

LP25 LP26

LP27 LPO7

LP25,26,27,07 DIAG  
CZLPLF0

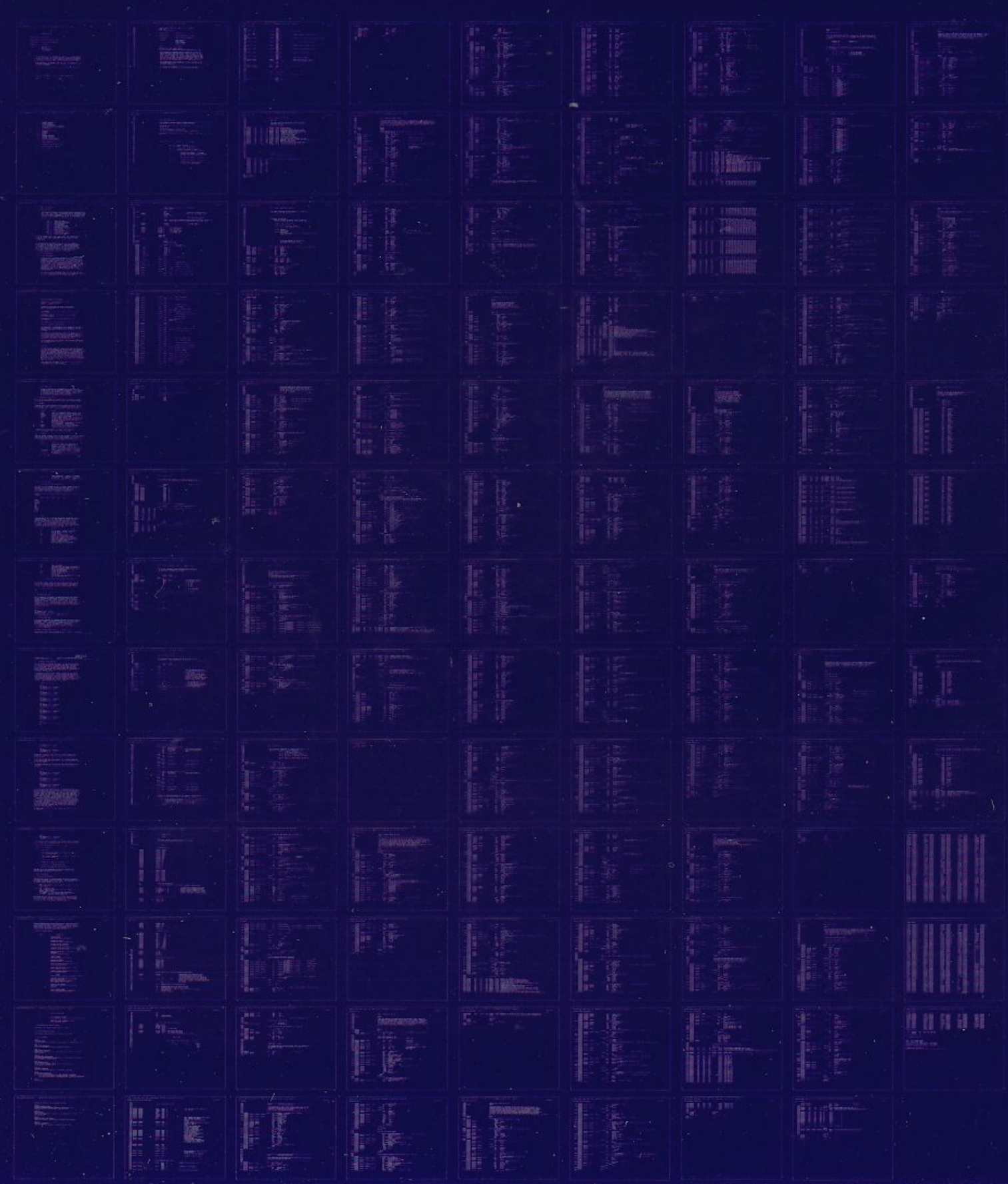
AH-E635F-MC

1 OF 1 OCT 1985

COPYRIGHT© 1979-85

**digital**

MADE IN USA



.REM 8

IDENTIFICATION  
-----

PRODUCT CODE : AC-E634F-MC  
PRODUCT NAME: CZLPLFO LP25/26/27/07 DIAG  
MAINTAINER:    SMALL SYSTEMS DIAGNOSTICS  
PRODUCT DATE: 10-APR-85  
AUTHOR:        JOHN CHATALIAN  
               DON RICE  
               RALPH SCHAUBER  
               GLENN PERNA

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979,1983,1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

## TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

## 1.0 GENERAL INFORMATION

### 1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC PROGRAM VERIFIES PROPER OPERATION OF THE LP25, LP26, LP27 OR LP07 LINE PRINTER AND ITS ASSOCIATED M7258 CONTROL UNIT WHICH INTERFACES TO THE PDP-11 CPU. THE BROAD RANGE OF TESTS ASSURES A COMPREHENSIVE TEST OF THE FUNCTIONAL CAPABILITY OF THE LINE PRINTER. THE INDIVIDUAL TESTS ARE IDENTIFIED AS FOLLOWS:

TEST 1	INTERFACE LOGIC
TEST 2	READY LINE INTERLOCKS
TEST 3	FORMS LENGTH SELECTION
TEST 4	PRINTING SPEED
TEST 5	DATA TRANSFER PATHS
TEST 6	PRINTABLE CHARACTERS
TEST 7	NON-PRINTABLE CHARACTERS
TEST 8	BAND PATTERN
TEST 9	SPURIOUS HAMMER FIRING
TEST 10	PRINT CONTROL
TEST 11	MULTIPLE LINE ADVANCE
TEST 12	CHARACTER ALIGNMENT

ANY MIX OF PRINTER TYPES (LP25, LP26, LP27, LP07) CAN BE TESTED UP TO A TOTAL OF SIXTEEN UNITS. BAND CONFIGURATION (64 OR 96 CHAR) IS HANDELED ON A UNIT BY UNIT BASIS. ALL UNITS NEED NOT HAVE THE SAME BAND.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

### 1.2 SYSTEM REQUIREMENTS

#### 1.2 SYSTEM REQUIREMENTS

A TEST STATION IS REQUIRED CONSISTING OF A PDP-11 CPU WITH A MINIMUM OF 16K WORDS OF MEMORY AND A CONSOLE TERMINAL WITH INTERFACE AT DEVICE ADDRESS 777560. THE SYSTEM ALSO REQUIRES AN XXDP SUPPORTED DEVICE SUCH AS AN RK05/RK11 DISK DRIVE TO AFFORD A MEANS TO LOAD THE DIAGNOSTIC PROGRAM. A KW11-L LINE TIME CLOCK OR A KW11-P PROGRAMMABLE REAL-TIME CLOCK IS NECESSARY FOR MEASURING THE TIME INTERVAL FROM WHICH PRINTING SPEED IS DETERMINED. IF A CLOCK IS NOT INSTALLED IN THE SYSTEM, THE OPERATOR WILL HAVE TO USE MANUAL MODE TO MANUALLY TIME PRINTER OPERATION FOR A FIXED TIME INTERVAL TO CALCULATE THE PRINTING SPEED.

IN A MANUFACTURING ENVIRONMENT WHERE APT/ACT/SLIDE ARE USED, THE TEST STATION MUST BE EQUIPPED WITH THE APPROPRIATE INTERFACE AND A HOST PROCESSOR WITH THE NECESSARY SOFTWARE.

### 1.3 RELATED DOCUMENTS AND STANDARDS

PROJECT PLAN FOR LP25 DIAGNOSTIC PROGRAM  
DOCUMENT: RAS-78-008-00-U  
DATE: 6-SEP-78

DIAGNOSTIC ENGINEERING FUNCTIONAL SPECIFICATION  
FOR CZLPLAO LP25 DIAGNOSTIC PROGRAM (PRELIMINARY)  
DATE: 29-SEP-78

LINE PRINTER, 250 LPM (LP25) PURCHASE SPECIFICATION  
(PRELIMINARY)

DATAPRODUCTS 300 LPM LINE PRINTER FIELD MAINTENANCE  
GUIDE (PRELIMINARY)

DATAPRODUCTS 300 LPM LINE PRINTER OPERATOR'S GUIDE  
(PRELIMINARY)

LINE PRINTER, LP-07 PURCHASE SPECIFICATION (PRELIMINARY)

### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THIS DIAGNOSTIC IS COMPATIBLE WITH ALL MEMBERS OF THE PDP-11  
COMPUTER FAMILY. THE DIAGNOSTIC IS INTERFACED TO THE PDP-11  
DIAGNOSTIC SUPERVISOR THROUGH WHICH IT INTERFACES TO THE  
ENVIRONMENT.

THE DIAGNOSTIC CAN BE USED IN A VARIETY OF OPERATING SYSTEMS  
TO FULFILL DIFFERENT REQUIREMENTS. THE DIAGNOSTIC CAN BE  
LOADED USING XXDP IN A FIELD SERVICE OPERATION, LOADED USING  
THE APT/ACT/SLIDE DIAGNOSTIC MONITORS IN A MANUFACTURING  
ENVIRONMENT, OR MANUALLY LOADED USING PAPER TAPE.

THE APPLICABLE PDP-11 CPU, MEMORY, AND PERIPHERALS SHOULD BE  
RUN TO VALIDATE PROPER OPERATION OF THE SYSTEM BEFORE RUNNING  
THIS DIAGNOSTIC.

### 1.5 ASSUMPTIONS

THE LINE PRINTERS UNDER TEST SHOULD HAVE POWER APPLIED AND BE  
PLACED ON LINE IN READINESS FOR TESTING. EACH LINE PRINTER  
MUST HAVE ITS OWN M7258 CONTROLLER SET UP AT A DIFFERENT DEVICE  
ADDRESS. THE DIAGNOSTIC PROVIDES A DEFAULT DEVICE ADDRESS OF  
777514 WHICH CAN BE USED WHEN A SINGLE LINE PRINTER IS BEING  
TESTED OR FOR THE FIRST UNIT WHEN MULTIPLE LINE PRINTERS ARE  
UNDER TEST. IT WILL BE NECESSARY FOR THE OPERATOR TO RUN THE  
LINE PRINTER OFF LINE IN THE SELF TEST MODE BEFORE RUNNING THE  
DIAGNOSTIC IN ORDER TO DETERMINE WHETHER THE 64 OR 96 CHARACTER  
BAND IS INSTALLED.

FORMS LENGTH MUST BE PRESET TO 11 INCHES. VERTICAL PRINTING  
MUST BE PRESET TO 6 LINES PER INCH.  
IF PRINTER IS LP07, IT MUST BE EQUIPPED WITH SPECIAL DIAG.

## TCVFU PAPER TAPE.

A PATCH IS REQUIRED IN THE DIAGNOSTIC TO CIRCUMVENT AN INCOMPATIBILITY IN THE DIAGNOSTIC SUPERVISOR. IT IS NECESSARY TO ADD 11236 TO THE CONTENTS OF THE ADDRESS "L\$LAST" WHICH IS FOUND AT THE END OF THE ASSEMBLY LISTING. THIS SUM IS USED AS THE ADDRESS INTO WHICH 42760 IS DEPOSITED. 177777 IS DEPOSITED INTO THE SUBSEQUENT MEMORY ADDRESS.

## 2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

## 2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER +C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

## 2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDD	EXECUTE DDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY

/UNITS:LIST            DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)  
 TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED  
 IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12  
 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

### 2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES

BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

\*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

#### 2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

#UNITS (D) ? 1

UNIT 1

LP11 ADDRESS: (0) (177514) ?

INTERRUPT VECTOR : (0) (200) ?

ENTER 0 IF LP25, 1 IF LP26, 2 IF LP07, 3 IF LP27 (0) (0) ?

96 CHARACTER BAND (L) ? ANSWER Y OR N.

#### 2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

RUN MANUAL INTERVENTION TESTS (N) ? DEFAULT IS NO

PERFORM MANUAL PRINT SPEED MEASUREMENTI (N) ? DEFAULT IS AUTOMATIC  
DESIRED TIME INTERVAL FOR PRINT SPEED CALCULATION (60) ? DEFAULT IS 60 SECONDS





```
SUB-DEVICE # (0) ? 6<CR>
Q-FACTOR (0) 0 ? 1<CR>
```

```
UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 1 ? <CR>
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

```
# UNITS (0) ? 8<CR>
```

```
UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0,1<CR>
Q-FACTOR (0) 0 ? 1,0<CR>
```

```
UNIT 3
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 2-5<CR>
Q-FACTOR (0) 0 ? 0<CR>
```

```
UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6,7<CR>
Q-FACTOR (0) 0 ? 1<CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

```
# UNITS (0) ? 8<CR>
```

UNIT 1  
CSR ADDRESS (0) ? 160000<CR>  
SUB-DEVICE # (0) ? 0-7<CR>  
Q-FACTOR (0) 0 ? 0.1.0,...1.1<CR>

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

## 2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

## 3.0 ERROR INFORMATION

### 3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX  
ERROR MESSAGE

WHERE: NAME = DIAGNOSTIC NAME  
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)  
NUMBER = ERROR NUMBER  
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)  
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED  
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

### 3.2 SPECIFIC ERROR MESSAGES

ERROR	DESCRIPTION
1	"PRINTER ERROR" ERROR CONDITION IN THE PRINTER.
2	"PRINTER NOT READY" PRINTER NOT READY TO ACCEPT DATA.
3	"PRINTER DID NOT INTERRUPT" FAILURE IN INTERFACE LOGIC.
4	"LOADING PRINTER BUFFER DOES NOT CLEAR READY" FAILURE IN INTERFACE LOGIC.
5	"PRINTER INTERRUPTED AT SAME LEVEL AS THE PROCESSOR" FAILURE IN INTERFACE LOGIC.
6	"PRINTER ERROR" ERROR CONDITION IN THE PRINTER.
7	"PRINTER NOT READY" PRINTER NOT READY TO ACCEPT DATA.
8	"PAPER LOW INTERLOCK SWITCH FAILURE" FAULTY INTERLOCK SWITCH.
9	"HAMMER BANK INTERLOCK SWITCH FAILURE" FAULTY INTERLOCK SWITCH.
10	"CHARACTER BAND INTERLOCK SWITCH FAILURE" FAULTY INTERLOCK SWITCH.
	"NOTE" ERROR MESSAGES #11 AND #12 HAVE BEEN ELIMINATED
13	"INTERRUPT SERVICING FOR THE FOLLOWING DEVICE DID NOT OCCUR" GLOBAL ERROR INDICATING INTERRUPT FOR DATA TRANSFER DID NOT OCCUR.
14	"PRINTER STATUS ERROR" GLOBAL ERROR INDICATING PRINTER ERROR CONDITION.
15	"OUTPUT TIMEOUT ERROR" GLOBAL ERROR INDICATING TRANSMISSION OF LAST CHARACTER DID NOT OCCUR

WITHIN A GIVEN TIME.

- 16 "VFU INTERLOCK FAILURE"  
FAULTY INTERLOCK SWITCH ON VFU TAPE READER COVER
- 17 "BAND GATE LATCH INTERLOCK FAILURE"  
FAULTY INTERLOCK SWITCH

#### 4.0 PERFORMANCE AND PROGRESS REPORTS

PERFORMANCE AND PROGRESS REPORTS ARE NOT SUPPLIED.

#### 5.0 DEVICE INFORMATION TABLES

DEVICE INFORMATION APPEARS IN THE GLOBAL DATA SECTION.

#### 6.0 TEST SUMMARIES

##### TEST 1

###### INTERFACE LOGIC

VERIFIES OPERATION OF INTERFACE LOGIC BETWEEN THE PRINTER AND THE CPU.

##### TEST 2

###### READY LINE INTERLOCKS

VERIFIES OPERATION OF THE READY INTERLOCK SWITCHES.

##### TEST 3

###### FORMS LENGTH SELECTION

VERIFIES ALL POSITIONS OF THE FORM LENGTH SELECT SWITCH FOR PROPER PAPER MOVEMENT.

##### TEST 4

###### PRINTING SPEED MEASUREMENT

DETERMINES PRINTING SPEED ON THE BASIS OF THE PRINTING TIME INTERVAL AND THE NUMBER OF LINES PRINTED.

##### TEST 5

###### DATA TRANSFER PATHS

CHECKS THE DATA TRANSFER PATHS FROM THE PRINTER OUTPUT TO THE PROCESSOR INTERFACE.

##### TEST 6

###### PRINTABLE CHARACTERS

CHECKS FOR PROPER PRINTING OF ALL PRINTABLE CHARACTERS.

##### TEST 7

###### NON-PRINTABLE CHARACTERS

CHECKS FOR PROPER DETECTION OF ALL NON-PRINTABLE CHARACTERS.

ALSO, ON PRINTERS WITH 64 CHARACTER BANDS, IT CHECKS TO MAKE SURE THAT CODES (140 THRU 177) ARE CONVERTED TO CODES (100 THRU 137).

##### TEST 8

###### BAND PATTERN

PRODUCES AN IMAGE OF THE ENTIRE BAND PATTERN.

TEST 9  
SPURIOUS HAMMER FIRING  
CHECKS FOR SPURIOUS HAMMER FIRINGS BY TAKING NOTE OF ANY  
PRINTING THAT MAY OCCUR OUTSIDE A WEDGE PATTERN.

TEST 10  
PRINT CONTROL  
CHECKS THAT CHARACTERS IN EXCESS OF 132 CHARACTERS ON A LINE  
ARE DISREGARDED.

TEST 11  
CHECKS MULTIPLE LINE ADVANCE  
CHECKS THE MULTIPLE LINE ADVANCE FOR PROPER PAPER MOVEMENT.

TEST 12  
CHARACTER ALIGNMENT  
CHECKS CHARACTER ALIGNMENT BY OVERPRINTING EACH LINE.

```
688 .TITLE CZLPLFO LP25, LP26, LP07, LP27 DIAGNOSTIC
689 .ENABL AMA
690 .SBTTL IDENTIFICATION
691
692 ; PRODUCT CODE: AC-E634F-MC
693
694 ; PRODUCT NAME: CZLPLFO LP25/26/27/07 DIAG
695
696 ; MAINTAINER: SMALL SYSTEMS DIAGNOSTICS
697
698 ; AUTHORS: JOHN CHATALIAN
699 ; DONALD RICE
700 ; RALPH SCHAUBER
701 ; GLENN A. PERNA
702
703 ; DATE 10-APR-85
704
705 ;COPYRIGHT (C) 1979, 1983, 1985 BY
706 ;DIGITAL EQUIPMENT CORPORATION, MAYNARD MASSACHUSSETTS 01754
707 ;
708 ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
709 ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
710 ; SION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY
711 ; OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
712 ; AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM
713 ; AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND
714 ; OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
715 ;
716 ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
717 ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
718 ; EQUIPMENT CORPORATION.
719 ;
720 ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
721 ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

## IDENTIFICATION

```

723
724
725      ;**
726      ; FUNCTIONAL DESCRIPTION
727      ;
728      ; THIS DIAGNOSTIC PROGRAM VERIFIES PROPER OPERATION OF THE LP25, LP26, LP27, LP07
729      ; LINE PRINTER, AND IT'S ASSOCIATED INTERFACE MODULE.
730      ;
731      ; ANY MIX OF LP25, LP26, LP27, LP07 PRINTERS MAY BE TESTED, UP TO A TOTAL OF
732      ; SIXTEEN UNITS.
733      ;
734      ; THE PROGRAM CONSISTS OF TWELVE TESTS,
735      ; THREE OF THE PRINTER TESTS INVOLVE MANUAL INTERVENTION.
736      ;
737      ; THE PROGRAM IS COMPATIBLE TO THE PDP-11 DIAGNOSTIC SUPERVISOR, ACT/SLIDE, AND
738      ; XXDP+.
739      ;
740      ;--
741
742
743      ; VERSION      A-0      27-SEP-79      R. SCHAUBER
744
745
746      ; HISTORY      REV. A-0  INITIAL RELEASE
747      ;              REV-C SUPERVISOR / XXDP+ COMPATABLE
748
749      ;              REV. B-0  DOCUMENTATION CHANGE 29-NOV-79
750      ;              CHANGE INIT CODE TO SET PRIO ON NEW PASS
751      ;
752      ;              REV. C-0      INCLUDE LP26 SUPPORT      13-JUN-80
753      ;              INCLUDE TEST MESSAGES TO THE PRINTERS
754      ;              INCLUDES LINE CLOCK SUPPORT FOR LSI-11
755      ;
756      ;              REV. D-0      INCLUDE LP07 SUPPORT      12-JAN-81
757      ;
758      ;              REV. E-0      INCLUDE LP27 SUPPORT      7-APR-83
759      ;
760      ;              REV. F-0      MODIFIED CODE TO WORK ON PDP 11/73 + 11/83
761      ;              10-APR-85 BY HOWARD MARSHALL
762

```



IDENTIFICATION

