 041440 HDR0: .ASCIZ /000 CKLUMNS/<ACRLF><12>
3077 016722 046117 046525 051516
3078 016730 005200 000
3079 016733 124 050131 020105 E025MA: .ASCII /TYPE ANY PRINTABLE CHARACTER /
3080 016740 047101 020131 051120
3081 016746 047111 040524 046102
3082 016754 020105 044103 051101
3083 016762 041501 042524 020122
3084 016770 047101 020104 044514 .ASCIZ /AND LISTEN FOR BELL...../
3085 016776 052123 047105 043040
3086 017004 051117 041040 046105
3087 017012 027114 027056 027056
3088 017020 027056 027056 027056
3089 017026 027056 027056 000
3090 017033 200 047516 020124 E025MB: .ASCIZ <ACRLF>/NOT ENOUGH COLUMNS/<ACRLF>
3091 017040 047105 052517 044107
3092 017046 041440 046117 046525
3093 017054 051516 000200
3094 017060 042600 044103 020117 ECOEND: .ASCIZ <ACRLF>/ECHO TEST TERMINATED/<ACRLF>
3095 017066 042524 052123 052040
3096 017074 051105 044515 040516
3097 017102 042524 100104 000
3098 017107 040 040 E023M: .ASCII / /
3099 017111 040 020040 040 LINES: .ASCII / /
3100 017115 040 020040 020040 LINESA: .ASCIZ / / ;MSG FOR TEST E024
<ACRLF>

```

K08

3101	017122	020040	100040	000	
3102	017127	200	005017	042523	MSG3: .ASCIZ <ACRLF><17><12>/SELECT TEST NUMBER /
3103	017134	042514	052103	052040	
3104	017142	051505	020124	052516	
3105	017150	041115	051105	020040	
3106	017156	000			
3107	017157	105	042526	116	EVEN: .ASCII /EVEN/
3108	017163	117	042104	040	ODD: .ASCII /ODD /
3109					
3110		000001			.END

ACRLF = 000200	ADTENP 004312	AHEAD = 104021	ATO 005520
ATOX 005522	AT1 005552	AT10 006170	AT11 006216
AT12 005256	AT13 006330	AT14 006410	AT15 006474
AT16 006574	AT17 006632	AT2 005604	AT20 006672
AT21 006754	AT22 007044	AT23 007142	A3 005636
AT4 005570	AT5 005760	AT6 006030	AT7 006120
BIT0 = 000001	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
BRCTR 001162	BTASC= 104023	BUFR 012654	CHAIN = 104005
CHAINN 001666	CHAINY 002010	CHALT = 104006	CHLT 001214
CNTLSW 001132	CNVCTR 004304	CONADD 001102	CONIT 004122
CONSET 004126	CONVEC 001104	COUNT3 001164	CR = 104022
CRBUF 001144	CRLF = 104012	CTRA 001146	CURTST 001174
DELAY = 104010	DIGIT 004306	DISPRE 000174	DLADR 001106
DLCNT 001154	DLY 003650	DL115 016641	DL11S1 016654
ECOEND 017060	EHALT = 104002	EHLT 003560	EMTINT 003176
EMTTAB 003252	ENDPAS 016613	END2 001606	END2A 001574
END3 001536	END4 001610	E020 011606	E021 011656
E021A 011664	E022 011714	E022A 011722	E023 012166
E023M 017107	E024 012510	E024B 012516	E025 013256
EC25MA 016733	E025MB 017033	ERR 003532	ERRHLT 003566
ERROR = 104001	EVEN 017157	FORWD = 104024	FORWDA 004026
FORWDB 004034	FSTDL 001130	HDRMSG 016703	HDRD 016715
HERE 011436	ICTR 001156	INCHK 001206	LEVEL 001152
LF = 104014	LFCNT 001204	LINE2 007626	LINE3 010350
LINES 017111	LINESA 017115	LOGICA 011426	MACHER 000004
MESG3 017127	MG24 012500	MG25 012504	MONIC 011762
MCVNUM 012426	NEXT 002130	NEXT1 002150	NITRSW= 004000
NXTST 001136	ODD 017163	OPEN = 000000	OUTPUT 012602
PARITY 001200	PASCNT 011604	PASMES 016632	PC =%000007
PCHAR 001202	PFAIL 003672	POPSP = 005726	POPSP2= 022626
PRGID 001142	PRGTAB 002776	PRINTC= 104015	PRNT = 104017
PRTHDR= 104016	PRTY4 = 000200	PRTY7 = 000340	PSW = 177776
PT0 007242	PT1 007316	PT10 011172	PT11 011330
PT12 011362	PT12A 011370	PT17 011452	PT17B 011442
PT17D 011470	PT2 007440	PT3 010070	PT4 010210
PT5 010366	PTSAL 010534	PT6 010572	PT7 010760
READ = 104020	READC = 104025	READ1 004424	REF 010320
REPT 001160	RESTRT 003730	RTNNO 001134	R0 =%000000
R1 =%000001	R2 =%000002	R3 =%000003	R4 =%000004
RE =%000005	SAVR6 003726	SCOPSW= 040000	SCOPTR 001140
SCRLF = 104013	SELHLT 001230	SKIP 002104	SP =%000006
SPBOT 001100	SPC 007430	SPCNT 001172	SPSP 007410
SP2 007422	SR 001212	START 001304	STARTM 016505
STARTX 001320	START1 001232	START2 001250	START3 001266
STLSPV 003620	STLSRV 003570	STORE 014655	STPCHV= 104004
STPPA 003636	STPRA 003606	STRDRV= 104003	STRLN 012350
SWREG 000176	TEMP 001210	TEMPCH 001176	TENPWR 004310
TERM 012644	TESTC 002734	TEST30 013366	TEST31 013550
TEST32 014424	TEST33 014702	TEST34 015466	TEST35 016030
TIMER 001170	TKB 001112	TKLVL 001122	TKS 001110
TKVTR 001120	TPB 001116	TPBS 004216	TPLVL 001126

TPS	001114	TPSS	004214	TPVTR	001124	TTYCTL=	104011
TTY1	002252	TTY1B	002336	TYP	003326	TYPE =	104000
TYPEN =	104007	TYPM	003404	T3OSC	013546	WAITF	002154
WIDTH	001150	XCSR	001166	\$AREAD	004052	\$BTASC	004220
\$CR	003446	\$CRLF	003434	\$FORWD	003774	\$LF	003436
\$PRHDR	003456	\$PRT	004526	\$PRTC	004536	\$READ	004324
\$READC	004416	\$SCRLF	003362	.	=	017167	

ERRORS DETECTED: 0

*,DZLACC+DZLACC/SOL
RUN-TIME: 10 19 0 SECONDS
CORE USED: 5K