```

764          .TITLE CZLPLF0 LP25, LP26, LP27, LP07 TEST
765          .SBTTL PROGRAM HEADER
766
767          .MCALL SVC
768 000000    SVC
769          000005    ;INITIALIZE SUPERVISOR MACROS
770          000005    ;DON'T CALL ALL MACROS. $$$
771          .MCALL STRUCT
772 000000    STRUCT
773          ;STRUCTURED MACRO PACKAGE
774          .MCALL SELECT,LET,IF,PUSH,DECR,INCR,WHILE,ELSE,POP,UNTIL,INLINE ;$$$
775          .MCALL IFCOND,ENDINC,ENDSELECT,ENDDO,ENDDECR,CASE,ENDIF,REPEAT ;$$$
776
777          000000    $LSTIN= 0 ; LIST ASSY CODE LEFT
778          000000    $LSTTAG= 0 ; LIST TAGS LEFT
779          177777    $LOCTAG= -1
780
781          000000    SVCINS= 0 ;LIST INSTRUCTIONS
782          000000    SVCTST= 0 ;LIST TEST TAGS
783          000000    SVCSUB= 0 ;LIST SUBTEST TAGS
784          000000    SVCGBL= 0 ;LIST GLOBAL TAGS
785          000000    SVCTAG= 0 ;LIST OTHER TAGS
786
787          .ENABL AMA
788 000000    .ENABL ABS
789          .ENABL LC
790          002000    . =2000
791
792          002000    BGNMOD
793          002000    POINTER BGNSW,BGNSFT
794
795          002000    HEADER CZLPL,F,0,60,1,340
          002000    L$NAME:: ;DIAGNOSTIC NAME
          002000    103 .ASCII /C/
          002001    132 .ASCII /Z/
          002002    114 .ASCII /L/
          002003    120 .ASCII /P/
          002004    114 .ASCII /L/
          002005    000 .BYTE 0
          002006    000 .BYTE 0
          002007    000 .BYTE 0
          002010    L$REV:: ;REVISION LEVEL
          002010    106 .ASCII /F/
          002011    L$DEPO:: ;0
          002011    060 .ASCII /0/
          002012    L$UNIT:: ;NUMBER OF UNITS
          002012    000000 .WORD 0
          002014    L$TIML:: ;LONGEST TEST TIME
          002014    000060 .WORD 60
          002016    L$HPCP:: ;PTR. TO H.W. QUES.
          002016    037646 .WORD L$HARD
          002020    L$SPCP:: ;PTR. TO S.W. QUES.
          002020    040060 .WORD L$SOFT
          002022    L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
          002022    002250 .WORD L$HW
          002024    L$SPTP:: ;PTR. TO S.W. PTABLE
          002024    002266 .WORD L$SW
    
```

## PROGRAM HEADER

002026		L\$LADP::			;DIAG. END ADDRESS
002026	040436	.WORD	L\$LAST		
002030		L\$STA::			;RESERVED FOR APT STATS
002030	000000	.WORD	0		
002032		L\$CO::			
002032	000000	.WORD	0		
002034		L\$DTYP::			;DIAGNOSTIC TYPE
002034	000001	.WORD	1		
002036		L\$APT::			;APT EXPANSION
002036	000000	.WORD	0		
002040		L\$DTP::			;PTR. TO DISPATCH TABLE
002040	002132	.WORD	L\$DISPATCH		
002042		L\$PRIO::			;DIAGNOSTIC RUN PRIORITY
002042	000340	.WORD	340		
002044		L\$ENVI::			;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000	.WORD	0		
002046		L\$EXP1::			;EXPANSION WORD
002046	000000	.WORD	0		
002050		L\$MREV::			;SVC REV AND EDIT #
002050	003	.BYTE	C\$REVISION		
002051	003	.BYTE	C\$EDIT		
002052		L\$EF::			;DIAG. EVENT FLAGS
002052	000000	.WORD	0		
002054	000000	.WORD	0		
002056		L\$SPC::			
002056	000000	.WORD	0		
002060		L\$DEVP::			; POINTER TO DEVICE TYPE LIST
002060	002222	.WORD	L\$DVTYP		
002062		L\$REPP::			;PTR. TO REPORT CODE
002062	000000	.WORD	0		
002064		L\$EXP4::			
002064	000000	.WORD	0		
002066		L\$EXP5::			
002066	000000	.WORD	0		
002070		L\$AUT::			;PTR. TO ADD UNIT CODE
002070	000000	.WORD	0		
002072		L\$DUT::			;PTR. TO DROP UNIT CODE
002072	000000	.WORD	0		
002074		L\$LUN::			;LUN FOR EXERCISERS TO FILL
002074	000000	.WORD	0		
002076		L\$DESP::			;POINTER TO DIAG. DESCRIPTION
002076	002162	.WORD	L\$DESC		
002100		L\$LOAD::			;GENERATE SPECIAL AUTOLOAD EMT
002100	104035	EMT	E\$LOAD		
002102		L\$ETP::			;POINTER TO ERR_TBL
002102	000000	.WORD	0		
002104		L\$ICP::			;PTR. TO INIT CODE
002104	006016	.WORD	L\$INIT		
002106		L\$CCP::			;PTR. TO CLEAN-UP CODE
002106	010154	.WORD	L\$CLEAN		
002110		L\$ACP::			;PTR. TO AUTO CODE
002110	002260	.WORD	L\$AUTO		
002112		L\$PRT::			;PTR. TO PROTECT TABLE
002112	002122	.WORD	L\$PROT		
002114		L\$TEST::			;TEST NUMBER
002114	000000	.WORD	0		
002116		L\$DLY::			;DELAY COUNT

PROGRAM HEADER

002116 000000  
002120  
002120 000000  
796  
797  
798  
799  
800 002122  
002122  
801 002122 000000  
802 002124 177777  
803 002126 177777  
804 002130

L\$HIME:: .WORD 0 ;PTR. TO HIGH MEM  
.WORD 0  
;  
; THE FOLLOWING IS A LOAD PROTECTION TABLE  
;  
BGNPROT  
L\$PROT:: .WORD 0  
.WORD -1  
.WORD -1  
ENDPROT

DISPATCH TABLE

806  
807  
808  
809  
810  
811  
812  
813  
  
814  
815  
816  
817  
  
818  
  
819  
820  
821

.SBTTL DISPATCH TABLE

```

; **
; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
; --
    
```

```

002130
002130 000014
002132
002132 010350
002134 011640
002136 014612
002140 017720
002142 026006
002144 026432
002146 027330
002150 031050
002152 033756
002154 034434
002156 036002
002160 036460
    
```

```

DISPATCH      12      ;X= NUMBER OF TESTS
.WORD 12
L$DISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11
.WORD T12
    
```

```

;
; FOR USE ON REVISION C OF THE SUPERVISOR
;
    
```

```

002162
002162 103 132 114
002165 120 114 106
002170 060 040 114
002173 111 116 105
002176 040 120 122
002201 111 116 124
002204 105 122 040
002207 104 111 101
002212 107 116 117
002215 123 124 111
002220 103 000
    
```

```

DESCRIP      <CZLPLFO LINE PRINTER DIAGNOSTIC>
L$DESC::
.ASCIZ /CZLPLFO LINE PRINTER DIAGNOSTIC/
    
```

```

002222
002222 114 120 062
002225 065 054 114
002230 120 062 066
002233 054 114 120
002236 062 067 054
002241 114 120 060
002244 067 000
    
```

```

.EVEN
DEV TYP      <LP25,LP26,LP27,LP07>
L$DVTYP::
.ASCIZ *LP25,LP26,LP27,LP07*
    
```

.EVEN

## DEFAULT HARDWARE P-TABLE

```

823          .SBTTL  DEFAULT HARDWARE P-TABLE
824
825          ;++
826          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
827          ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
828          ; IS IDENTICAL TO THE RUN-TIME P-TABLE.
829          ;--
830
831 002246          BGNHW  DFPTBL
      002246 000004  .WORD  L10001-L$HW/2
      002250          L$HW::
      002250          DFPTBL::
832 002250 177514  .WORD  177514          ;LP25 REGISTER ADDRESS
833 002252 000200  .WORD  200          ;LP25 INTERRUPT VECTOR
834 002254 000000  .WORD  0          ; 0 IF LP25
835          ; 1 IF LP26
836          ; 2 IF LP07
837          ; 3 IF LP27
838
839 002256 000000  .WORD  0          ; 0 IF 64 CHAR BAND
840          ; 1 IF 96 CHAR BAND
841          ;
842          ;INTERRUPT VECTOR PRIORITY IS 4 AND CANNOT BE CHANGED
843
844
845 002260          ENDNHW
      002260          L10001:
846
847
848
849 002260          BGNAUTO
      002260          L$AUTO::
850
851 002260 000240  NOP          ; NOT USED
852
853 002262          ENDAUTO
      002262          L10002:
      002262 104461  TRAP  C$AUTO

```

## SOFTWARE P-TABLE

```

855      .SBTTL  SOFTWARE P-TABLE
856
857      ;++
858      ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
859      ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
860      ;--
861
862      002264      BGNSW  SFPTBL
                   002264      .WORD  L10003-L$SW/2
                   002266
                   002266      L$SW::
863      SFPTBL::
864      002266      000000      INHINT: .WORD  0
865
866      002270      000000      MANSPP: .WORD  0
867
868      002272      000074      PERIOD: .WORD  60.
869
870
871
872
873
874      002274      000001      USA:      .WORD  1
875
876
877      002276      000005      MAXERR: .WORD  5
878      ; IF ERROR COUNT EXCEEDS MAXERR THE UNIT WILL BE DROPPED FROM TEST
879
880      002300
                   002300
881

```

; 0 IF NO INTERVENTION TESTS  
; 1 IF MANUAL INTERVENTION TESTS  
; DEFAULT IS NO  
; 0 FOR AUTOMATIC PRINT SPEED  
; 1 FOR MANUAL PRINT SPEED TEST  
; AUTOMATIC DEFAULT VALUE  
; OPERATOR TO SELECT TIMING VALUE  
; FROM 4 TO 60 SECONDS. INITIAL  
; DEFAULT VALUE IS 60 SECONDS.  
; 1 FOR TESTING IN U.S.A.  
; 0 FOR TESTING IN G.B./EUROPE  
; \* DIFFERENT BAND PATTERNS \*  
; AUTODROP ERROR COUNT

ENDSW  
L10003:



## GLOBAL AREAS

935  
 936  
 937 002300  
 938  
 939  
 940  
 941  
 942  
 943  
 947 002300

.SBTTL GLOBAL AREAS

BGNMOD

```

;
; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES
; THAT ARE USED IN MORE THAN ONE TEST.
;

```

EQUALS

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

; BIT DIFINITIONS

```

;
BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

```

```

;
BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

```

; EVENT FLAG DEFINITIONS

; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040  
 000037  
 000036  
 000035  
 000034

```

;
EF.START== 32. ; BIT POSITION IN SECOND STATUS WORD
EF.RESTART== 31. ; (100000) START COMMAND WAS ISSUED
EF.CONTINUE== 30. ; (040000) RESTART COMMAND WAS ISSUED
EF.NEW== 29. ; (020000) CONTINUE COMMAND WAS ISSUED
EF.PWR== 28. ; (010000) A NEW PASS HAS BEEN STARTED
; ; (004000) A POWER-FAIL/POWER-UP OCCURRED

```

; PRIORITY LEVEL DEFINITIONS

000340  
 000300  
 000240

```

;
PRI07== 340
PRI06== 300
PRI05== 240

```



## GLOBAL AREAS

```

000200          PRI04== 200
000140          PRI03== 140
000100          PRI02== 100
000040          PRI01== 40
000000          PRI00== 0
;
;OPERATOR FLAG BITS
;
000004          EVL==      4
000010          LOT==     10
000020          ADR==     20
000040          IDU==     40
000100          ISR==    100
000200          UAM==    200
000400          BOE==    400
001000          PNT==   1000
002000          PRI==   2000
004000          IXE==   4000
010000          IBE==  10000
020000          IER==  20000
040000          LOE==  40000
100000          HOE== 100000

948
952          000012          LF==12
953          000014          FF==14
954          000015          CR==15
955          000177          DEL==177
956
957          ;
958          ;PRIORITY LEVEL DEFINITIONS
959          ;
959          000340          PRI07== 340
960          000300          PRI06== 300
961          000240          PRI05== 240
962          000200          PRI04== 200
963          000140          PRI03== 140
964          000100          PRI02== 100
965          000040          PRI01== 40
966          000000          PRI00== 0
967
968          ;
969          ;GLOBAL ERROR CODES FOR USE BY GENERAL ERROR ROUTINE
970          ;
971          000001          STATER= 1          ;TRANSMITTER STATUS ERROR IN OUTPUT
972          000002          TIMOUT= 2         ;TIMEOUT ERROR IN IO DRIVER MODULE
973
974
975          000003          NOINTR= 3          ;THIS ERROR INDICATES THE LAST CHARACTER
976
977
978
979          ;
980          .SBTTL GENERAL REGISTER USAGE DEFINITIONS
981          ;
981          ;R0 RESERVED FOR USE BY THE MACRO PACKAGES
982          ;R1 MAXIMUM NUMBER OF UNITS TO TEST L$UNIT-1
983          ;R2 UNIT NUMBER BY 2. USED TO CALCULATE OFFSET INTO PROPER
984          ; PRINTER TABLE
985          ;R3 TEMPORARY STORAGE

```

GENERAL REGISTER USAGE DEFINITIONS

```

986          ;R4          "      "
987          ;R5
988          ;R6      STACK POINTER
989          ;R7      PROGRAM COUNTER
990          ;
991
992
993          ;
994          ; LP STATUS TABLE BIT DEFINITIONS
995          ;
996          100000      ERROR = BIT15
997          040000      DROPED = BIT14
998          020000      ACTIVE = BIT13
999          010000      FLAG96 = BIT12 ; 96 CHAR BAND
1000          ;BIT11
1001          002000      FLAG07 = BIT10 ;SEE DEVICE CODE BELOW
1002          001000      FLAG26 = BIT9  ;SEE DEVICE CODE BELOW
1003          000377      LOBYTE = 377  ; BIT MASK FOR CLEARING LOBYTE (COUNTER)
1004
1005
1006          ;
1007          ;
1008          ;
1009          ;
1010          ;
1011          ;
1012          ;
1013          ;
1014          ;
1015          ;

```

DEVICE CODE

BITS 9 AND 10 ARE A DEVICE CODE . SUCH THAT:

BIT 10	BIT 9	DEVICE
0	0	LP25
0	1	LP26
1	0	LP07
1	1	LP27

## GENERAL REGISTER USAGE DEFINITIONS

```

1017
1018
1019
1020
1021
1022 002300 000000      FLAG:  .WORD  0
1023 002302 000000      LINCNT: .WORD  0
1024 002304 000000      LSTCNT: .WORD  0
1025 002306 000000      COUNT:  .WORD  0
1026 002310 000000      CCNT:   .WORD  0
1027 002312 000000      STRCNT: .WORD  0
1028 002314 000000      CHRGEN: .WORD  0
1029 002316 000000      UNIT:   .WORD  0
1030 002320 000000      LUNIT:  .WORD  0
1031
1032
1033 002322 000000      PTABAD: .WORD  0
1034 002324 000000      PRINTR: .WORD  0
1035
1036 002326 000000      CLKTYP: .WORD  0
1037
1038
1039
1040 002330 000000      CLOCKP: .WORD  0
1041 002332 000000      CLKCSR: .WORD  0
1042 002334 000000      CLKSET: .WORD  0
1043 002336 000000      CLKVEC: .WORD  0
1044 002340 000000      CLKENA: .WORD  0
1045 002342 000000      ERRCOD: .WORD  0
1046
1047 002344 000000      ERRFLG: .WORD  0
1048 002346 000000      UUT:    .WORD  0
1049
1050
1051
1052 002350 000000      INDEX:  .WORD  0
1053 002352 000000      VFUCMD: .WORD  0
1054
1055
1056
1057 002354 000000      BUFADD: .WORD  0
1058
1059 002356 000000      BUFCNT: .WORD  0
1060
1061 002360 000000      BUFREP: .WORD  0
1062
1063
1064
1065
1066
1067 002362 000020      LPCSR:  .REPT 16.
1068
1069
1070 002422 000016      LPVEC:  .REPT 16
1071
1072
1073 002456 000020      LPBUF:  .REPT 16.

```

.SBTTL GLOBAL DATA SECTION

```

; <CR> FLAG FOR USE BY SUPERVISOR
; LINE COUNTER

; UNIT COUNTER FOR SINGLE UNIT TESTING
; UNIT COUNTER FOR ERRORS
; AND TESTS NOT USING THE OUTPUT
; MACROS.
; P-TABLE ADDRESS RETURNED BY GPHARD
; SELECTED LINE NO.
; MACRO
; CLOCK TYPE CONTROL WORD
; 1= NO CLOCK AVAILABLE
; 2= KW11-L LINE CLOCK
; 3= KW11-P PROGRAMABLE CLOCK
; CLOCK P-TABLE ADDRESS
; CLOCK CSR ADDRESS
; CLOCK TIME SET REG ADDRESS
; CLOCK VECTOR ADDRESS
; CLOCK ENABLE BITS
; ERROR CODE TYPE FOR GENERAL
; ERROR ROUTINE
; EXPECTED ERROR INDICATOR
; # UNITS ACTUALLY UNDER TEST
; EXITS BACK TO IO DRIVER EQUAL
; 1 IF ERROR WAS EXPECTED.

; BUFFER ADDRESS OF DATA TO BE SENT
; TO THE PRINTER
; NUMBER OF BYTES TO TRANSFER

; NUMBER OF TIMES TO PRINT

; LP25 PARAMETER WORD TABLES
;
; ADDRESS OF CSR FOR EACH LP11
; INTERRUPT VECTOR ADDRESS
; DATA BUFFER REGISTER ADDRESS

```

## GLOBAL DATA SECTION

```

1074          .WORD 0
1075          .ENDR
1076 002516 000020 STATUS: .REPT 16.          ; UNIT STATUS
1077          .WORD 0
1078          .ENDR
1079 002556 000020 CURADD: .REPT 16.          ; CURRENT ADDRESS OF OUTPUT DATA BYTE
1080          .WORD 0
1081          .ENDR
1082 002616 000020 MSGCNT: .REPT 16.          ; INITIAL BYTE COUNT OF MSG FOR REPEAT RESTORE
1083          .WORD 0
1084          .ENDR
1085 002656 000020 REPCNT: .REPT 16.          ; NO. OF TIMES TO REPEAT MESSAGE
1086          .WORD 0
1087          .ENDR
1088 002716 000020 MSGADR: .REPT 16.          ; ADDRESS OF DATA TO PRINT START OF DATA
1089          .WORD 0
1090          .ENDR
1091 002756 000020 CURCNT: .REPT 16.          ; CURRENT COUNT REMAINING TO OUTPUT
1092          .WORD -1
1093          .ENDR
1094 003016 000020 LPINTR: .REPT 16.          ; INTERRUPT ROUTINE ADDRESS
1095          .WORD 0
1096          .ENDR
1097 0C3056 000020 DELCNT: .REPT 16.          ; TIMEOUT DELAY COUNTER
1098          .WORD 0
1099          .ENDR
1100 003116 000000 ERRSVC: .WORD 0          ; ERROR ROUTINE DISPATCH ADDRESS
1101 003120 000020 ERRTBL:: .REPT 16.          ; ERROR COUNT FOR EACH UNIT
1102          .WORD 0
1103          .ENDR
1104
1105 003160 000000 WORK:: .WORD 0          ; WORK AREA
1106 003162 000000 WORK1: .WORD 0
1107
1108
1109
1110          .SBTTL OUTPUT BUFFER
1111          ;
1112          ;150 BYTES IS RESERVED FOR THE OUTPUT BUFFER AREA
1113          ;
1114
1115
1116 003164 000226 OUTBUF: .EVEN
1117          .REPT 150.
1118          .BYTE 0
1119          .ENDR

```

GLOBAL TEXT SECTION

```

1121          .SBTTL GLOBAL TEXT SECTION
1122
1123          .NLIST BEX
1124          ;**
1125          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1126          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1127          ; MORE THAN ONE TEST.
1128          ;--
1129 003412      120      122      111 CSRERR: .ASCIZ /PRINTER ERROR/
1130 003430      120      122      111 RDYERR: .ASCIZ /PRINTER NOT READY/
1131 003452      120      101      120 PAPSWI: .ASCIZ /PAPER LOW INTERLOCK SWITCH FAILURE/
1132 003515      110      101      115 BNKSWI: .ASCIZ /HAMMER BANK INTERLOCK SWITCH FAILURE/
1133 003562      103      110      101 BNSWI: .ASCIZ /CHARACTER BAND INTERLOCK SWITCH FAILURE/
1134 003632      124      122      101 INTER1: .ASCIZ /TRANSMIT INTERRUPT TIMEOUT/
1135 003665      120      122      111 TXERR: .ASCIZ /PRINTER STATUS ERROR/
1136 003712      117      125      124 OUTTIM: .ASCIZ /OUTPUT TIMEOUT ERROR/
1137 003737      125      116      111 TXNOIN: .ASCIZ /UNIT FAILED TO INTERRUPT/
1138 003770      101      114      114 UUTEQ0: .ASCIZ /ALL UNITS HAVE BEEN DROPPED..RESTART../
1139 004037      045      116      045 VFUSEL: .ASCII /%N%AINSURE THAT VFU-FLS SWITCH ON EACH UNIT IS IN THE /
1140 004125      045      116      045 VFUSE1: .ASCIZ /%N%A"VFU" POSITION.%N/
1141 004153      116      117      040 NOCLK: .ASCIZ /NO CLOCK AVAILABLE FOR TIMING TESTS/<7><7>
1142 004221      102      101      116 BGTSWI: .ASCIZ /BAND GATE LATCH INTERLOCK FAILURE/
1143 004263      126      106      125 VFUINF: .ASCIZ /VFU INTERLOCK FAILURE/
1144          .EVEN
1145
1146
1147          ;
1148          ;
1149
1150          .LIST BEX
1151          ;
1152          ; FORMAT STATEMENTS USED IN PRINT CALLS
1153          ;
1154
1155 004312      045      101      114 LPDROP: .ASCIZ /%ALP11 UNIT %D2%A DROPPED FROM TEST%N/
          004315      120      061      061
          004320      040      125      116
          004323      111      124      040
          004326      045      104      062
          004331      045      101      040
          004334      104      122      117
          004337      120      120      105
          004342      104      040      106
          004345      122      117      115
          004350      040      124      105
          004353      123      124      045
          004356      116      000
1156
1157
1158
1159

```

GLOBAL SUBROUTINES SECTION

1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193

.SBTTL GLOBAL SUBROUTINES SECTION

```

; **
; THE GLOBAL SUBROUTINE SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED BY MORE THAN ONE TEST.
; --
    
```

```

; **
; FUNCTIONAL DESCRIPTION:
; SUBROUTINE TO PRINT THE GENERAL ERROR INFORMATION.
; PRINTS THE ERROR MESSAGE IN THE FOLLOWING FORMAT:
;
; "ERROR AT CSR XXXXXX UNIT YY"
;
; WHERE XXXXXX= DEVICE CSR ADDRESS
;             YY= UNIT NUMBER THAT FAILED
;
; CALLING SEQUENCE
;             JSR PC,LPERR
; REQUIRED PARAMETERS
;             ERRCOD MUST BE SET TO ONE OF THE ERROR CODES DESCRIBED
;             UNDER ERROR CODES.
    
```

```

;
; R2 IS USED INTERNAL TO THE ROUTINE.
; THE ROUTINE DOES A SAVE ON R2
; AND RESTORES IT PRIOR TO EXITING.
;
    
```

```

004360 013746 002342
004364 002455
004366 023727 002342 000003
004374 003051
004376 006316
004400 062716 004406
004404 013607

004406
004406 004524
004410 004416
004412 004444
004414 004472

1194
1195 004416
004416
1196 004416 005262 003120
1197 004422
004422 010237 002074
004426 006237 002074
1198 004432
004432 104456
004434 000016
    
```

```

LPERR: SELECT ERRCOD OF 3 VERIFY ;SELECT PROPER MESSAGE FORMAT
MOV ERRCOD, -(SP)
BLT 50005$
CMP ERRCOD, #3
BGT 50005$
ASL (SP)
ADD #50000$, (SP)
MOV @ (SP)+, PC
; DISPATCH TABLE
50000$:
.WORD 50004$ ; CASE 0
.WORD 50003$ ; CASE 1
.WORD 50002$ ; CASE 2
.WORD 50001$ ; CASE 3

CASE 1 ;STATUS ERROR
50003$:
LET ERRTBL(R2) := ERRTBL(R2) + #1
INC ERRTBL(R2)
LET L$LUN := R2 SHIFT -1
MOV R2, L$LUN
ASR L$LUN
ERRHRD 14, TXERR
TRAP C$ERHRD
.WORD 14
    
```

GLOBAL SUBROUTINES SECTION

```

004436 003665          .WORD  TXERR
004440 000000          .WORD  0
1199
1200 004442          CASE 2          ;OUTPUT TIMEOUT ERROR
004442 000430          BR      50006$
004444          50002$:
1201 004444          LET ERRTBL(R2) := ERRTBL(R2) + #1
004444 005262 003120          INC      ERRTBL(R2)
1202 004450          LET L$LUN := R2 SHIFT -1
004450 010237 002074          MOV      R2,L$LUN
004454 006237 002074          ASR      L$LUN
1203 004460          ERRHRD 15,OUTTIM          ;
004460 104456          TRAP   C$ERHRD
004462 000017          .WORD  15
004464 003712          .WORD  OUTTIM
004466 000000          .WORD  0
1204
1205 004470          CASE 3
004470 000415          BR      50006$
004472          50001$:
1206          ; NEVER RECIEVED THE INTERRUPT
1207 004472          LET ERRTBL(R2) := ERRTBL(R2) + #1
004472 005262 003120          INC      ERRTBL(R2)
1208 004476          LET L$LUN := R2 SHIFT -1
004476 010237 002074          MOV      R2,L$LUN
004502 006237 002074          ASR      L$LUN
1209 004506          ERRHRD 16,TXNOIN
004506 104456          TRAP   C$ERHRD
004510 000020          .WORD  16
004512 003737          .WORD  TXNOIN
004514 000000          .WORD  0
1210
1211
1212
1213 004516          ENDSELECT
004516 000402          BR      50006$
004520          50005$:
004520 062706 000002          ADD      #2,SP
004524          50004$:
004524          50006$:
1214
1215 004524          IF ERRTBL(R2) GT MAXERR THEN
004524 026237 003120 002276          CMP      ERRTBL(R2),MAXERR
004532 003402          BLE      50007$
1216 004534          JSR PC,DROPIT          ; MAXIMUM ERROR COUNT EXCEEDED !
1217 004540          ENDIF
004540          50007$:
1218 004540          LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR
004540 042762 100000 002516          BIC      #ERROR,STATUS(R2)
1219 004546          LET ERRCOD := #0
004546 005037 002342          CLR      ERRCOD
1220 004552          LET @LPCSR(R2) := #100          ; CLEAR THE ERROR BIT AND ENABLE INTERRUPTS
004552 012772 000100 002362          MOV      #100,@LPCSR(R2)
1221 004560          RTS      PC          ;AND EXIT
1222
1223
1224          ;-----
          ; BIN2DA          BINARY TO DECIMAL ASCII CONVERSION ROUTINE

```

GLOBAL SUBROUTINES SECTION

```

1225 ;
1226 ; ENTER WITH NUMBER TO BE CONVERTED ON THE STACK
1227 ; FOLLOWED BY THE ADDRESS OF A 5 BYTE BUFFER
1228 ; FOR THE ASCII STRING. 5 DIGITS WILL BE CONVERTED
1229 ; LEADING ZEROES WILL BE CONVERTED TO SPACES.
1230 ; CALL BY JSR PC,BIN2DA
1231 ;-----
1232 004562 BIN2DA: PUSH R4,R5
      004562 010446 MOV R4,-(SP)
      004564 010546 MOV R5,-(SP)
1233 004566 LET R4 := 6(SP) ; GET ADDRESS FOR ASCII STRING
      004566 016604 000006 MOV 6(SP),R4
1234 004572 LET R5 := #TABLDA ; GET ADDRESS OF DECIMAL TABLE
      004572 012705 004754 MOV #TABLDA,R5
1235 004576 LET FLAGDA := #0 ; LEADING ZERO FLAG
      004576 005037 004766 CLR FLAGDA
1236 004602 LET COUNTD := #0
      004602 005037 004770 CLR COUNTD
1237 ; 8.(SP) HAS NUMBER TO BE CONVERTED
1238 004606 DECR DIGITS FROM #4 TO #0 BY #1 ; DO 5 DIGITS
      004606 012737 000004 004772 MOV #4,DIGITS
      004614 000402 BR 50010$
      004616 50011$: DEC DIGITS
      004616 005337 004772 50010$: TST DIGITS
      004622 005737 004772 BLT 50012$
      004626 002435 WHILE 8.(SP) GE (R5) DO ; CREATE A DIGIT
1239 004630 50013$: CMP 8.(SP),(R5)
      004630 026615 000010 BLT 50014$
      004634 002405 LET 8.(SP) := 8.(SP) - (R5)
1240 004636 SUB (R5),8.(SP)
      004636 161566 000010 INC LET COUNTD := COUNTD + #1
1241 004642 INC COUNTD
      004642 005237 004770 BR ENDDO
1242 004646 BR 50013$
      004646 000770 50014$: ; CONVERT DIGIT TO ASCII OR SUPPLY A SPACE
      004650 IF COUNTD GT #0 OR FLAGDA GT #0 THEN
1243 004650 TST COUNTD
1244 004650 005737 004770 BGT 50015$
      004654 003003 TST FLAGDA
      004656 005737 004766 BLE 50016$
      004662 003410 50015$: LET COUNTD := COUNTD SET.BY #60
      004664 #60,COUNTD
1245 004664 052737 000060 004770 MOVB LET (R4)+ :B= COUNTD
      004664 113724 004770 COUNTD,(R4)+
1246 004672 LET FLAGDA := FLAGDA + #1
      004672 113724 004770 INC FLAGDA
1247 004676 INC ELSE
      004676 005237 004766 BR 50017$
1248 004702 BR 50016$: LET (R4)+ :B= #40
      004702 000402 MOVB #40,(R4)+
      004704 112724 000040 ENDIF
1249 004704
1250 004710

```



GLOBAL SUBROUTINES SECTION

1251	004710				50017\$:	
1252	004710	062705	000002		LET R5 := R5 + #2	; DO THE NEXT DIGIT
1253	004714	005037	004770		ADD #2,R5	
1254	004720	000736			LET COUNTD := #0	
1255	004722				CLR COUNTD	
1256	004722	005737	004766		ENDDCCR	
1257	004730	112744	000060		BR 50011\$	
1258	004734				50012\$:	
1259	004734					; IF NUMBER WAS A ZERO PRINT A '0'
1260	004734	016666	000004	000010	IF FLAGDA EQ #0 THEN	
1261	004742	012605			TST FLAGDA	
1262	004742	012604			BNE 50020\$	
1263	004746	062706	000004		LET -(R4) :B= #60	
1264	004752	000207			MOVB #60,-(R4)	
1265	004754	023420	001750	000144	ENDIF	
1266	004762	000012	000001		50020\$:	
1267	004766	000000				; CLEAN UP THE STACK AND EXIT
1268	004770	000000			LET 8.(SP) := 4(SP)	
1269	004772	000000			MOV 4(SP),8.(SP)	
1270					POP R5,R4	
					MOV (SP)+,R5	
					MOV (SP)+,R4	
					LET SP := SP + #4	
					ADD #4,SP	
					RTS PC	
					TABLDA: .WORD	10000.,1000.,100.,10.,1
					FLAGDA: .WORD	0
					COUNTD: .WORD	0
					DIGITS: .WORD	0

## I/O DRIVER

```

1272          .SBTTL  I/O DRIVER
1273
1274          ;
1275          ;
1276          ;++
1277          ;THE I/O DRIVER ROUTINE IS INVOKED BY MEANS OF THE INTERRUPT SYSTEM.
1278          ;CALL TO IT IS JMP  IODRV.
1279          ;RETURN  RTI.
1280          ;ENTER ROUTINE WITH R2 SET UP TO DESIRED UNIT *2. R2 IS USED
1281          ;TO CALCULATE OFFSET INTO PROPER TABLE.
1282          ;R1 EQUALS MAXIMUM NUMBER OF UNITS ON SYSTEM UNDER TEST.
1283          ;
1284          ;--
1285          ;
1286          ; CHECK FOR ERROR FLAG IN STATUS REG.
1287          ;
1288 004774      IODRV:  IF #BIT15 NOTSETIN @LPCSR(R2) THEN
004774 032772      BIT      #BIT15,@LPCSR(R2)
005002 001061      BNE      50021$
1289          ;
1290          ; IF COUNT NOT ZERO SEND NEXT BYTE
1291          ;
1292          IF CURCNT(R2) GT #0 THEN
005004      TST      CURCNT(R2)
005010 003416      BLE      50022$
1293 005012      MOVB   @LPCSR(R2),@LPCSR(R2)
005012 117272      MOVB   @CURADD(R2),@LPCSR(R2)
1294 005020      INC     CURCNT(R2)
005020 005262      INC     CURADD(R2)
1295          ;
1296          ; ENABLE INTERRUPT FOR NEXT BYTE
1297          ;
1298 005024      BIS     STATUS(R2),STATUS(R2) SET.BY #ACTIVE
005024 052762      BIS     #ACTIVE,STATUS(R2)
1299 005032      DEC     CURCNT(R2)
005032 005362      DEC     CURCNT(R2) - #1
1300 005036      BIS     @LPCSR(R2),@LPCSR(R2) SET.BY #100
005036 052772      BIS     #100,@LPCSR(R2)
1301 005044      BR      ELSE
005044 000437      BR      50023$
1302          50022$:
1303          ; CURRENT MSG DONE, IF PRINT COUNT NOT ZERO SEND AGAIN
005046      DEC     REPCNT(R2)
005046 005362      DEC     REPCNT(R2) - #1
1304 005052      IF REPCNT(R2) GT #0 THEN
005052 005762      TST     REPCNT(R2)
005056 003424      BLE     50024$
1305 005060      MOV     MSGADR(R2),CURADD(R2) ; RESTORE THE MSG ADDR
005060 016262      MOV     MSGCNT(R2),CURCNT(R2) ; RESTORE THE BYTE COUNT
1306 005066      MOV     @LPCSR(R2),@LPCSR(R2) ; RESEND THE MESSAGE
005066 016262      MOVB   @CURADD(R2),@LPCSR(R2)
1307 005074      INC     CURCNT(R2) ; BUMP THE POINTER
005074 117272      INC     CURADD(R2) + #1
1308 005102      DEC     CURCNT(R2) ; DROP BYTE COUNT
005102 005262      DEC     CURCNT(R2) - #1
1309 005106      DEC     CURCNT(R2)
005106 005362      DEC     CURCNT(R2)

```

## I/O DRIVER

```

1310 005112          LET STATUS(R2) := STATUS(R2) SET.BY #ACTIVE
      005112 052762 020000 002516      BIS  #ACTIVE,STATUS(R2)
1311 005120          LET @LPCSR(R2) := #100 ; RE-ENABLE INTERRUPTS
      005120 012772 000100 002362      MOV  #100,@LPCSR(R2)
1312 005126          ELSE
      005126 000406          BR    50025$
      005130          50024$:
1313          ; CURRENT MSG DONE, REPEAT COUNT =0
1314          ; CLEAR ACTIVE AND DISABLE INTERRUPTS.
1315 005130          LET STATUS(R2) := STATUS(R2) CLR.BY #ACTIVE
      005130 042762 020000 002516      BIC  #ACTIVE,STATUS(R2)
1316 005136          LET @LPCSR(R2) := #00
      005136 012772 000000 002362      MOV  #00,@LPCSR(R2)
1317 005144          ENDIF
      005144          50025$:
1318 005144          ENDIF
      005144          50023$:
1319 005144          ELSE
      005144 000410          BR    50026$
      005146          50021$:
1320          ; CLEAR ERROR CONDITION, ENABLE INTERRUPTS
1321          ; SET ERROR FLAG
1322 005146          LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
      005146 052762 100000 002516      BIS  #ERROR,STATUS(R2)
1323 005154          LET ERRCOD := #STATER ; STATUS ERROR
      005154 012737 000001 002342      MOV  #STATER,ERRCOD
1324 005162          JSR PC,@ERRSVC
      004777 175730          ; ERROR SERVICE SHOULD CLEAR ERROR BIT AND ENABLE INTR
1325          ENDIF
1326 005166          50026$:
      005166          POP R2
1327 005166          MOV  (SP)+,R2
      005166 012602
1328 005170          RTI
      000002

```

## I/O CONTROL

```

1330      .SBTTL I/O CONTROL
1331      ;++
1332      ;
1333      ; THE I/O CONTROL SUBROUTINE IS A SINGLE ENTRY QUEUE MANAGER.
1334      ; THIS ROUTINE IS INVOKED BY A JSR FROM AN I/O CALL.
1335      ; INPUTS:      PRINTR  -1 FOR ALL TERMINALS
1336      ;              N FOR PRINTER NUMBER 'N'
1337      ;              BUFADD  ADDRESS OF MESSAGE TO PRINT
1338      ;              BUFCNT  BYTE COUNT TO TRANSMIT TO PRINTER
1339      ;
1340      ;              ERRSVC  ADDRESS OF ERROR SERVICE SUBROUTINE
1341      ;              BUFREP  IS NO. OF TIMES TO PRINT THE MSG
1342      ;--
1343
1344      IOCTRL: PUSH R2,R3
1345              MOV     R2,-(SP)
1346              MOV     R3,-(SP)
1347
1348      ;
1349      ; IF PRINTR IS -1 QUE OUTPUT TO ALL PRINTERS SELECTED
1350      ; OTHERWISE TO UNIT NUMBER IN PRINTR.
1351      ;
1352      IF PRINTR EQ #-1 THEN
1353      CMP     PRINTR,#-1
1354      BNE     50027$
1355      LET R3 := L$UNIT
1356      MOV     L$UNIT,R3
1357      LET L$LUN := #0
1358      CLR     L$LUN
1359      ELSE
1360      BR      50030$
1361
1362      50027$:
1363      LET R3 := #1
1364      MOV     #1,R3
1365      LET L$LUN := PRINTR
1366      MOV     PRINTR,L$LUN
1367      ENDIF
1368
1369      50030$:
1370      ;
1371      ; REPEAT TILL R3 = 0
1372      ;
1373      CTLLOP:
1374      IF R3 EQ #0 THEN
1375      TST     R3
1376      BNE     50031$
1377      INLINE <JMP CTLEND>
1378      JMP     CTLEND
1379      ENDIF
1380
1381      50031$:
1382      ;
1383      ; USE R2 AS AN INDEX INTO THE UNIT TABLES
1384      ;
1385      LET R2 := L$LUN SHIFT 1
1386      MOV     L$LUN,R2
1387      ASL     R2
1388      LET ERRCOD := #0
1389      CLR     ERRCOD
1390      ;

```

## I/O CONTROL

```

1369 ; IF THE UNIT HAS BEEN DROPPED SELECT THE NEXT UNIT
1370 ;
1371 005254 ; IF #DROPED NOTSETIN STATUS(R2) THEN
      005254 032762 040000 002516 BIT #DROPED,STATUS(R2)
      005262 001123 BNE 50032$
1372 ;
1373 ;TEST FOR DVC ERROR BIT SET
1374 ;
1375 005264 ; IF #BIT15 SETIN @LPCSR(R2) THEN
      005264 032772 100000 002362 BIT #BIT15,@LPCSR(R2)
      005272 001407 BEQ 50033$
1376 005274 ; LET ERRCOD := #STATER ; STATUS REG ERROR BIT 15 SET IN CSR
      005274 012737 000001 002342 MOV #STATER,ERRCOD
1377 005302 ; LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
      005302 052762 100000 002516 BIS #ERROR,STATUS(R2)
1378 005310 ; ELSE
      005310 000455 BR 50034$
      005312 50033$:
1379 ;
1380 ; MAKE SURE PREVIOUS MSG IS DONE
1381 ;
1382 005312 ; IF CURCNT(R2) GT #0 THEN
      005312 005762 002756 TST CURCNT(R2)
      005316 003452 BLE 50035$
1383 005320 ; IF #ACTIVE NOTSETIN STATUS(R2) THEN
      005320 032762 020000 002516 BIT #ACTIVE,STATUS(R2)
      005326 001004 BNE 50036$
1384 ;
1385 ; OUTPUT WAS QUEUED BUT I/O DRIVER WAS NEVER INVOKED (VIA INTERRUPT)
1386 ;
1387 005330 ; LET ERRCOD := #NOINTR ; NO INTERRUPT
      005330 012737 000003 002342 MOV #NOINTR,ERRCOD
1388 005336 ; ELSE
      005336 000442 BR 50037$
      005340 50036$:
1389 005340 ; WHILE #ACTIVE SETIN STATUS(R2) DO
      005340 032762 020000 002516 50040$: BIT #ACTIVE,STATUS(R2)
      005346 001436 BEQ 50041$
1390 ;
1391 005350 ; LET DELCNT(R2) := #100. ; 20 SEC. DELAY MAX
      005350 012762 000144 003056 MOV #100.,DELCNT(R2)
1392 005356 ; DELAY 2. ; 200MS LOOPS
      005356 012727 000002 MOV #2.,(PC)+
      005362 000000 .WORD 0
      005364 013727 002116 MOV L#DLY,(PC)+
      005370 000000 .WORD 0
      005372 005367 177772 DEC -6(PC)
      005376 001375 BNE .-4
      005400 005367 177756 DEC -22(PC)
      005404 001367 BNE .-20
1393 005406 ; LET DELCNT(R2) := DELCNT(R2) - #1
      005406 005362 003056 DEC DELCNT(R2)
1394 005412 ; IF DELCNT(R2) EQ #0 THEN
      005412 005762 003056 TST DELCNT(R2)
      005416 001011 BNE 50042$
1395 005420 ; LET ERRCOD := #TIMOUT

```

## I/O CONTROL

```

005420 012737 000002 002342      MOV      #TIMOUT,ERRCOD
1396 005426      LET STATUS(R2) := STATUS(R2) CLR.BY #ACTIVE
005426 042762 020000 002516      BIC      #ACTIVE,STATUS(R2)
1397 005434      LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
005434 052762 100000 002516      BIS      #ERROR,STATUS(R2)
1398 005442      ENDIF
005442      50042$:
1399 005442      ENDDO
005442 000736      BR      50040$
005444      50041$:
1400 005444      ENDIF
005444      50037$:
1401 005444      ENDIF
005444      50035$:
1402 005444      ENDIF
005444      50034$:
1403 005444      IF ERRCOD NE #0 THEN
005444 005737 002342      TST      ERRCOD
005450 001403      BEQ      50043$
1404      ;
1405      ; REPORT THE ERROR
1406      ;
1407 005452 004777 175440      JSR      PC,@ERRSVC
1408 005456      ELSE
005456 000425      BR      50044$
005460      50043$:
1409      ;
1410      ; Q UP THE MESSAGE AND ENABLE INTERRUPTS
1411      ; THE I/O DRIVER WILL PICK UP FROM HERE.
1412      ;
1413 005460      LET CURADD(R2) := BUFADD      ; BYTE ADDRESS
005460 013762 002354 002556      MOV      BUFADD,CURADD(R2)
1414 005466      LET MSGADR(R2) := BUFADD      ; MESSAGE ADDRESS
005466 013762 002354 002716      MOV      BUFADD,MSGADR(R2)
1415 005474      LET CURCNT(R2) := BUF CNT      ; OUTPUT COUNT
005474 013762 002356 002756      MOV      BUF CNT,CURCNT(R2)
1416 005502      LET MSGCNT(R2) := BUF CNT      ; BYTE COUNT
005502 013762 002356 002616      MOV      BUF CNT,MSGCNT(R2)
1417 005510      LET REPCNT(R2) := BUF REP      ; PRINT COUNT
005510 013762 002360 002656      MOV      BUF REP,REPCNT(R2)
1418 005516      IF CURCNT(R2) GT #0 THEN
005516 005762 002756      TST      CURCNT(R2)
005522 003403      BLE      50045$
1419 005524      LET @LPCSR(R2) := #100      ; ENABLE INTERRUPTS
005524 012772 000100 002362      MOV      #100,@LPCSR(R2)
1420 005532      ENDIF
005532      50045$:
1421 005532      ENDIF
005532      50044$:
1422 005532      ENDIF
005532      50032$:
1423      ;
1424 005532      DELAY      1      ;PAUSE 100 MICRO-SECONDS TO ALLOW THE LINE $$$
005532 012727 000001      MOV      #1,(PC)+
005536 000000      .WORD      0
005540 013727 002116      MOV      L$DLY,(PC)+
005544 000000      .WORD      0

```

## I/O CONTROL

```

005546 005367 177772      DEC    -6(PC)
005552 001375             BNE    .-4
005554 005367 177756      DEC    -22(PC)
005560 001367             BNE    .-20
1425                                     ;PRINTER TO INTERRUPT. $$$
1426
1427                                     ;
1428                                     ; CLEAR OUT ANY TIMEOUT COUNT
1429                                     ;
1429 005562 005062 003056      CLR    LET DELCNT(R2) := #0
                                DELCNT(R2)
1430                                     ;
1431                                     ; SELECT THE NEXT UNIT AND DECRIMENT THE LINECOUNT
1432                                     ;
1433 005566             LET R3 := R3 - #1
005566 005303             DEC    R3
1434 005570             LET L$LUN := L$LUN + #1
005570 005237 002074      INC    L$LUN
1435 005574 000137 005232      JMP    CTLL0P
1436 005600
1437 005600      CTLEND:      POP    R3,R2
                                MOV    (SP)+,R3
                                MOV    (SP)+,R2
                                RTS    PC
1438 005604 0012603
005600 012602
005602 012602
1438 005604 000207
1439
1440                                     ;++++
1441                                     ; SUBROUTINE QUIET
1442                                     ;
1443                                     ; THIS SUBROUTINE WILL EFFECTIVLY DELAY UNTIL ALL QUEUED OUTPUT
1444                                     ; IS FINISHED. THE DELAY IS ACCOMPLISHED BY QUEUEING A NULL
1445                                     ; MESSAGE TO ALL LINES.
1446                                     ;-----
1447
1448 005606      QUIET:      OUTPUT #0,#0           ; NULL MESSAGE OUTPUT
1449 005650 000240      NOP
1450 005652 000207      RTS    PC
1451

```

I/O CONTROL

```

1453 ;-----
1454 ; DROPIT      FUNCTIONAL DESCRIPTION :
1455 ;
1456 ; THIS SUBROUTINE IS USED TO DROP A BAD PRINTER FROM THE TEST
1457 ; DISABLE ANY INTERRUPTS FROM THE PRINTER, AND NOTIFY THE
1458 ; OPERATOR THAT THE PRINTER WAS DROPPED.
1459 ;-----
1460
1461 005654      052762  040000  002516  DROPIT: LET STATUS(R2) := STATUS(R2) SET.BY #DROPE
1462 005662      012762  177777  002756  BIS      #DROPE,STATUS(R2)
1463 005670      005072  002362  LET CURCNT(R2) := #-1
1464 005674      013746  002074  MOV      #-1,CURCNT(R2)
1465 005700      012746  004312  LET @LPCSR(R2) := #0
1466 005704      012746  000002  CLR      @LPCSR(R2)
1467 005710      010600  PRINTF  #LPDROP, L$LUN
1468 005712      104417  MOV      L$LUN,-(SP)
1469 005714      062706  000006  MOV      #LPDROP,-(SP)
1470 005720      005062  003120  MOV      #2,-(SP)
1471 005724      005337  002346  MOV      SP,R0
1472 005730      005737  002346  TRAP    C#PNTF
1473 005734      001011  ADD     #6,SP
1474 005736      012746  003770  LET ERRTBL(R2) := #0
1475 005742      012746  000001  CLR     ERRTBL(R2)
1476 005746      010600  LET UUT := UUT - #1
1477 005750      104417  DEC    UUT
1478 005752      062706  000004  IF UUT EQ #0 THEN
1479 005756      104444  TST    UUT
1480 005760      000207  BNE    50046#
1481 005762      005037  002074  PRINTF #UUTEQ0
1482 005766      023737  002074  002012  MOV     #UUTEQ0,-(SP)
1483 005774      002007  MOV     #1,-(SP)
1484 005776      013700  002074  MOV     SP,R0
1485 005776      TRAP    C#PNTF
1486 005776      ADD     #4,SP
1487 005776      DOCLN   ; NOTHING TO TEST
1488 005776      TRAP    C#DCLN
1489 005776      ENDIF
1490 005776      50046#:
1491 005776      RTS     PC
1492
1493 ;-----
1494 ; FAKE      FUNCTIONAL DESCRIPTION:
1495 ;
1496 ; THIS SUBROUTINE IS REQUIRED TO INSURE PROPER PASS COUNT REPORTS
1497 ; IN A MULTI UNIT MODE OF OPERATION.
1498 ;-----
1499
1500 FAKE: LET L$LUN := #0
1501 CLR    L$LUN
1502 WHILE L$LUN LT L$UNIT DO
1503 50047#:
1504 CMP    L$LUN,L$UNIT
1505 BGE    50050#
1506 GPHARD L$LUN, R3
1507 MOV    L$LUN,R0

```



I/O CONTROL

006002 104442  
006004 010003  
1485 006006  
006006 005237 002074  
1486 006012  
006012 000765  
006014  
1487 006014 000207  
1488  
1489  
1490 006016

TRAP C\$GPHRD  
MOV RO,R3  
LET L\$LUN := L\$LUN + #1  
INC L\$LUN  
ENDDO  
BR 50047\$  
50050\$:  
RTS PC  
  
ENDMOD

## INITIALIZATION SECTION

```

1492          .SBTTL  INITIALIZATION SECTION
1493          ;**
1494          ;THE INITIALIZE ROUTINE IS EXECUTED AT THE BEGINNING OF EACH SUB-PASS AND IS
1495          ;PRIMARILY USED FOR REQUESTING P-TABLE PARAMETERS. INFORMATION REQUESTED FROM
1496          ;THE OPERATOR INCLUDE THE NUMBER OF UNITS UNDER TEST, DEVICE ADDRESSES, VECTORS,
1497          ;CLOCK TYPE, AUTO OR MANUAL PRINTING SPEED MEASUREMENT, AND WHETHER A DAVFU
1498          ;OPTION IS INSTALLED IN THE SYSTEM.
1499          ;--
1500 006016     BGNMOD
1501 006016     BGNINIT
1502          L$INIT::
1503          ;RESET EXTERNAL BUS IF START EVENT FLAG IS SET
1504          ;OR POWER FAIL RESTART
1504 006016     READEF  #EF.START                ;TEST START EF INDICATOR
1504 006016 012700 000040     MOV  #EF.START,RO
1504 006022 104447     TRAP  C$REFG
1505 006024     BCOMPLETE 1$                ;BRANCH IF FROM START UP
1505 006024 103466     BCS  1$
1506 006026     READEF  #EF.RESTART            ;NOW THE RESTARTFLAG
1506 006026 012700 000037     MOV  #EF.RESTART,RO
1506 006032 104447     TRAP  C$REFG
1507 006034     BCOMPLETE 1$                ;IF EITHER START OR POWER FAIL RESTART
1507 006034 103462     BCS  1$
1508          ;DO A BUS RESET
1509 006036 004737 005762     JSR  PC,FAKE
1509          ; UPDATE PASS COUNT
1510 006042     SETPRI  #PRIO0
1510          ; PRIORITY ZERO
1510 006042 012700 000000     MOV  #PRIO0,RO
1510 006046 104441     TRAP  C$SPRI
1511 006050     LET OUTBUF :B= #14
1511 006050 112737 000014 003164     MOV  #14,OUTBUF
1512 006056     OUTPUT #OUTBUF,#1
1513 006120     DECR WORK1 FROM #6 TO #1 BY #1
1513 006120 012737 000006 003162     MOV  #6,WORK1
1513 006126 000402     BR  50051$
1513 006130     50052$:
1513 006130 005337 003162     DEC  WORK1
1513 006134     50051$:
1513 006134 023727 003162 000001     CMP  WORK1,#1
1513 006142 002415     BLT  50053$
1514 006144     DELAY 250
1514 006144 012727 000250     MOV  #250,(PC)+
1514 006150 000000     .WORD 0
1514 006152 013727 002116     MOV  L$DLY,(PC)+
1514 006156 000000     .WORD 0
1514 006160 005367 177772     DEC  -6(PC)
1514 006164 001375     BNE  .-4
1514 006166 005367 177756     DEC  -22(PC)
1514 006172 001367     BNE  .-20
1515 006174     ENDDEC
1515 006174 000755     BR  50052$
1515 006176     50053$:
1516 006176     EXIT INIT                ; ELSE EXIT INIT CODE
1516 006176 104432     TRAP  C$EXIT
1516 006200 001666     .WORD  L10004-.
1517          ;
1518          ;POWER UP RESTART OR START COMMAND ISSUED
1519          ;

```

## INITIALIZATION SECTION

```

1520 006202          1$:      BRESET          ;RESET THE BUS
      006202 104433      TRAP      C$RESET
1521 006204          IF L$UNIT GT #16. THEN
      006204 023727 002012 000020      CMP      L$UNIT,#16.
      006212 003420      BLE      50054$
1522 006214          PRINTF #NRGT16
      006214 012746 007246      MOV      #NRGT16,-(SP)
      006220 012746 000001      MOV      #1,-(SP)
      006224 010600      MOV      SP,R0
      006226 104417      TRAP      C$PNTF
      006230 062706 000004      ADD      #4,SP
1523 006234          PRINTF #NRGT17
      006234 012746 007331      MOV      #NRGT17,-(SP)
      006240 012746 000001      MOV      #1,-(SP)
      006244 010600      MOV      SP,R0
      006246 104417      TRAP      C$PNTF
      006250 062706 000004      ADD      #4,SP
1524 006254          ENDIF
      006254
1525 006254          50054$:      MANUAL          ; CHECK FOR UNATTENDED MODE
      006254 104450      TRAP      C$MANI
1526 006256          BNCOMPLETE 2$          ; IF UNATTENDED BYPASS MANUAL INSTRUCTIONS
      006256 103034      BCC      2$
1527
1528 006260          PRINTF #RESET1
      006260 012746 007464      MOV      #RESET1,-(SP)
      006264 012746 000001      MOV      #1,-(SP)
      006270 010600      MOV      SP,R0
      006272 104417      TRAP      C$PNTF
      006274 062706 000004      ADD      #4,SP
1529 006300          PRINTF #RESET2
      006300 012746 007557      MOV      #RESET2,-(SP)
      006304 012746 000001      MOV      #1,-(SP)
      006310 010600      MOV      SP,R0
      006312 104417      TRAP      C$PNTF
      006314 062706 000004      ADD      #4,SP
1530
1531          ;WAIT FOR A "CR" BEFORE GOING ON
1532          ;
1533 006320          LET FLAG := #0
      006320 005037 002300      CLR      FLAG
1534 006324          LET ERRCOD := #0
      006324 005037 002342      CLR      ERRCOD
1535 006330          LET UUT := #0
      006330 005037 002346      CLR      UUT
1536 006334          100$:
1537 006334          GMANIL  READY,FLAG,100000,YES
      006334 104443      TRAP      C$GMAN
      006336 000404      BR      10000$
      006340 002300      .WORD   FLAG
      006342 000130      .WORD   T$CODE
      006344 007673      .WORD   READY
      006346 100000      .WORD   100000
      006350
1538
1539          ;REQUEST P-TABLE FOR PRINTERS UNDER TEST
1540          ;

```

INITIALIZATION SECTION

```

1541 006350          2$:   LET R1 := L$UNIT - #1           ;MAXIMUM NUMBER OF UNITS
      006350 013701 002012   MOV     L$UNIT,R1
      006354 005301          DEC     R1
1542 006356          INCR L$LUN FROM #0 TO R1 BY #1
      006356 005037 002074   CLR     L$LUN
      006362 000402          BR      50055$
      006364          50056$: INC     L$LUN
      006364 005237 002074   50055$: CMP     L$LUN,R1
      006370          BGT     50057$
      006370 023701 002074   GPHARD L$LUN,R3           ;REQUEST P-TABLE ADDRESS
      006374 003150          MOV     L$LUN,R0
1543 006376          TRAP   C$GPHRD
      006376 013700 002074   MOV     R0,R3
      006402 104442          BNCOMPLETE 3$           ;BRANCH IF DEVICE NOT PRESENT
      006404 010003          BCC     3$
1544 006406          LET R2 := L$LUN SHIFT 1
      006406 103137          MOV     L$LUN,R2
      006410          ASL     R2
1545 006410          ;
      006410 013702 002074   ; CLEAR ERROR COUNT, OUTPUT COUNT, GET DEVICE TYPE TO STATUS.
      006414 006302          ;
1546
1547
1548
1549 006416          IF 4(R3) EQ #0 THEN
      006416 005763 000004   TST     4(R3)
      006422 001004          BNE     50060$
1550 006424          LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG26!FLAG07
      006424 042762 003000 002516 BIC     #FLAG26!FLAG07,STATUS(R2)
1551 006432          BIC     #FLAG26!FLAG07,STATUS(R2)
      006432 000434          ELSE
      006434          BR      50061$
1552 006434          50060$: IF 4(R3) EQ #1 THEN
      006434 026327 000004 000001 CMP     4(R3),#1
      006442 001007          BNE     50062$
1553 006444          LET STATUS(R2) := STATUS(R2) SET.BY #FLAG26
      006444 052762 001000 002516 BIS     #FLAG26,STATUS(R2)
1554 006452          LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG07
      006452 042762 002000 002516 BIC     #FLAG07,STATUS(R2)
1555 006460          ELSE
      006460 000421          BR      50063$
1556 006462          50062$: IF 4(R3) EQ #2 THEN
      006462 026327 000004 000002 CMP     4(R3),#2
      006470 001007          BNE     50064$
1557 006472          LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG26
      006472 042762 001000 002516 BIC     #FLAG26,STATUS(R2)
1558 006500          LET STATUS(R2) := STATUS(R2) SET.BY #FLAG07
      006500 052762 002000 002516 BIS     #FLAG07,STATUS(R2)
1559 006506          ELSE
      006506 000406          BR      50065$
1560 006510          50064$: LET STATUS(R2) := STATUS(R2) SET.BY #FLAG26
      006510 052762 001000 002516 BIS     #FLAG26,STATUS(R2)
1561 006516          LET STATUS(R2) := STATUS(R2) SET.BY #FLAG07
      006516 052762 002000 002516 BIS     #FLAG07,STATUS(R2)
1562 006524          ENDIF
      006524          50065$:

```

## INITIALIZATION SECTION

```

1563 006524                                ENDIF
      006524                                50063$:
1564 006524                                ENDIF
      006524                                50061$:
1565                                     ;
1566                                     ; NOW GET THE BAND TYPE 64 OR 96 CHARACTER
1567                                     ;
1568 006524                                IF 6(R3) EQ #0 THEN
      006524 005763 000006                TST    6(R3)
      006530 001004                        BNE    50066$
1569 006532                                LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG96
      006532 042762 010000 002516        BIC    #FLAG96,STATUS(R2)
1570 006540                                ELSE
      006540 000403                        BR     50067$
      006542                                50066$:
1571 006542                                LET STATUS(R2) := STATUS(R2) SET.BY #FLAG96
      006542 052762 010000 002516        BIS    #FLAG96,STATUS(R2)
1572 006550                                ENDIF
      006550                                50067$:
1573 006550                                LET ERRTBL(R2) := #0
      006550 005062 003120                CLR    ERRTBL(R2)
1574 006554                                LET CURCNT(R2) := #-1
      006554 012762 177777 002756        MOV    #-1,CURCNT(R2)
1575 006562                                LET DELCNT(R2) := #0
      006562 005062 003056                CLR    DELCNT(R2)
1576 006566                                LET REPCNT(R2) := #0
      006566 005062 002656                CLR    REPCNT(R2)
1577                                     ;
1578                                     ;LOAD CSR ADDRESS INTO TABLE
1579                                     ;
1580 006572                                LET LPCSR(R2) := (R3)+ ;SET UP CSR ADDRESS FOR DEVICE
      006572 012362 002362                MOV    (R3)+,LPCSR(R2)
1581 006576                                LET LPBUF(R2) := LPCSR(R2) + #2
      006576 016262 002362 002456        MOV    LPCSR(R2),LPBUF(R2)
      006604 062762 000002 002456        ADD    #2,LPBUF(R2)
1582                                     ;
1583                                     ;SET UP VECTOR ADDRESS INTO GIVEN TABLE
1584                                     ;
1585 006612                                LET LPVEC(R2) := (R3)+
      006612 012362 002422                MOV    (R3)+,LPVEC(R2)
1586                                     ;
1587                                     ;SET UP DEVICE INTERRUPT VECTOR INFORMATION
1588                                     ;
1589 006616                                LET WORK := R2 SHIFT 3
      006616 010237 003160                MOV    R2,WORK
      006622 006337 003160                ASL    WORK
      006626 006337 003160                ASL    WORK
      006632 006337 003160                ASL    WORK
1590 006636                                LET WORK := WORK + #INT00
      006636 062737 037166 003160        ADD    #INT00,WORK
1591 006644                                LET LPINTR(R2) := WORK
      006644 013762 003160 003016        MOV    WORK,LPINTR(R2)
1592 006652                                SETVEC LPVEC(R2), LPINTR(R2), #PRI04
      006652 012746 000200                MOV    #PRI04,-(SP)
      006656 016246 003016                MOV    LPINTR(R2),-(SP)
      006662 016246 002422                MOV    LPVEC(R2),-(SP)
      006666 012746 000003                MOV    #3,-(SP)

```

INITIALIZATION SECTION

```

006672 104437          TRAP  C$SVEC
006674 062706 000010  ADD   #10,SP
1593
1594 ; ADD ONE TO UNIT UNDER TEST COUNT
1595 ;
1596 006700          LET UUT := UUT + #1
006700 005237 002346  INC   UUT
1597 006704 000403          BR   4$
1598
1599 ;INDICATE L$LUN NOT AVAILABLE FOR TESTING
1600 ;
1601 006706          3$: LET STATUS(R2) := STATUS(R2) SET.BY #DROPED
006706 052762 040000 002516  BIS   #DROPED,STATUS(R2)
1602 006714          4$: ENDINC                               ;GO BACK AND DO IT AGAIN
006714 000623          BR   50056$
006716          50057$:
1603 ;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1604 ; SETUP TO HANDLE CLOCK INTERRUPTS
1605 ; IF AN L-CLOCK IS ON THE SYSTEM THEN SETUP A NOOP INTERRUPT
1606 ; HANDLER BECAUSE LSI SYSTEMS MAY HAVE THE CLOCK ENABLED AT ALL TIMES.
1607 006716          LET CLKTYP := #1 ; DEFAULT FOR NO CLOCK ON SYSTEM
006716 012737 000001 002326  MOV   #1,CLKTYP
1608 006724          CLOCK  L,R4 ; TEST FOR L-CLOCK
006724 012700 000114  MOV   #'L,R0
006730 104462          TRAP  C$CLK
006732 010004          MOV   R0,R4
1609 006734          IFCOND CS THEN ; WE HAVE AN L-CLOCK
006734 103031          BCC   50070$
1610 006736          LET CLKTYP := #2
006736 012737 000002 002326  MOV   #2,CLKTYP
1611 006744          LET CLOCKP := R4
006744 010437 002330  MOV   R4,CLOCKP
1612 006750          LET CLKCSR := @CLOCKP
006750 017737 173354 002332  MOV   @CLOCKP,CLKCSR
1613 006756          LET @CLKCSR := #00 ; TRY TO DISABLE INTERRUPTS
006756 012777 000000 173346  MOV   #00,@CLKCSR
1614 ; SETUP THE NOOP HANDLER
1615 006764          LET CLKVEC := 4(R4)
006764 016437 000004 002336  MOV   4(R4),CLKVEC
1616 006772          SETVEC CLKVEC,#IGNORE,#PRIO6
006772 012746 000300  MOV   #PRIO6,-(SP)
006776 012746 010070  MOV   #IGNORE,-(SP)
007002 013746 002336  MOV   CLKVEC,-(SP)
007006 012746 000003  MOV   #3,-(SP)
007012 104437          TRAP  C$SVEC
007014 062706 000010  ADD   #10,SP
1617 007020          ENDIF
007020          50070$:
1618 ; IF THE OPERATOR WANTS MANUAL SPEED TEST SET CLOCK TYPE = 4
1619 007020 005737 002270  TST  MANS PD
1620 007024 001410          BEQ   CK1
1621 007026          LET CLKTYP := #4
007026 012737 000004 002326  MOV   #4,CLKTYP
1622 007034          SETPRI #PRIO0 ; START TEST AT PRI 0
007034 012700 000000  MOV   #PRIO0,R0
007040 104441          TRAP  C$SPRI
1623 007042          EXIT  INIT

```

INITIALIZATION SECTION

```

007042 104432          TRAP  C$EXIT
007044 001022          .WORD  L10004-.
1624      ; IF A P-CLOCK IS ON THE SYSTEM UPGRADE CLOCK TYPE TO 3
1625 007046          CK1:  CLOCK  P,R4
      007046 012700 000120  MOV   #'P,R0
      007052 104462          TRAP  C$CLCK
      007054 010004          MOV   RO,R4
1626 007056          IFCOND CS THEN          ; WE HAVE A P-CLOCK
      007056 103016          BCC   50071$
1627 007060          LET  CLKTYP := #3
      007060 012737 000003 002326  MOV   #3,CLKTYP
1628 007066          LET  CLOCKP := R4
      007066 010437 002330  MOV   R4,CLOCKP
1629 007072          LET  CLKCSR := @CLOCKP
      007072 017737 173232 002332  MOV   @CLOCKP,CLKCSR
1630 007100          LET  CLKVEC := 4(R4)
      007100 016437 000004 002336  MOV   4(R4),CLKVEC
1631      ; TRY TO DISABLE THE P-CLOCK
1632 007106          LET  @CLKCSR := #00
      007106 012777 000000 173216  MOV   #00,@CLKCSR
1633 007114          ENDIF
      007114          50071$:
1634      ; IF NO CLOCKS ON THE SYSTEM NOTIFY THE OPERATOR
1635 0C7114          IF  CLKTYP EQ #1 THEN
      007114 023727 002326 000001  CMP   CLKTYP,#1
      007122 001020          BNE   50072$
1636 007124          PRINTF #NOCLCK
      007124 012746 007730          MOV   #NOCLCK,-(SP)
      007130 012746 000001          MOV   #1,-(SP)
      007134 010600          MOV   SP,RO
      007136 104417          TRAP  C$PNTF
      007140 062706 000004          ADD   #4,SP
1637 007144          PRINTF #NOTIM
      007144 012746 007772          MOV   #NOTIM,-(SP)
      007150 012746 000001          MOV   #1,-(SP)
      007154 010600          MOV   SP,RO
      007156 104417          TRAP  C$PNTF
      007160 062706 000004          ADD   #4,SP
1638 007164          ENDIF
      007164          50072$:
1639 007164          SETPRI #PRI00
      007164 012700 000000          MOV   #PRI00,RO
      007170 104441          TRAP  C$SPRI
1640 007172          LET  OUTBUF :B= #14
      007172 112737 000014 003164  MOVB  #14,OUTBUF
1641 007200          OUTPUT #OUTBUF,#1
1642 007242          EXIT  INIT
      007242 104432          TRAP  C$EXIT
      007244 000622          .WORD  L10004-.
1643      .NLIST BEX
1644
1645 007246          045      116      045  NRGT16: .ASCIZ  /%N%ANUMBER OF LINE PRINTERS UNDER TEST EXCEEDS 16./
1646 007331          045      116      045  NRGT17: .ASCIZ  /%N%ONLY 16 WILL BE TESTED./
1647 007365          045      116      045  MRESET: .ASCIZ  /%N%ARESET LINE PRINTER(S), DO FORM FEED, AND PLACE ON LINE.%N/
1648 007464          045      116      045  RESET1: .ASCIZ  /%N%ARESET LINE PRINTER(S),SET FORMS LENGTH TO 11 INCHES,%N/
1649 007557          045      101      123  RESET2: .ASCIZ  /%A%SET VERTICAL DENSITY TO 6 LINES PER INCH,DO FORM FEED AND PLACE ON LINE%N/
1650

```

INITIALIZATION SECTION

```

1651 007673      104      105      120  READY:  .ASCIZ  /DEPRESS "RETURN" WHEN READY./
1652 007730      045      116      045  NOCLCK: .ASCIZ  /%N%AHARDWARE CLOCK NOT AVAILABLE./
1653 007772      045      116      045  NOTIM:  .ASCIZ  /%N%AAUTO PRINTING SPEED MEASUREMENT CANNOT BE PERFORMED./
1654                                     .EVEN
1655 010064      000000    PLOC:   .WORD   0
1656
1657                                     .LIST BEX
1658 010066      ENDINIT
      010066      L10004:
      010066      104411    TRAP   C$INIT
1659
1660                                     ;
1661                                     ; IGNORE          AN INTERRUPT CATCHER FOR THE L-CLOCK
1662                                     ;                  THAT IGNORES THE INTERRUPT.
1663                                     ;                  USED FOR SYSTEMS WHERE CLOCK CANNOT BE TURNED OFF.
1664                                     ;
1665                                     ;
1666 010070      IGNORE:   ; NOOP
1667 010070      000002    RTI
1668
1669
1670
1671
1672                                     ;
1673                                     ; RESVEC          FUNCTIONAL DESCRIPTION
1674                                     ;
1675                                     ; THIS SUBROUTINE WILL SETUP ALL UNITS VECTOR AREAS
1676                                     ; TO THE 'NORMAL' INTERRUPT ROUTINES STARTING AT INT00.
1677                                     ;
1678                                     ;
1679 010072      RESVEC::   PUSH   R3,R4
      010072      010346    MOV    R3,-(SP)
      010074      010446    MOV    R4,-(SP)
1680 010076      LET    R4 := #0
      010076      005004    CLR    R4
1681 010100      LET    R3 := L$UNIT
      010100      013703    002012  MOV    L$UNIT,R3
1682 010104      WHILE  R3 GT #0 DO
      010104
      50073$:
      010104      005703    TST   R3
      010106      003417    BLE   50074$
1683 010110      SETVEC LPVEC(R4), LPINTR(R4), #PRI04
      010110      012746    000200  MOV    #PRI04,-(SP)
      010114      016446    003016  MOV    LPINTR(R4),-(SP)
      010120      016446    002422  MOV    LPVEC(R4),-(SP)
      010124      012746    000003  MOV    #3,-(SP)
      010130      104437    TRAP  C$SVEC
      010132      062706    000010  ADD    #10,SP
1684 010136      LET    R4 := R4 + #2
      010136      062704    000002  ADD    #2,R4
1685 010142      LET    R3 := R3 - #1
      010142      005303    DEC   R3
1686 010144      ENDDO
      010144      000757    BR    50073$
      010146
      50074$:
1687 010146      POP   R4,R3
      010146      012604    MOV   (SP)+,R4

```



INITIALIZATION SECTION

010150 012603  
1688 010152 000207  
1689

MOV (SP)+,R3  
RTS PC

CLEANUP CODING SECTION

```

1691
1692 010154

1693
1694
1695
1696
1697
1698
1699
1700
1701
1702 010154

1703 010154
      010154
1704 010154
      010154 012700 000340
      010160 104441
1705 010162
      010162 104433

1706
1707 010164
      010164 013701 002012
      010170 005301
1708 010172
      010172 005037 002074
      010176 000402
      010200
      010200 005237 002074
      010204
      010204 023701 002074
      010210 003020

1709
1710
1711 010212
      010212 013702 002074
      010216 006302

1712
1713 010220
      010220 042762 160377 002516
1714 010226
      010226 012762 177777 002756
1715 010234
      010234 005062 003120
1716 010240
      010240 005062 003056
1717 010244
      010244 005062 002656
1718 010250
      010250 000753
      010252
1719 010252 004737 010072
1720 010256
      010256 023727 002326 000003
      010264 001006
1721 010266
    
```

```

.SBTTL CLEANUP CODING SECTION
STARS
;*****
;+
;THE PURPOSE OF THE CLEANUP SECTION IS TO CLEANUP ALL PRINTERS UNDER TEST
;AND RETEST ANY UNITS WHICH HAVE BEEN DROPPED FROM TESTING TO INSURE THAT
;THEY HAVE NOT COME BACK ON LINE. IF THE DEVICE HAS COME BACK ON LINE
;TESTING WILL BE RESTARTED ON THE DEVICE. THIS INSURES THAT
;IN THE EVENT A PAPER OUT OCCURRED AND THE OPERATOR HAS PUT ADDITIONAL PAPER
;INTO THE UNIT UNDER TEST, THE INITIALIZATION SEQUENCE DOES NOT
;HAVE TO BE DONE AGAIN IN ORDER TO GET THE DEVICE ACTIVE.
;--
STARS
;*****
BGNCLN
L$CLEAN::
      SETPRI #PRI07
      MOV #PRI07,R0
      TRAP C$SPRI
      BRESET
      TRAP C$RESET

CLEAN: LET R1 := L$UNIT - #1 ;NUMBER OF UNITS-1
      MOV L$UNIT,R1
      DEC R1
      INCR L$LUN FROM #0 TO R1 BY #1
      CLR L$LUN
      BR 50075$

50076$: INC L$LUN
50075$: CMP L$LUN,R1
      BGT 50077$
; DISABLE ALL INTERRUPTS, SELECT ALL LINES
; ZERO ALL ERROR COUNTS
      LET R2 := L$LUN SHIFT 1
      MOV L$LUN,R2
      ASL R2
; CLEAR ALL BITS IN STATUS EXCEPT DEVICE TYPE
      LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR!DROPE!ACTIVE!LOBYTE
      BIC #ERROR!DROPE!ACTIVE!LOBYTE,STATUS(R2)
      LET CURCNT(R2) := #-1
      MOV #-1,CURCNT(R2)
      LET ERRTBL(R2) := #0
      CLR ERRTBL(R2)
      LET DELCNT(R2) := #0
      CLR DELCNT(R2)
      LET REPCNT(R2) := #0
      CLR REPCNT(R2)
      ENDINC
      BR 50076$

50077$: JSR PC,RESVEC ; RESET THE VECTORS
      IF CLKTYP EQ #3 THEN
      CMP CLKTYP,#3
      BNE 50100$
      CLRVEC @CLKVEC
    
```

CLEANUP CODING SECTION

010266	017700	172044		MOV	@CLKVEC,R0
010272	104436			TRAP	C\$CVEC
1722 010274				LET	@CLKCSR := #00
010274	012777	000000	172030	MOV	#00,@CLKCSR
1723 010302				ENDIF	
010302				50100\$:	
1724 010302				IF	CLKTYP EQ #2 THEN
010302	023727	002326	000002	CMP	CLKTYP,#2
010310	001013			BNE	50101\$
1725 010312				SETVEC	CLKVEC,#IGNORE,#PRI06
010312	012746	000300		MOV	#PRI06,-(SP)
010316	012746	010070		MOV	#IGNORE,-(SP)
010322	013746	002336		MOV	CLKVEC,-(SP)
010326	012746	000003		MOV	#3,-(SP)
010332	104437			TRAP	C\$SVEC
010334	062706	000010		ADD	#10,SP
1726 010340				ENDIF	
010340				50101\$:	
1727 010340				SETPRI	#PRI00
010340	012700	000000		MOV	#PRI00,R0
010344	104441			TRAP	C\$SPRI
1728 010346				ENDCLN	
010346				L10005:	
010346	104412			TRAP	C\$CLEAN
1729					
1730 010350				ENDMOD	

## INTERFACE LOGIC

```

1732          .SBTTL  INTERFACE LOGIC
1733
1734 010350    BGNMOD
1735          ;**
1736          ;THIS TEST VERIFIES THE OPERATION OF THE INTERFACE LOGIC. TESTS ARE
1737          ;PERFORMED FOR PRINTER ERROR, PRINTER READY, AND CLEARING PRINTER READY
1738          ;BY LOADING A CHARACTER INTO THE OUTPUT BUFFER. ALSO IT IS VERIFIED
1739          ;THAT THE PRINTER WILL NOT INTERRUPT IF IT IS AT THE SAME PRIORITY LEVEL
1740          ;AS THE PROCESSOR, BUT WILL INTERRUPT IF THE PROCESSOR IS AT A LOWER
1741          ;PRIORITY LEVEL. THE PRINTER IS AT PRIORITY LEVEL 4.
1742          ;
1743          ;
1744          ;--
1745 010350    BGNTST 1
1746          010350    T1::
1746 010350    LET R1 := L$UNIT - #1          ;MAX NUMBER OF UNITS ON SYSTEM
1746 010350    013701    MOV      L$UNIT,R1
1746 010354    005301    DEC      R1
1747
1748          ;
1749          ;HARD CODED INCREMEMNT LOOP
1750          ;INCR LUNIT FROM #0 TO R1 BY #1          ;START LOOP
1751 010356    005037    002320    CLR      LUNIT          ;UNIT TO 0
1752 010362    000402          BR      T1C          ;DO COMPARE
1753 010364
1754 010364    005237    002320    T1A:    INC      LUNIT          ;UPDATE UNIT NUMBER
1755 010370
1756 010370    023701    002320    T1C:    CMP      LUNIT,R1          ;DO COMPARISON OF UNIT NUMBER
1757 010374    003402          BLE     1$          ;ONTO NEXT UNIT
1758 010376    000137    011142          JMP     T1B          ;EXIT LOOP
1759 010402
1760 010402
1760 010402    013702    002320    1$:    LET R2 := LUNIT SHIFT 1
1760 010406    006302          MOV      LUNIT,R2
1761 010410          ASL      R2
1761 010410    032772    100000    002362    IF #BIT15 SETIN @LPCSR(R2) THEN
1761 010416    001416          BIT      #BIT15,@LPCSR(R2)
1762 010420          BEQ     50102$
1762 010420    052762    100000    002516    LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
1763 010426    005262    003120          BIS     #ERROR,STATUS(R2)
1764 010432          LET ERRTBL(R2) := ERRTBL(R2) + #1
1764 010432    013737    002320    002074    INC     ERRTBL(R2)
1765 010440          LET L$LUN := LUNIT
1765 010440    104456          MOV     LUNIT,L$LUN
1765 010442    000001          ERRHRD 1,CSRERR          ;ERROR BIT WAS SET. SAY SO
1765 010444    003412          TRAP   C$ERHRD
1765 010446    000000          .WORD  1
1766 010450          .WORD  CSRERR
1766 010450    005072    002362          .WORD  0
1767 010454          LET @LPCSR(R2) := #0
1767 010454    010454          CLR     @LPCSR(R2)
1768          ENDF
1769          50102$:
1770 010454          ;TIME DELAY
1770 010454    032772    000200    002362    ; IF NOT READY ALLOW 3 SECONDS TO COME UP
1770 010462    001027          IF #BIT7 NOTSETIN @LPCSR(R2) THEN
1770 010462          BIT     #BIT7,@LPCSR(R2)
1770 010462          BNE     50103$

```

## INTERFACE LOGIC

```

1771 010464          DECRC WORK1 FROM #12. TO #1 BY #1
      010464 012737 000014 003162      MOV    #12.,WORK1
      010472 000402          BR     50104#
      010474          50105#:
      010474 005337 003162          DEC    WORK1
      010500          50104#:
      010500 023727 003162 000001      CMP    WORK1,#1
      010506 002415          BLT    50106#
1772 010510          DELAY 250
      010510 012727 000250          MOV    #250,(PC)+
      010514 000000          .WORD 0
      010516 013727 002116          MOV    L#DLY,(PC)+
      010522 000000          .WORD 0
      010524 005367 177772          DEC    -6(PC)
      010530 001375          BNE   -.4
      010532 005367 177756          DEC    -22(PC)
      010536 001367          BNE   -.20
1773 010540          ENDDC
      010540 000755          BR     50105#
      010542          50106#:
1774 010542          ENDIF
      010542          50103#:
1775          ;
1776          ;NOW TEST FOR PRINTER READY
1777          ;
1778 010542          IF #BIT07 NOTSETIN @LPCSR(R2) THEN          ;TEST FOR THE READY BIT
      010542 032772 000200 002362      BIT    #BIT07,@LPCSR(R2)
      010550 001014          BNE   50107#
1779 010552          LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
      010552 052762 100000 002516      BIS    #ERROR,STATUS(R2)
1780 010560          LET L#LUN := LUNIT
      010560 013737 002320 002074      MOV    LUNIT,L#LUN
1781 010566          LET ERRTBL(R2) := ERRTBL(R2) + #1
      010566 005262 003120          INC    ERRTBL(R2)
1782 010572          ERRHRD 2,RDYERR          ;REPORT AN ERROR
      010572 104456          TRAP   C#ERRHRD
      010574 000002          .WORD 2
      010576 003430          .WORD RDYERR
      010600 000000          .WORD 0
1783 010602          ENDDC
      010602          50107#:
1784          ;
1785          ;INSURE LOADING CHARACTER CAUSES PRINTER READY TO GO AWAY
1786          ;
1787 010602          LET @LPBUF(R2) := #12
      010602 012772 000012 002456      MOV    #12,@LPBUF(R2)
1788 010610          IF #BIT07 SETIN @LPCSR(R2) THEN
      010610 032772 000200 002362      BIT    #BIT07,@LPCSR(R2)
      010616 001416          BEQ   50110#
1789 010620          LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
      010620 052762 100000 002516      BIS    #ERROR,STATUS(R2)
1790 010626          LET ERRTBL(R2) := ERRTBL(R2) + #1
      010626 005262 003120          INC    ERRTBL(R2)
1791 010632          LET L#LUN := LUNIT
      010632 013737 002320 002074      MOV    LUNIT,L#LUN
1792 010640          ERRHRD 3,ERR11          ;REPORT AN ERROR
      010640 104456          TRAP   C#ERRHRD

```

## INTERFACE LOGIC

```

010642 000003          .WORD 3
010644 011422          .WORD ERR11
010646 000000          .WORD 0
1793 010650          LET @LPCSR(R2) := #0
010650 005072 002362  CLR @LPCSR(R2)
1794 010654          ENDIF
010654          50110$:
1795          ;
1796          ;VERIFY THAT THE PRINTER WILL NOT INTERRUPT IF IT IS AT A PRIORITY LEVEL
1797          ;THE SAME AS THE CPU
1798          ;
1799 010654          SETPRI #PRI04          ;CPU TO PRIORITY 4
010654 012700 000200  MOV @PRI04,R0
010660 104441          TRAP C$SPRI
1800 010662          SETVEC LPVEC(R2),#INTERR,#PRI04  ;LP VECTOR SET UP
010662 012746 000200  MOV @PRI04,-(SP)
010666 012746 011304  MOV #INTERR,-(SP)
010672 016246 002422  MOV LPVEC(R2),-(SP)
010676 012746 000003  MOV #3,-(SP)
010702 104437          TRAP C$SVEC
010704 062706 000010  ADD #10,SP
1801 010710          LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100  ;INTERRUPT ENABLE
010710 052772 000100 002362  BIS #100,@LPCSR(R2)
1802 010716          DECR WORK1 FROM #12 TO #1 BY #1
010716 012737 000012 003162  MOV #12,WORK1
010724 000402          BR 50111$
010726          50112$:
010726 005337 003162  DEC WORK1
010732          50111$:
010732 023727 003162 000001  CMP WORK1,#1
010740 002415          BLT 50113$
1803 010742          DELAY 250.          ; ALLOW 3 SEC FOR DELAY
010742 012727 000372  MOV #250.,(PC)+
010746 000000          .WORD 0
010750 013727 002116  MOV L$DLY,(PC)+
010754 000000          .WORD 0
010756 005367 177772  DEC -6(PC)
010762 001375          BNE -.4
010764 005367 177756  DEC -22(PC)
010770 001367          BNE .-20
1804 010772          ENDDC
010772 000755          BR 50112$
010774          50113$:
1805          ;
1806          ;NOW TEST THAT THE PRINTER WILL INTERRUPT IF THE CPU PRIORITY IS LOWER THAN
1807          ;THE PRINTER PRIORITY
1808          ;
1809 010774          LET @LPCSR(R2) := @LPCSR(R2) CLR.BY #100  ;CLEAR INTERRUPT ENABLE
010774 042772 000100 002362  BIC #100,@LPCSR(R2)
1810 011002          SETPRI #PRI03          ;CPU TO PRIORITY 3
011002 012700 000140  MOV @PRI03,R0
011006 104441          TRAP C$SPRI
1811 011010          SETVEC LPVEC(R2),#INTHDL,#PRI04
011010 012746 000200  MOV @PRI04,-(SP)
011014 012746 011334  MOV #INTHDL,-(SP)
011020 016246 002422  MOV LPVEC(R2),-(SP)
011024 012746 000003  MOV #3,-(SP)

```

## INTERFACE LOGIC

```

011030 104437
011032 062706 000010
1812 011036 052772 000100 002362
011036 052772 000100 002362
1813 011044 012727 000030
011044 012727 000030
011050 000000
011052 013727 002116
011056 000000
011060 005367 177772
011064 001375
011066 005367 177756
011072 001367
1814 011074 005262 003120
011074 005262 003120
1815 011100 013737 002320 002074
011100 013737 002320 002074
1816 011106 104456
011106 104456
011110 000004
011112 011561
011114 000000
1817 011116 012772 000000 002362
011116 012772 000000 002362
1818 011124 042762 160000 002516
011124 042762 160000 002516
1819 011132 005062 003056
011132 005062 003056
1820
1821
1822
1823
1824 011136 000137 010364
011136 000137 010364
1825 011142 004737 010072
011142 004737 010072
1826 011146 012700 000000
011146 012700 000000
011152 104441
1827 011154
011154
1828 011216 112737 000014 003164
011216 112737 000014 003164
1829 011224
011224
1830 011266 032772 000200 002362
011266 032772 000200 002362
011274 001001
011274 001001
1831 011276 000773
011276 000773
011300
1832 011300
011300
011302 004472
011302 004472
000534
1833
1834
1835
1836
1837
1838 011304

TRAP C$SVEC
ADD #10,SP
LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100 ;INTERRUPT ENABLE
BIS #100,@LPCSR(R2)
DELAY 30 ; ALLOW 3 SEC DELAY
MOV #30,(PC)+
.WORD 0
MOV L$DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -.4
DEC -22(PC)
BNE -.20
LET ERRTBL(R2) := ERRTBL(R2) + #1
INC ERRTBL(R2)
LET L$LUN := LUNIT
MOV LUNIT,L$LUN
ERRHRD 4,ERR13
TRAP C$ERHRD
.WORD 4
.WORD ERR13
.WORD 0
END2: LET @LPCSR(R2) := #00 ; CLEAR THE LPCSR
MOV #00,@LPCSR(R2)
LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR!DROPE!ACTIVE
BIC #ERROR!DROPE!ACTIVE,STATUS(R2)
LET DELCNT(R2) := #0
CLR DELCNT(R2)
;
;END OF HARD CODED INCREMENT LOOP
;ENDINC
;
T1B: JMP T1A ;UPDATE UNIT #
JSR PC,RESVEC ; RESET STANDARD VECTORS
SETPRI #PRI00
MOV #PRI00,R0
TRAP C$SPRI
OUTPUT #INTFAC,#47.
LET OUTBUF :B= #14
MOVB #14,OUTBUF
OUTPUT #OUTBUF,#1
WHILE #BIT7 NOTSETIN @LPCSR(R2) DO ;WAIT FOR READY
50114$: BIT #BIT7,@LPCSR(R2)
BNE 50115$
ENDDO
BR 50114$
50115$:
EXIT TST ;EXIT THE TEST
TRAP C$EXIT
.WORD L10006-.
;
;INTERRUPT HANDLER TO SERVICE FAULTY INTERRUPT FROM LP INTERFACE.
;THIS ROUTINE IS ENTERED ONLY WHEN THE LP INTERRUPTS AT THE SAME LEVEL AS
;THE CPU AND IS CONSIDERED AN ERROR.
;
BGNSRV

```

## INTERFACE LOGIC

```

1839 011304          INTERR: LET ERRTBL(R2) := ERRTBL(R2) + #1
      011304 005262 003120      INC ERRTBL(R2)
1840 011310          LET L$LUN := LUNIT
      011310 013737 002320 002074  MOV LUNIT,L$LUN
1841 011316          ERRHRD 5,ERR12
      011316 104456      TRAP C$ERRHD
      011320 000005      .WORD 5
      011322 011476      .WORD ERR12
      011324 000000      .WORD 0
1842 011326          LET (SP) := #END2
      011326 012716 011116      MOV #END2,(SP)
1843 011332          ENDSRV
      011332          L10007:
      011332 000002          RTI

1844          ;
1845          ;INTERRUPT HANDLER FOR EXPECTED INTERRUPT
1846          ;
1847 011334          BGNSRV
1848          ;
1849 011334          INTHDL: LET (SP) := #END2
      011334 012716 011116      MOV #END2,(SP)
1850 011340          ENDSRV
      011340          L10010:
      011340 000002          RTI

1851          ;
1852          ;.NLIST BEX
1853 011342          INTFAC: .ASCIZ /INTERFACE LOGIC TEST 1 ---- TEST COMPLETE/<12><12>
      111 116 124
1854          ;
1855          ;ERROR MESSAGES ASSOCIATED WITH THIS TEST
1856          ;
1857 011422          ERR11: .ASCIZ /LOADING PRINTER BUFFER DOES NOT CLEAR READY/
      114 117 101
1858 011476          ERR12: .ASCIZ /PRINTER INTERRUPTED AT SAME LEVEL AS THE PROCESSOR/
      120 122 111
1859 011561          ERR13: .ASCIZ /PRINTER DID NOT INTERRUPT AT CPU PRIORITY 3/
      120 122 111
1860          .EVEN
1861 011636          ENDTST
      011636          L10006:
      011636 104401          TRAP C$ETST
1862          ;.LIST BEX
1863 011640          ENDMOD
1864

```



## READY LINE INTERLOCKS TEST 2

```

1866          .SBTTL  READY LINE INTERLOCKS  TEST 2
1867
1868 011640    BGNMOD
1869          ;**
1870          ;THIS TEST CHECKS THE OPERATION OF THE
1871          ;PRINTER READY INTERLOCK SWITCHES.
1872          ;MANUAL INTERVENTION IS USED TO
1873          ;OPEN THE INTERLOCKS TO PRODUCE FAULTS
1874          ;IN THE PRINTER AFTER WHICH THE RESULTANT ERROR
1875          ;INDICATION IS VERIFIED.
1876          ;--
1877
1878 011640    BGNTST 2
1879          T2::
1880          ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
1881          MANUAL
1882          TRAP  C$MANI
1883          BCOMPLETE 11$
1884          BCS  11$
1885          EXIT TST
1886          TRAP  C$EXIT
1887          .WORD  L10011-.
1888          ;EXIT TEST IF MANUAL INTERVENTION TESTS ARE NOT SPECIFIED
1889 11$:      IF INHINT EQ #0 THEN
1890          TST  INHINT
1891          BNE  50116$
1892          EXIT TST
1893          TRAP  C$EXIT
1894          .WORD  L10011-.
1895
1896          ENDF
1897          50116$:
1898          LET FLAG := #0
1899          CLR  FLAG
1900          LET R1 := L$UNIT - #1
1901          MOV  L$UNIT,R1
1902          DEC  R1
1903
1904          ;CHECK FOR ERROR IN EACH PRINTER UNDER TEST
1905          INCR LUNIT FROM #0 TO R1 BY #1
1906          CLR  LUNIT
1907          BR   50117$
1908          50120$:
1909          INC  LUNIT
1910          50117$:
1911          CMP  LUNIT,R1
1912          BGT  50121$
1913          LET R2 := LUNIT SHIFT 1
1914          MOV  LUNIT,R2
1915          ASL  R2
1916          IF #BIT15 SETIN @LPCSR(R2) THEN
1917          BIT  #BIT15,@LPCSR(R2)
1918          BEQ  50122$
1919          LET ERRTBL(R2) := ERRTBL(R2) + #1
1920          INC  ERRTBL(R2)
1921          ERRHRD 6, CSRERR
1922          TRAP  C$ERHRD

```

READY LINE INTERLOCKS TEST 2

011740	000006			.WORD	6
011742	003412			.WORD	CSRERR
011744	000000			.WORD	0
1897 011746				LET @LPCSR(R2) := #0	
011746	005072	002362		CLR	@LPCSR(R2)
1898 011752				ENDIF	
011752				50122\$:	
1899 011752				ENDINC	
011752	000753			BR	50120\$
011754				50121\$:	
1900				;CHECK FOR READY IN EACH PRINTER UNDER TEST	
1901 011754				INCR LUNIT FROM #0 TO R1 BY #1	
011754	005037	002320		CLR	LUNIT
011760	000402			BR	50123\$
011762				50124\$:	
011762	005237	002320		INC	LUNIT
011766				50123\$:	
011766	023701	002320		CMP	LUNIT,R1
011772	003021			BGT	50125\$
1902 011774				LET R2 := LUNIT SHIFT 1	
011774	013702	002320		MOV	LUNIT,R2
012000	006302			ASL	R2
1903 012002				LET L\$LUN := LUNIT	
012002	013737	002320	002074	MOV	LUNIT,L\$LUN
1904 012010				IF #BIT07 NOTSETIN @LPCSR(R2) THEN	
012010	032772	000200	002362	BIT	#BIT07,@LPCSR(R2)
012016	001006			BNE	50126\$
1905 012020				LET ERRTBL(R2) := ERRTBL(R2) + #1	
012020	005262	003120		INC	ERRTBL(R2)
1906 012024				ERRHRD	7, RDYERR
012024	104456			TRAP	C\$ERHRD
012026	000007			.WORD	7
012030	003430			.WORD	RDYERR
012032	000000			.WORD	0
1907 012034				ENDIF	
012034				50126\$:	
1908 012034				ENDINC	
012034	000752			BR	50124\$
012036				50125\$:	
1909				; PRINT TEST NAME	
1910				; OUTPUT #INTLK,#29.	
1911				;VERIFY OPERATION OF PAPER LOW INTERLOCK SWITCH	
1912 012036				;HARD CODED INCREMENT LOOP	
1913				;	
1914				LET ERRFLG := #0	
1915				CLR	ERRFLG
1916 012100				CLR	LUNIT
012100	005037	002344		BR	1\$
1917 012104				2\$:	
012104	005037	002320		INC	LUNIT
1918 012110				LET R2 := LUNIT SHIFT 1	
012110	000405			MOV	LUNIT,R2
1919 012112				ASL	R2
012112	005237	002320		1\$:	
1920 012112				CMP	LUNIT,R1
012112	013702	002320			
012116	006302				
1922 012124					
012124	023701	002320			

READY LINE INTERLOCKS TEST 2

```

1924 012130 003402
1925 012132 000137 012472
1926 012136
1927 012136
    012136 005037 002300
1928 012142
    012142 013746 002320
    012146 012746 013466
    012152 012746 000002
    012156 010600
    012160 104417
    012162 062706 000006
1929 012166
    012166 012746 013536
    012172 012746 000001
    012176 010600
    012200 104417
    012202 062706 000004
1930 012206
    012206 104443
    012210 000404
    012212 002300
    012214 000130
    012216 007673
    012220 100000
    012222
10000$:
1931 012222
    012222 012737 000310 002302
1932 012230
    012230 005037 002344
1933 012234
    012234
1934 012234
1935 012276
    012276 005337 002302
1936 012302
    012302 005737 002302
    012306 001403
    012310 005737 002344
    012314 001747
    012316
50127$:
1937 012316
    012316 005737 002344
    012322 001011
1938 012324
    012324 104456
    012326 000010
    012330 003452
    012332 000000
1939 012334
    012334 005262 003120
1940 012340
    012340 000137 012352
1941 012344
    012344 000402
    012346
1942 012346

```

```

BLE 3$
JMP 4$
3$:
LET FLAG := #0
CLR FLAG
PRINTF #PAPRSW,LUNIT
MOV LUNIT,-(SP)
MOV #PAPRSW,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD #6,SP
PRINTF #PAPSW1
MOV #PAPSW1,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD #4,SP
GMANIL READY, FLAG, 100000, YES
TRAP C$GMAN
BR 10000$
.WORD FLAG
.WORD T$CODE
.WORD READY
.WORD 100000
10000$:
LET LINCNT := #200. ; ALLOW FOR ABOUT 3 PAGES OF PAPER
MOV #200.,LINCNT
LET ERRFLG := #0
CLR ERRFLG
REPEAT
50127$:
OUTPUT #PAPTST,#15.,#5$,LUNIT
LET LINCNT := LINCNT - #1
DEC LINCNT
UNTIL LINCNT EQ #0 OR ERRFLG NE #0
TST LINCNT
BEQ 50130$
TST ERRFLG
BEQ 50127$
50130$:
IF ERRFLG EQ #0 THEN
TST ERRFLG
BNE 50131$
ERRHRD 8,PAPSWI
TRAP C$ERHRD
.WORD 8
.WORD PAPSWI
.WORD 0
LET ERRTBL(R2) := ERRTBL(R2) + #1
INC ERRTBL(R2)
INLINE <JMP 1002$>
JMP 11002$
ELSE
BR 50132$
50131$:
LET ERRFLG := #0

```

## READY LINE INTERLOCKS TEST 2

1943	012346	005037	002344	CLR	ERRFLG	
	012352			ENDIF		
1944	012352			50132\$:		
	012352			11002\$:	PRINTF #PAPRDY,LUNIT	
	012352	013746	002320		MOV LUNIT,-(SP)	
	012356	012746	013605		MOV #PAPRDY,-(SP)	
	012362	012746	000002		MOV #2,-(SP)	
	012366	010600			MOV SP,R0	
	012370	104417			TRAP C\$PNTF	
	012372	062706	000006		ADD #6,SP	
1945	012376				LET FLAG := #0	
	012376	005037	002300		CLR FLAG	
1946	012402				GMANIL READY,FLAG,100000,YES	
	012402	104443			TRAP C\$GMAN	
	012404	000404			BR 10001\$	
	012406	002300			.WORD FLAG	
	012410	000130			.WORD T\$CODE	
	012412	007673			.WORD READY	
	012414	100000			.WORD 100000	
	012416			10001\$:		
1947	012416				LET R2 := LUNIT SHIFT 1	
	012416	013702	002320		MOV LUNIT,R2	
	012422	006302			ASL R2	
1948	012424				LET @LPCSR(R2) := #0 ; RESET THE LP CSR	
	012424	005072	002362		CLR @LPCSR(R2)	
1949	012430	000137	012112		JMP 2\$	
1950					;EXPECTED ERROR HANDLER.	
1951					;JUST SET EXPECTED ERROR INDICATOR.	
1952					;	
1953	012434			5\$:	LET ERRFLG := #1	
	012434	012737	000001	002344	MOV #1,ERRFLG	
1954	012442				LET ERRCOD := #0	
	012442	005037	002342		CLR ERRCOD	
1955	012446				LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR!ACTIVE	
	012446	042762	120000	002516	BIC #ERROR!ACTIVE,STATUS(R2)	
1956	012454				LET CURCNT(R2) := #0 ; CLEAN UP THE DRIVER PARAMETERS.	
	012454	005062	002756		CLR CURCNT(R2)	
1957	012460				LET CURADD(R2) := #0	
	012460	005062	002556		CLR CURADD(R2)	
1958	012464				LET REPCNT(R2) := #0	
	012464	005062	002656		CLR REPCNT(R2)	
1959	012470	000207			RTS PC ;AND RETURN	
1960					;VERIFY OPERATION OF HAMMER BANK INTERLOCK SWITCH ON LP25,26 -OR- VERIFY BAND GATE LATCH ON	
LP07 OR LP27.						
1961	012472			4\$:	INCR LUNIT FROM #0 TO R1 BY #1	
	012472	005037	002320		CLR LUNIT	
	012476	000402			BR 50133\$	
	012500			50134\$:		
	012500	005237	002320		INC LUNIT	
	012504			50133\$:		
	012504	023701	002320		CMP LUNIT,R1	
	012510	003127			BGT 50135\$	
1962	012512				LET R2 := LUNIT SHIFT 1	
	012512	013702	002320		MOV LUNIT,R2	
	012516	006302			ASL R2	
1963	012520				LET L\$LUN := LUNIT	
	012520	013737	002320	002074	MOV LUNIT,L\$LUN	
1964	012526				LET FLAG := #0	

## READY LINE INTERLOCKS TEST 2

```

012526 005037 002300 CLR FLAG
1965 012532 IF #FLAG07 SETIN STATUS(R2) THEN
012532 032762 002000 002516 BIT #FLAG07,STATUS(R2)
012540 001413 BEQ 50136$
1966 012542 PRINTF #BGTLCH,LUNIT
012542 013746 002320 MOV LUNIT,-(SP)
012546 012746 014325 MOV #BGTLCH,-(SP)
012552 012746 000002 MOV #2,-(SP)
012556 010600 MOV SP,R0
012560 104417 TRAP C$PNTF
012562 062706 000006 ADD #6,SP
1967 012566 ELSE
012566 000422 BR 50137$
012570 50136$:
1968 012570 PRINTF #HAMRSW,LUNIT
012570 013746 002320 MOV LUNIT,-(SP)
012574 012746 013714 MOV #HAMRSW,-(SP)
012600 012746 000002 MOV #2,-(SP)
012604 010600 MOV SP,R0
012606 104417 TRAP C$PNTF
012610 062706 000006 ADD #6,SP
1969 012614 PRINTF #HAMSW1
012614 012746 013771 MOV #HAMSW1,-(SP)
012620 012746 000001 MOV #1,-(SP)
012624 010600 MOV SP,R0
012626 104417 TRAP C$PNTF
012630 062706 000004 ADD #4,SP
1970 012634 ENDF
012634 50137$:
1971 012634 GMANIL READY, FLAG, 100000, YES
012634 104443 TRAP C$GMAN
012636 000404 BR 10002$
012640 002300 .WORD FLAG
012642 000130 .WORD T$CODE
012644 007673 .WORD READY
012646 100000 .WORD 100000
012650 10002$:
1972 012650 IF #BIT15 SETIN @LPCSR(R2) THEN
012650 032772 100000 002362 BIT #BIT15,@LPCSR(R2)
012656 001421 BEQ 50140$
1973 012660 PRINTF #HAMRDY,LUNIT
012660 013746 002320 MOV LUNIT,-(SP)
012664 012746 014032 MOV #HAMRDY,-(SP)
012670 012746 000002 MOV #2,-(SP)
012674 010600 MOV SP,R0
012676 104417 TRAP C$PNTF
012700 062706 000006 ADD #6,SP
1974 012704 GMANIL READY, FLAG, 100000, YES
012704 104443 TRAP C$GMAN
012706 000404 BR 10003$
012710 002300 .WORD FLAG
012712 000130 .WORD T$CODE
012714 007673 .WORD READY
012716 100000 .WORD 100000
012720 10003$:
1975 012720 ELSE
012720 000422 BR 50141$

```

READY LINE INTERLOCKS TEST 2

```

1976 012722          50140$:
012722          INC      LET ERRTBL(R2) := ERRTBL(R2) + #1
005262 003120          ERRTBL(R2)
1977 012726          013737 002320 002074          MOV      LET L$LUN := LUNIT
012726          LUNIT,L$LUN
1978
1979 012734          IF #FLAG07 SETIN STATUS(R2) THEN
012734 032762 002000 002516          BIT      #FLAG07,STATUS(R2)
012742 001405          BEQ      50142$
1980 012744          ERRHRD 17,BGTSWI
012744 104456          TRAP    C$ERHRD
012746 000021          .WORD   17
012750 004221          .WORD   BGTSWI
012752 000000          .WORD   0
1981 012754          ELSE
012754 000404          BR      50143$
012756          50142$:
1982 012756          ERRHRD 9,BNKSWI
012756 104456          TRAP    C$ERHRD
012760 000011          .WORD   9
012762 003515          .WORD   BNKSWI
012764 000000          .WORD   0
1983 012766          ENDIF
012766          50143$:
1984 012766          ENDIF
012766          50141$:
1985 012766          ENDINC
012766 000644          BR      50134$
012770          50135$:
1986          ;VERIFY OPERATION OF CHARACTER BAND INTERLOCK SWITCH ON LP25,26 - OR - VFU INTERLOCK ON LP07
OR LP27.
1987 012770          INCR LUNIT FROM #0 TO R1 BY #1
012770 005037 002320          CLR      LUNIT
012774 000402          BR      50144$
012776          50145$:
012776 005237 002320          INC      LUNIT
013002          50144$:
013002 023701 002320          CMP      LUNIT,R1
013006 003162          BGT      50146$
1988 013010          LET R2 := LUNIT SHIFT 1
013010 013702 002320          MOV      LUNIT,R2
013014 006302          ASL      R2
1989 013016          LET FLAG := #0
013016 005037 002300          CLR      FLAG
1990 013022          IF #FLAG07 SETIN STATUS(R2) AND #FLAG26 NOTSETIN STATUS(R2) THEN
013022 032762 002000 002516          BIT      #FLAG07,STATUS(R2)
013030 001417          BEQ      50147$
013032 032762 001000 002516          BIT      #FLAG26,STATUS(R2)
013040 001013          BNE      50147$
1991 013042          PRINTF #VFUINT,LUNIT
013042 013746 002320          MOV      LUNIT,-(SP)
013046 012746 014425          MOV      #VFUINT,-(SP)
013052 012746 000002          MOV      #2,-(SP)
013056 010600          MOV      SP,R0
013060 104417          TRAP    C$PNTF
013062 062706 000006          ADD      #6,SP
1992 013066          ELSE
013066 000422          BR      50150$

```

## READY LINE INTERLOCKS TEST 2

1993	013070				50147\$:		
	013070	013746	002320			PRINTF #BANDSW,LUNIT	
	013074	012746	014116		MOV	LUNIT,-(SP)	
	013100	012746	000002		MOV	#BANDSW,-(SP)	
	013104	010600			MOV	#2,-(SP)	
	013106	104417			MOV	SP,R0	
	013110	062706	000006		TRAP	C\$PNTF	
1994	013114				ADD	#6,SP	
	013114	012746	014204			PRINTF #BNDW1	
	013120	012746	000001		MOV	#BNDW1,-(SP)	
	013124	010600			MOV	#1,-(SP)	
	013126	104417			MOV	SP,R0	
	013130	062706	000004		TRAP	C\$PNTF	
1995	013134				ADD	#4,SP	
	013134				ENDIF		
1996	013134				50150\$:		
	013134	104443			GMANIL	READY, FLAG, 100000, YES	
	013136	000404			TRAP	C\$GMAN	
	013140	002300			BR	10004\$	
	013142	000130			.WORD	FLAG	
	013144	007673			.WORD	T\$CODE	
	013146	100000			.WORD	READY	
	013150				.WORD	100000	
1997	013150				10004\$:		
	013150	032772	100000	002362	IF	#BIT15 SETIN @LPCSR(R2) THEN	
	013156	001444			BIT	#BIT15,@LPCSR(R2)	
	013160				BEQ	50151\$	
1998	013160					IF #FLAG07 SETIN STATUS(R2) AND #FLAG26 NOTSETIN STATUS(R2) THEN	
	013160	032762	002000	002516	BIT	#FLAG07,STATUS(R2)	
	013166	001417			BEQ	50152\$	
	013170	032762	001000	002516	BIT	#FLAG26,STATUS(R2)	
	013176	001013			BNE	50152\$	
1999	013200					PRINTF #VFURDY,LUNIT	
	013200	013746	002320		MOV	LUNIT,-(SP)	
	013204	012746	014517		MOV	#VFURDY,-(SP)	
	013210	012746	000002		MOV	#2,-(SP)	
	013214	010600			MOV	SP,R0	
	013216	104417			TRAP	C\$PNTF	
	013220	062706	000006		ADD	#6,SP	
2000	013224					ELSE	
	013224	000412			BR	50153\$	
	013226				50152\$:		
2001	013226					PRINTF #BNDRDY,LUNIT	
	013226	013746	002320		MOV	LUNIT,-(SP)	
	013232	012746	014234		MOV	#BNDRDY,-(SP)	
	013236	012746	000002		MOV	#2,-(SP)	
	013242	010600			MOV	SP,R0	
	013244	104417			TRAP	C\$PNTF	
	013246	062706	000006		ADD	#6,SP	
2002	013252					ENDIF	
	013252				50153\$:		
2003	013252				GMANIL	READY, FLAG, 100000, YES	
	013252	104443			TRAP	C\$GMAN	
	013254	000404			BR	10005\$	
	013256	002300			.WORD	FLAG	
	013260	000130			.WORD	T\$CODE	
	013262	007673			.WORD	READY	

READY LINE INTERLOCKS TEST 2

```

013264 100000          .WORD 100000
013266
2004 013266          10005$: ELSE
013266 000426          BR 50154$
013270          50151$:
2005 013270          LET ERRTBL(R2) := ERRTBL(R2) + #1
013270 005262 003120  INC ERRTBL(R2)
2006 013274          LET L$LUN := LUNIT
013274 013737 002320 002074 MOV LUNIT,L$LUN
2007 013302          IF #FLAG07 SETIN STATUS(R2) AND #FLAG26 NOTSETIN STATUS(R2) THEN
013302 032762 002000 002516 BIT #FLAG07,STATUS(R2)
013310 001411          BEQ 50155$
013312 032762 001000 002516 BIT #FLAG26,STATUS(R2)
013320 001005          BNE 50155$
2008 013322          ERRHRD 16,VFUINF
013322 104456          TRAP C$ERHRD
013324 000020          .WORD 16
013326 004263          .WORD VFUINF
013330 000000          .WORD 0
2009 013332          ELSE
013332 000404          BR 50156$
013334          50155$:
2010 013334          ERRHRD 10, BNDSWI
013334 104456          TRAP C$ERHRD
013336 000012          .WORD 10
013340 003562          .WORD BNDSWI
013342 000000          .WORD 0
2011 013344          ENDIF
013344          50156$:
2012 013344          ENDIF
013344          50154$:
2013 013344          LET @LPCSR(R2) := #00
013344 012772 000000 002362 MOV #00,@LPCSR(R2)
2014 013352          ENDINC
013352 000611          BR 50145$
013354          50146$:
2015 013354          LET OUTBUF := #14
013354 012737 000014 003164 MOV #14,OUTBUF
2016 013362          OUTPUT #OUTBUF,#1
2017 013424          EXIT TST
013424 104432          TRAP C$EXIT
013426 001162          .WORD L10011-.

```

```

2018
2019          .NLIST BEX
2020
2021 013430          122      105      101 INTLK: .ASCIZ /READY LINE INTERLOCK TEST 2/<12><12>
2022 013466          045      116      045 PAPRSW: .ASCIZ /#N#ATEAR OFF PAPER JUST BELOW LUNIT #D2/
2023 013536          045      101      040 PAPSW1: .ASCIZ /#A TO CHECK PAPER LOW #N#AINTERLOCK.#N/
2024 013605          045      116      045 PAPRDY: .ASCIZ /#N#ARESTORE PAPER, CLEAR, PLACE LUNIT #D2#A ON LINE.#N/
2025 013674          120      101      120 PAPTST: .ASCIZ /PAPER LOW TEST/<12>
2026 013714          045      116      045 HAMRSW: .ASCIZ /#N#ADISENGAGE HAMMER BANK LATCH ON LUNIT #D2/
2027 013771          045      116      045 HAMSW1: .ASCIZ /#N#ATO CHECK INTERLOCK SWITCH.#N/
2028 014032          045      116      045 HAMRDY: .ASCIZ /#N#AENGAGE LATCH,CLEAR,PLACE LUNIT #D2#A ON LINE.#N/
2029 014116          045      116      045 BANDSW: .ASCIZ /#N#AOPEN CHARACTER BAND COVER ON LUNIT #D2#A TO CHECK/
2030 014204          045      116      045 BNDSW1: .ASCIZ /#N#AINTERLOCK SWITCH.#N/
2031 014234          045      116      045 BNDRDY: .ASCIZ /#N#ACLOSE BAND COVER ON LUNIT #D2#A,CLEAR,PLACE ON LINE./
2032 014325          045      116      045 BGTLCH: .ASCIZ /#N#AOPEN BAND GATE LATCH ON LUNIT #D2#A TO CHECK MICROSWITCH.#N/

```



READY LINE INTERLOCKS TEST 2

```
2033 014425      045      116      045 VFUINT: .ASCIZ /#N#ALIFT VFU COVER ON LUNIT #D2#A TO CHECK MICROSWITCH.#N/
2034 014517      045      116      045 VFURDY: .ASCIZ /#N#ACLOSE VFU COVER ON LUNIT #D2#A,CLEAR,PLACE ONLINE.#N/
2035
2036
2037
2038 014610
      014610
      014610 104401
2039
2040 014612
2041
```

.LIST BEX  
ENDTST  
L10011: TRAP C#ETST  
ENDMOD

## FORMS LENGTH SELECTION

2043				.SBTTL FORMS LENGTH SELECTION
2044	014612			BGNMOD
2045				;++
2046				;THIS TEST CHECKS ALL POSITIONS OF THE FORM LENGTH SELECT SWITCH. THE
2047				;PROGRAM INDICATES THE SPECIFIED SETTING OF THE FORM LENGTH SELECT SWITCH
2048				;AND WAITS FOR THE OPERATOR TO SET THE SWITCH ON THE PRINTER. THE PAPER
2049				;IS THEN ADVANCED UNDER PROGRAM CONTROL. THE PRINTER OUTPUT IS VISUALLY
2050				;INSPECTED AFTER ALL SWITCH SETTINGS HAVE BEEN RUN THROUGH BY THE OPERATOR
2051				;TO VERIFY THAT THE PROPER PAPER MOVEMENT HAS OCCURRED FOR EACH SWITCH
2052				;SETTING. ALL LP07'S OR LP27'S WILL BE TESTED ON ALL TCVFU CHANNELS AND LINE COUNTS.
2053				;--
2054	014612			BGNTST 3
	014612			T3::
2055				;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
2056	014612			MANUAL
	014612	104450		TRAP C\$MANI
2057	014614			BCOMplete 1\$
	014614	103402		BCS 1\$
2058	014616			EXIT TST
	014616	104432		TRAP C\$EXIT
	014620	003076		.WORD L10012-
2059				;EXIT TEST IF MANUAL INTERVENTION TESTS ARE NOT SPECIFIED
2060	014622	005737	002266	1\$: TST INHINT
2061	014626	001002		BNE 2\$
2062	014630			EXIT TST
	014630	104432		TRAP C\$EXIT
	014632	003064		.WORD L10012-
2063	014634			2\$: LET R1 := L\$UNIT - #1
	014634	013701	002012	MOV L\$UNIT,R1
	014640	005301		DEC R1
2064		000001		\$BRJMP=1
2065	014642			INCR LUNIT FROM #0 TO R1 BY #1
	014642	005037	002320	CLR LUNIT
	014646	000402		BR 50160\$
	014650			50157\$: INC LUNIT
	014650	005237	002320	50160\$: CMP LUNIT,R1
	014654			BLE 50161\$
	014654	023701	002320	JMP 50162\$
	014660	003402		50161\$: LET R2 := LUNIT SHIFT 1
	014662	000137	016436	MOV LUNIT,R2
	014666			ASL R2
2066	014666			IF #FLAG07 NOTSETIN STATUS(R2) THEN
	014666	013702	002320	#FLAG07,STATUS(R2)
	014672	006302		.+6
2067	014674			JMP 50163\$
	014674	032762	002000 002516	PRINTF @LINSWI,LUNIT
	014702	001402		MOV LUNIT,-(SP)
	014704	000137	015354	MOV @LINSWI,-(SP)
2068	014710			MOV #2,-(SP)
	014710	013746	002320	MOV SP,R0
	014714	012746	016504	TRAP C\$PNTF
	014720	012746	000002	ADD #6,SP
	014724	010600		PRINTF @LINSW1
	014726	104417		MOV @LINSW1,-(SP)
	014730	062706	000006	
2069	014734			
	014734	012746	016570	

## FORMS LENGTH SELECTION

	014740	012746	000001		MOV	#1,-(SP)	
	014744	010600			MOV	SP,R0	
	014746	104417			TRAP	C\$PNTF	
	014750	062706	000004		ADD	#4,SP	
2070	014754					PRINTF	#FLSSEL,LUNIT
	014754	013746	002320		MOV	LUNIT,-(SP)	
	014760	012746	016650		MOV	#FLSSEL,-(SP)	
	014764	012746	000002		MOV	#2,-(SP)	
	014770	010600			MOV	SP,R0	
	014772	104417			TRAP	C\$PNTF	
	014774	062706	000006		ADD	#6,SP	
2071	015000					PRINTF	#FLS1,LUNIT
	015000	013746	002320		MOV	LUNIT,-(SP)	
	015004	012746	016747		MOV	#FLS1,-(SP)	
	015010	012746	000002		MOV	#2,-(SP)	
	015014	010600			MOV	SP,R0	
	015016	104417			TRAP	C\$PNTF	
	015020	062706	000006		ADD	#6,SP	
2072	015024					PRINTF	#FLSMS1,LUNIT
	015024	013746	002320		MOV	LUNIT,-(SP)	
	015030	012746	017443		MOV	#FLSMS1,-(SP)	
	015034	012746	000002		MOV	#2,-(SP)	
	015040	010600			MOV	SP,R0	
	015042	104417			TRAP	C\$PNTF	
	015044	062706	000006		ADD	#6,SP	
2073	015050					INCR R3 FROM #0 TO #50 BY #4	
	015050	005003			CLR	R3	
	015052	000402			BR	50165\$	
	015054			50164\$:			
	015054	062703	000004		ADD	#4,R3	
	015060			50165\$:			
	015060	020327	000050		CMP	R3,#50	
	015064	003402			BLE	50166\$	
	015066	000137	015350		JMP	50167\$	
	015072			50166\$:			
2074	015072					LET T3SET := #FFSET + R3	
	015072	012737	017275	016442	MOV	#FFSET,T3SET	
	015100	060337	016442		ADD	R3,T3SET	
2075	015104					PRINTF	#FLSMMSG,LUNIT,T3SET
	015104	013746	016442		MOV	T3SET,-(SP)	
	015110	013746	002320		MOV	LUNIT,-(SP)	
	015114	012746	017351		MOV	#FLSMMSG,-(SP)	
	015120	012746	000003		MOV	#3,-(SP)	
	015124	010600			MOV	SP,R0	
	015126	104417			TRAP	C\$PNTF	
	015130	062706	000010		ADD	#10,SP	
2076	015134					PRINTF	#FLSMS1
	015134	012746	017443		MOV	#FLSMS1,-(SP)	
	015140	012746	000001		MOV	#1,-(SP)	
	015144	010600			MOV	SP,R0	
	015146	104417			TRAP	C\$PNTF	
	015150	062706	000004		ADD	#4,SP	
2077	015154					LET FLAG := #0	
	015154	005037	002300		CLR	FLAG	
2078	015160					GMANIL READY,FLAG,100000,YES	
	015160	104443			TRAP	C\$GMAN	
	015162	000404			BR	10000\$	

FORMS LENGTH SELECTION

	015164	002300			.WORD	FLAG	
	015166	000130			.WORD	T#CODE	
	015170	007673			.WORD	READY	
	015172	100000			.WORD	100000	
	015174			100000\$:			
2079	015174						
2080	015236						OUTPUTI #REFLIN,#62...LUNIT ;TEST ID + REF. LINE
2081	015300						OUTPUTI T3SET,#3,,LUNIT
2082	015342	004737	005606				OUTPUTI #MOVMSG,#62...LUNIT
2083	015346						JSR PC,QUIET
	015346	000642			BR	50164\$	ENDINC
	015350			50167\$:			
2084	015350						ELSE
	015350	000137	016142		JMP	50170\$	
	015354			50163\$:			
2085	015354						OUTPUTI #REFLIN,#30...LUNIT ;TEST ID
2086	015416	004737	005606				JSR PC,QUIET
2087	015422						OUTPUTI #CHNSECT,#13...LUNIT ;SECTION ID
2088	015464						LET WORK1 := #52 ;CODE FOR ASTERISK
	015464	012737	000052	003162	MOV	#52,WORK1	
2089	015472						LET OUTBUF := #200 ;CHANNEL CODE
	015472	012737	000200	003164	MOV	#200,OUTBUF	
2090	015500						INCR WORK FROM #1 TO #12. BY #1 ;REPEAT FOR ALL 12 CHANNELS
	015500	012737	000001	003160	MOV	#1,WORK	
	015506	000402			BR	50172\$	
	015510			50171\$:			
	015510	005237	003160		INC	WORK	
	015514			50172\$:			
	015514	023727	003160	000014	CMP	WORK,#12.	
	015522	003402			BLE	50173\$	
	015524	000137	015642		JMP	50174\$	
	015530			50173\$:			
2091	015530						OUTPUTI #OUTBUF,#1,,LUNIT
2092	015572						OUTPUTI #WORK1,#1,,LUNIT,WORK
2093	015634						LET OUTBUF := OUTBUF + #1 ;NEXT CHANNEL
	015634	005237	003164		INC	OUTBUF	
2094	015640						ENDINC
	015640	000723			BR	50171\$	
	015642			50174\$:			
2095	015642						LET OUTBUF := #14
	015642	012737	000014	003164	MOV	#14,OUTBUF	
2096	015650						OUTPUTI #OUTBUF,#1,,LUNIT
2097	015712						OUTPUTI #LINSECT,#15...LUNIT ;SECTION ID
2098	015754						LET WORK1 := #52 ;CODE FOR ASTERISK
	015754	012737	000052	003162	MOV	#52,WORK1	
2099	015762						LET OUTBUF := #220 ;CODE FOR VFU
	015762	012737	000220	003164	MOV	#220,OUTBUF	
2100	015770						INCR WORK FROM #0 TO #15. BY #1
	015770	005037	003160		CLR	WORK	
	015774	000402			BR	50176\$	
	015776			50175\$:			
	015776	005237	003160		INC	WORK	
	016002			50176\$:			
	016002	023727	003160	000017	CMP	WORK,#15.	
	016010	003402			BLE	50177\$	
	016012	000137	016142		JMP	50200\$	
	016016			50177\$:			

## FORMS LENGTH SELECTION

```

2101 016016
2102 016060          OUTPUT #OUTBUF,#1,,LUNIT
                016060 005737 003160          TST      WORK
                016064 003002                BGT      .+6
                016066 000137 016134          JMP      50201$
2103 016072
2104 016134          ENDIF      OUTPUT #WORK1,#1,,LUNIT,WORK ;PRINT ASTERISK
                016134          50201$:
2105 016134          INC      OUTBUF          LET OUTBUF := OUTBUF + #1
                016134 005237 003164          ENDINCR
2106 016140          BR      50175$
                016140 000716          50200$:
                016142          ENDIF
2107 016142          50170$:
                016142          ;SET FORMS LENGTH SELECT SWITCH TO ITS "REGULAR" SETTING
2108
2109 016142          IF #FLAG07 NOTSETIN STATUS(R2) THEN
                016142 032762 002000 002516    BIT      #FLAG07,STATUS(R2)
                016150 001402                BEQ      .+6
                016152 000137 016322          JMP      50202$
2110 016156          PRINTF #NMLFLS,LUNIT
                016156 013746 002320          MOV      LUNIT,-(SP)
                016162 012746 017203          MOV      #NMLFLS,-(SP)
                016166 012746 000002          MOV      #2,-(SP)
                016172 010600          MOV      SP,R0
                016174 104417          TRAP    C$PNTF
                016176 062706 000006          ADD      #6,SP
2111 016202          PRINTF #FLSMS1,LUNIT
                016202 013746 002320          MOV      LUNIT,-(SP)
                016206 012746 017443          MOV      #FLSMS1,-(SP)
                016212 012746 000002          MOV      #2,-(SP)
                016216 010600          MOV      SP,R0
                016220 104417          TRAP    C$PNTF
                016222 062706 000006          ADD      #6,SP
2112 016226          LET OUTBUF := #14
                016226 012737 000014 003164    MOV      #14,OUTBUF
2113 016234          LET FLAG := #0 ;CLEAR <CR> FLAG
                016234 005037 002300          CLR      FLAG
2114 016240          GMANIL READY,FLAG,100000,YES ;AND WAIT FOR RESPONSE
                016240 104443          TRAP    C$GMAN
                016242 000404          BR      10001$
                016244 002300          .WORD  FLAG
                016246 000130          .WORD  T$CODE
                016250 007673          .WORD  READY
                016252 100000          .WORD  100000
                016254          10001$:
2115 016254          OUTPUT #OUTBUF,#1,,LUNIT
2116 016316          ELSE
                016316 000137 016372          JMP      50203$
                016322          50202$:
2117 016322          LET OUTBUF := #14
                016322 012737 000014 003164    MOV      #14,OUTBUF
                016330          OUTPUT #OUTBUF,#1,,LUNIT
2118 016330          ENDIF
2119 016372          50203$:
                016372
2120 016372          PRINTF #PAPCHK ;MAKE SURE MOVEMENT WAS RIGHT
                016372 012746 017042          MOV      #PAPCHK,-(SP)

```

FORMS LENGTH SELECTION

```

016376 012746 000001      MOV    #1,-(SP)
016402 010600      MOV    SP,R0
016404 104417      TRAP   C$PNTF
016406 062706 000004      ADD    #4,SP
2121 016412      LET FLAG := #0
016412 005037 002300      CLR    FLAG
2122 016416      GMANIL READY,FLAG,100000,YES ;WAIT FOR RESPONSE
016416 104443      TRAP   C$GMAN
016420 000404      BR     10002$
016422 002300      .WORD FLAG
016424 000130      .WORD T$CODE
016426 007673      .WORD READY
016430 100000      .WORD 100000
016432      10002$:
2123 016432      ENDINC
016432 000137 014650      JMP    5( 157$
016436      50162$:
2124 016436 177777      $BRJMP=-1
2125 016436      EXIT TST
016436 104432      TRAP   C$EXIT
016440 001256      .WORD L10012-.
2126 016442 00C000      T3SET: .WORD 0
2127 016444 000000      T3MOV: .WORD 0
2128      .NLIST BEX
2129 016446      103      110      101 CHNSEC: .ASCIZ /CHANNEL 1-12/<12>
2130 016464      114      111      116 LINSEC: .ASCIZ /LINE COUNT 0-15/
2131 016504      045      116      045 LINSWI: .ASCIZ /%N$ASET LINES SWITCH ON UNIT %D2%A TO "6" TO SELECT/
2132 016570      045      116      045 LINSW1: .ASCIZ /%N$A6 LINES PER INCH VERTICAL PRINTING DENSITY./
2133 016650      045      116      062 FLSEL: .ASCIZ /%N2$ASET VFU-FLS SWITCH ON UNIT %D2%A TO THE "FLS" POSITION,%N/
2134 016747      045      116      045 FLS1: .ASCIZ /%N$ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO "8".%N/
2135 017042      045      116      045 PAPCHK: .ASCIZ /%N$AVERIFY PROPER PAPER MOVEMENT./
2136 017104      106      117      122 REFLIN: .ASCIZ /FORMS LENGTH SELECTION TEST #3---REFERENCE LINE...../<14>
2137 017203      045      116      045 NMLFLS: .ASCIZ /%N$ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO 11.%N/
2138      ;SWITCH SETTINGS FOR FORMS LENGTH MESSAGES
2139 017275      063      040      040 FFSET: .ASCIZ /3 /
2140 017301      063      056      065      .ASCIZ /3.5/
2141 017305      064      040      040      .ASCIZ /4 /
2142 017311      065      056      065      .ASCIZ /5.5/
2143 017315      066      040      040      .ASCIZ /6 /
2144 017321      067      040      040      .ASCIZ /7 /
2145 017325      070      040      040      .ASCIZ /8 /
2146 017331      070      056      065      .ASCIZ /8.5/
2147 017335      061      061      040      .ASCIZ /11 /
2148 017341      061      062      040      .ASCIZ /12 /
2149 017345      061      064      040      .ASCIZ /14 /
2150 017351      045      116      045 FLMSG: .ASCIZ /%N$ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO %T%A,/
2151 017443      045      116      045 FLMS1: .ASCIZ /%N$ADEPRESS "CLEAR","TOF", AND PLACE ON LINE.%N/
2152 017523      045      116      045 FLMS2: .ASCIZ /%N$ADEPRESS "CLEAR","TOF",AND PLACE ON LINE,ON LUNIT %D2%A/
2153 017616      040      111      116 MOVMSG: .ASCIZ / INCHES SHOULD OCCUR BETWEEN THIS AND THE REFERENCE LINE...../<12>
2154      .EVEN
2155      .EVEN
2156      .LIST BEX
2157 017716      ENDTST
017716      L10012:
017716 104401      TRAP   C$ETST
2158 017720      ENDMOD
2159

```

## PRINTING SPEED MEASUREMENT

```

2161          .SBTTL PRINTING SPEED MEASUREMENT
2162 017720    BGNMOD
2163          ;**
2164          ; THE PRINT SPEED TEST WILL REPORT TO THE OPERATOR THE TOTAL NUMBER OF
2165          ; LINES PRINTED WITHIN A SPECIFIED TIME PERIOD.  THE DATA PATTERN USED
2166          ; IS DESIGNED TO CAUSE PRINTING SPEED TO BE MINIMAL AND IS DEPENDENT
2167          ; ON PRINTER TYPE AND THE CHARACTER SET (BAND TYPE ) ON EACH PRINTER.
2168          ; THE TIME PERIOD CAN BE CONTROLLED THRU MANUAL OPERATION, OR IF THE
2169          ; SYSTEM HAS A CLOCK VIA SUPPLYING A COUNT OF SECONDS.  ANY TIME INTERVAL OF
2170          ; 4 TO 60 SECONDS MAY BE SELECTED.  THIS IS ONE OF THE "SW" QUESTIONS .
2171          ;--
2172 017720    BGNTST 4
2173          T4::
2174          ;
2174 017720    LET R1 := L$UNIT - #1                ;NUMBER OF UNITS TO TEST
2175          017720 013701 002012                MOV     L$UNIT,R1
2175          017724 005301                        DEC     R1
2175 017726    IF MANSPPD NE #0 THEN                ; DETERMIN IF MANUAL TESTING SELECTED
2176          017726 005737 002270                TST     MANSPPD
2176          017732 001416                        BEQ     50204$
2176 017734    MANUAL                                ;DETERMINE IF MANUAL INTERVENTION ALLOWED
2177          017734 104450                        TRAP    C$MANI
2177 017736    BCOMPLETE 1$
2178          017736 103402                        BCS     1$
2178 017740    EXIT TST
2179          017740 104432                        TRAP    C$EXIT
2179          017742 006042                        .WORD  L10013-.
2179 017744    1$: IF INHINT EQ #0 THEN                ; EXIT IF INTERVENTION INHIBITED
2180          017744 005737 002266                TST     INHINT
2180          017750 001003                        BNE     50205$
2180 017752    EXIT TST
2181          017752 104432                        TRAP    C$EXIT
2181          017754 006030                        .WORD  L10013-.
2181 017756    ELSE
2182          017756 000403                        BR      50206$
2182 017760    50205$:
2183          017760 013737 002272 003160          LET WORK := PERIOD
2183          017766                                MOV     PERIOD,WORK
2183          017766                                ENDIF
2184 017766    50206$:
2184          017766 000403                        ELSE
2185          017770                                BR      50207$
2185 017770    50204$:
2186          017770 013737 002272 003160          LET WORK := PERIOD
2186          017776                                MOV     PERIOD,WORK
2186          017776                                ; CLOCK TEST TIME
2187 017776    ENDIF
2188 020040    50207$:
2188          020040 013746 002326                OUTPUT #PRTSPD,#36.
2188          020044 002502                        SELECT CLKTYP OF 4 VERIFY
2188          020046 023727 002326 000004          MOV     CLKTYP,-(SP)
2188          020054 003076                        BLT     50216$
2188          020056 006316                        CMP     CLKTYP,#4
2188          020060 062716 020066                BGT     50216$
2188          020064 013607                        ASL     (SP)
2188          ADD     #50210$,(SP)
2188          MOV     @ (SP)+,PC
2188          ; DISPATCH TABLE

```

## PRINTING SPEED MEASUREMENT

```

020066          50210$:
020066 020256          .WORD 50215$ ; CASE 0
020070 020100          .WORD 50214$ ; CASE 1
020072 020106          .WORD 50213$ ; CASE 2
020074 020160          .WORD 50212$ ; CASE 3
020076 020246          .WORD 50211$ ; CASE 4

2189
2190 020100          CASE 1
020100          50214$:
2191 020100 000137 022142      JMP      END4          ;JUST EXIT TEST NO CLOCK AVAILBLE
2192
2193 020104          CASE 2          ;KW11-L LINE CLOCK SELECTED
020104 000464          BR      50217$
2194 020106          50213$:
020106 012737 000100 002340      LET CLKENA := #100          ;INTERRUPT ENABLE/ CLR MONITOR
                                MOV      #100,CLKENA
                                ; SET PRI7 WHILE CHANGING VECTOR ADDRESS
2195          SETPRI #PRI07
2196 020114          MOV      #PRI07,R0
020114 012700 000340          TRAP   C$SPRI
020120 104441
2197 020122          SETVEC CLKVEC,#CLKTCK,#PRI06      ;SET UP INTERRUPT VECTOR
020122 012746 000300          MOV      #PRI06,-(SP)
020126 012746 037566          MOV      #CLKTCK,-(SP)
020132 013746 002336          MOV      CLKVEC,-(SP)
020136 012746 000003          MOV      #3,-(SP)
020142 104437          TRAP   C$SVEC
020144 062706 000010          ADD      #10,SP
2198 020150          SETPRI #PRI00
020150 012700 000000          MOV      #PRI00,R0
020154 104441          TRAP   C$SPRI

2199
2200 020156          CASE 3          ;KW11-P REAL TIME CLOCK
020156 000437          BR      50217$
020160
2201 020160          50212$:
020160 013737 002332 002334      LET CLKSET := CLKCSR + #2
020166 062737 000002 002334      MOV      CLKCSR,CLKSET
                                ADD      #2,CLKSET
2202 020174          LET CLKENA := #111          ;SET UP ENABLE BITS
020174 012737 000111 002340      MOV      #111,CLKENA
                                ; RUN, RATE = 10KHZ, REPEAT INTR, DOWN,INT ENABLE
2203          SETPRI #PRI07
2204 020202          MOV      #PRI07,R0
020202 012700 000340          TRAP   C$SPRI
020206 104441
2205 020210          SETVEC CLKVEC,#CLKTCK,#PRI06      ;INTERRUPT VECTOR
020210 012746 000300          MOV      #PRI06,-(SP)
020214 012746 037566          MOV      #CLKTCK,-(SP)
020220 013746 002336          MOV      CLKVEC,-(SP)
020224 012746 000003          MOV      #3,-(SP)
020230 104437          TRAP   C$SVEC
020232 062706 000010          ADD      #10,SP
2206 020236          SETPRI #PRI00
020236 012700 000000          MOV      #PRI00,R0
020242 104441          TRAP   C$SPRI

2207
2208 020244          CASE 4
020244 000404          BR      50217$
020246          50211$:

```



## PRINTING SPEED MEASUREMENT

```

2209 020246 000240      NOP                ;THIS IS JUST A DUMMY
2210 020250      ENDSELECT
      020250 000402      BR          50217$
      020252      50216$:
      020252 062706 000002      ADD          #2,SP
      020256      50215$:
      020256      50217$:
2211 020256      LET OUTBUF :B= #LF
      020256 112737 000012 003164      MOVB         #LF,OUTBUF
2212 020264      LET LUNIT := R1
      020264 010137 002320      MOV          R1,LUNIT
2213 020270      11$:
2214 020270      LET ERRFLG := #0
      020270 005037 002344      CLR          ERRFLG
2215 020274      LET R2 := LUNIT SHIFT 1
      020274 013702 002320      MOV          LUNIT,R2
      020300 006302      ASL          R2
2216
2217      ;
2218      ; DETERMINE WHICH BAND, AND SEND APPROPRIATE PATTERN
2219      ;
2219      000001      $BRJMP=1
2220 020302      IF #FLAG96 NOTSETIN STATUS(R2) THEN ; 64 CHAR BAND
      020302 032762 010000 002516      BIT          #FLAG96,STATUS(R2)
      020310 001402      BEQ          .+6
      020312 000137 021214      JMP          50220$
2221 020316      IF #FLAG07 NOTSETIN STATUS(R2) THEN
      020316 032762 002000 002516      BIT          #FLAG07,STATUS(R2)
      020324 001402      BEQ          .+6
      020326 000137 020344      JMP          50221$
2222 020332      LET BNDPAT := #TAB64
      020332 012737 024754 023426      MOV          #TAB64,BNDPAT
2223 020340      MOV
      020340 000137 020352      ELSE
      020344      JMP          50222$
2224 020344      50221$:
      020344 012737 025370 023426      LET BNDPAT := #TB0764
      MOV          #TB0764,BNDPAT
2225 020352      ENDIF
      020352      50222$:
2226 020352      LET WORK := #133.
      020352 012737 000205 003160      MOV          #133.,WORK
2227 020360      IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG07 NOTSETIN STATUS(R2) THEN
      020360 032762 001000 002516      BIT          #FLAG26,STATUS(R2)
      020366 001004      BNE          50223$
      020370 032762 002000 002516      BIT          #FLAG07,STATUS(R2)
      020376 001402      BEQ          .+6
      020400      50223$:
2228 020400      JMP          50224$
      020404 012746 024071      PRINTF #LPM64          ;SEND SPEED MESSAGE TO CONSOLE
      020410 012746 000001      MOV          #LPM64,-(SP)
      020414 010600      MOV          #1,-(SP)
      020416 104417      MOV          SP,R0
      020420 062706 000004      TRAP         C$PNTF
      020424      ADD          #4,SP
2229 020424      OUTPUTI #LPM64+4,#42...,LUNIT      ;SEND SPEED MESSAGE TO PRINTER
2230 020466      OUTPUTI #OUTBUF,#1.,LUNIT      ;LINEFEED
2231 020530      ELSE
      020530 000137 021210      JMP          50225$

```

## PRINTING SPEED MEASUREMENT

```

020534
2232 020534      032762 001000 002516      50224$:      IF #FLAG26 SETIN STATUS(R2) AND #FLAG07 NOTSETIN STATUS(R2) THEN
020534      032762 001000 002516      BIT          #FLAG26,STATUS(R2)
020542      001404                      BEQ          50226$
020544      032762 002000 002516      BIT          #FLAG07,STATUS(R2)
020552      001402                      BEQ          .+6
020554
020554      000137 020710      50226$:      JMP          50227$
2233 020560      012746 024233      PRINTF #L26M64
020564      012746 000001      MOV          #L26M64,-(SP)
020570      010600                      MOV          #1,-(SP)
020572      104417                      MOV          SP,R0
020574      062706 000004      TRAP        C$PNTF
                                ADD          #4,SP
2234 020600
2235 020642      OUTPUTI #L26M64+4,#42,,LUNIT
2236 020704      OUTPUTI #OUTBUF,#1,,LUNIT
020704      000137 021210      JMP          ELSE
020710      50227$:      50230$
2237 020710      IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG07 SETIN STATUS(R2) THEN
020710      032762 001000 002516      BIT          #FLAG26,STATUS(R2)
020716      001004                      BNE          50231$
020720      032762 002000 002516      BIT          #FLAG07,STATUS(R2)
020726      001002                      BNE          .+6
020730
020730      000137 021064      50231$:      JMP          50232$
2238 020734      012746 024375      PRINTF #L07M64
020740      012746 000001      MOV          #L07M64,-(SP)
020744      010600                      MOV          #1,-(SP)
020746      104417                      MOV          SP,R0
020750      062706 000004      TRAP        C$PNTF
                                ADD          #4,SP
2239 020754      OUTPUTI #L07M64+4,#42,,LUNIT
2240 021016      OUTPUTI #OUTBUF,#1,,LUNIT
2241 021060
021060      000137 021210      JMP          ELSE
021064      50232$:      50233$
2242 021064      012746 024537      PRINTF #L27M64
021070      012746 000001      MOV          #L27M64,-(SP)
021074      010600                      MOV          #1,-(SP)
021076      104417                      MOV          SP,R0
021100      062706 000004      TRAP        C$PNTF
                                ADD          #4,SP
2243 021104      OUTPUTI #L27M64+4,#42,,LUNIT
2244 021146      OUTPUTI #OUTBUF,#1,,LUNIT
2245 021210
021210      50233$:      ENDIF
2246 021210      50230$:      ENDIF
021210      50225$:      ENDIF
2247 021210      ELSE
021210      000137 022106      JMP          50234$
2248 021214      50220$:
021214      032762 002000 002516      IF #FLAG07 NOTSETIN STATUS(R2) THEN
021222      001402                      BIT          #FLAG07,STATUS(R2)
                                BEQ          .+6

```

## PRINTING SPEED MEASUREMENT

```

021224 000137 021242          JMP      50235$
2250 021230          LET BNDPAT := #TABA96
021230 012737 025162 023426    MOV      #TABA96,BNDPAT
2251 021236          ELSE
021236 000137 021250          JMP      50236$
021242          50235$:
2252 021242          LET BNDPAT := #TB0796
021242 012737 025576 023426    MOV      #TB0796,BNDPAT
2253 021250          ENDIF
021250          50236$:
2254 021250          LET WORK := #133.
021250 012737 000205 003160    MOV      #133.,WORK
2255 021256          IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG07 NOTSETIN STATUS(R2) THEN
021256 032762 001000 002516    BIT      #FLAG26,STATUS(R2)
021264 001004          BNE      50237$
021266 032762 002000 002516    BIT      #FLAG07,STATUS(R2)
021274 001402          BEQ      .+6
021276          50237$:
2256 021276 000137 021432    JMP      50240$
021302          PRINTF #LPM96
021302 012746 024152    MOV      #LPM96,-(SP)
021306 012746 000001    MOV      #1,-(SP)
021312 010600          MOV      SP,R0
021314 104417          TRAP    C$PNTF
021316 062706 000004    ADD      #4,SP
2257 021322          OUTPUTI #LPM96+4,#42.,,LUNIT
2258 021364          OUTPUTI #OUTBUF,#1.,,LUNIT
2259 021426          ELSE
021426 000137 022106    JMP      50241$
021432          50240$:
2260 021432          IF #FLAG26 SETIN STATUS(R2) AND #FLAG07 NOTSETIN STATUS(R2) THEN
021432 032762 001000 002516    BIT      #FLAG26,STATUS(R2)
021440 001404          BEQ      50242$
021442 032762 002000 002516    BIT      #FLAG07,STATUS(R2)
021450 001402          BEQ      .+6
021452          50242$:
2261 021452 000137 021606    JMP      50243$
021456          PRINTF #L26M96
021456 012746 024314    MOV      #L26M96,-(SP)
021462 012746 000001    MOV      #1,-(SP)
021466 010600          MOV      SP,R0
021470 104417          TRAP    C$PNTF
021472 062706 000004    ADD      #4,SP
2262 021476          OUTPUTI #L26M96+4,#42.,,LUNIT
2263 021540          OUTPUTI #OUTBUF,#1.,,LUNIT
2264 021602          ELSE
021602 000137 022106    JMP      50244$
021606          50243$:
2265 021606          IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG07 SETIN STATUS(R2) THEN
021606 032762 001000 002516    BIT      #FLAG26,STATUS(R2)
021614 001004          BNE      50245$
021616 032762 002000 002516    BIT      #FLAG07,STATUS(R2)
021624 001002          BNE      .+6
021626          50245$:
2266 021626 000137 021762    JMP      50246$
021632          PRINTF #L07M96
021632 012746 024456    MOV      #L07M96,-(SP)

```

## PRINTING SPEED MEASUREMENT

```

021636 012746 000001      MOV    #1,-(SP)
021642 010600      MOV    SP,R0
021644 104417      TRAP  C$PNTF
021646 062706 000004      ADD   #4,SP
2267 021652      OUTPUTI #L07M96+4,#42,,LUNIT
2268 021714      OUTPUTI #OUTBUF,#1,,LUNIT
2269 021756      ELSE
021756 000137 022106      JMP   50247$
021762      50246$:
2270 021762      PRINTF #L27M96
021762 012746 024620      MOV   #L27M96,-(SP)
021766 012746 000001      MOV   #1,-(SP)
021772 010600      MOV   SP,R0
021774 104417      TRAP  C$PNTF
021776 062706 000004      ADD   #4,SP
2271 022002      OUTPUTI #L27M96+4,#42,,LUNIT
2272 022044      OUTPUTI #OUTBUF,#1,,LUNIT
2273 022106      ENDIF
022106      50247$:
2274 022106      ENDIF
022106      50244$:
2275 022106      ENDIF
022106      50241$:
2276 022106      ENDIF
022106      50234$:
2277 022106      LET @LPCSR(R2) := #0      ;DISABLE INTERRUPTS IN CASE OF MANUAL TEST CASE
022106 005072 002362      CLR   @LPCSR(R2)
2278 177777      $BRJMP=-1
2279 022112      LET LINCNT := #0      ;CLEAR LINE COUNTER
022112 005037 002302      CLR   LINCNT
2280 022116      LET TICK := #60.      ;SET UP INITIAL CLOCK VALUE
022116 012737 000074 037642      MOV   #60.,TICK
2281 022124 004737 022330      JSR  PC,REPLUP      ;DO THE OUTPUT
2282 022130      LET LUNIT := LUNIT - #1
022130 005337 002320      DEC   LUNIT
2283 022134      IFCOND GE THEN
022134 002402      BLT   50250$
2284 022136 000137 020270      JMP  11$
2285 022142      ENDIF
022142      50250$:
2286
2287
2288 022142      END4: IF CLKTYP EQ #3 THEN
022142 023727 002326 000003      CMP   CLKTYP,#3
022150 001011      BNE   50251$
2289 022152      SETPRI #PRI07
022152 012700 000340      MOV   #PRI07,R0
022156 104441      TRAP  C$SPRI
2290 022160      CLRVEC CLKVEC
022160 013700 002336      MOV   CLKVEC,R0
022164 104436      TRAP  C$CVEC
2291 022166      LET @CLKCSR := #00
022166 012777 000000 160136      MOV   #00,@CLKCSR
2292 022174      ENDIF
022174      50251$:
2293 022174      IF CLKTYP EQ #2 THEN
022174 023727 002326 000002      CMP   CLKTYP,#2

```

## PRINTING SPEED MEASUREMENT

```

022202 001021      BNE      50252$
2294 022204      SETPRI #PRI07
022204 012700 000340  MOV      #PRI07,R0
022210 104441      TRAP     C$SPRI
2295 022212      SETVEC CLKVEC,#IGNORE,#PRI06
022212 012746 000300  MOV      #PRI06,-(SP)
022216 012746 010070  MOV      #IGNORE,-(SP)
022222 013746 002336  MOV      CLKVEC,-(SP)
022226 012746 000003  MOV      #3,-(SP)
022232 104437      TRAP     C$SVEC
022234 062706 000010  ADD      #10,SP
2296 022240      LET @CLKCSR := #00
022240 012777 000000 160064  MOV      #00,@CLKCSR
2297 022246      ENDIF
022246      50252$:
2298 022246      SETPRI #PRI00
022246 012700 000000  MOV      #PRI00,R0
022252 104441      TRAP     C$SPRI
2299 022254      LET OUTBUF :B= #14
022254 112737 000014 003164  MOV      #14,OUTBUF
2300 022262      OUTPUT #OUTBUF,#1
2301 022324      EXIT TST
022324 104432      TRAP     C$EXIT
022326 003456      .WORD   L10013-.
2302      ;
2303      ;
2304      ;THIS IS SUBROUTINED TO DECREASE THE SIZE OF THE INITIAL INCREMENT LOOP.
2305      ;
2306      ;
2307 022330      REPLUP:
2308 022330      IF CLKTYP EQ #4 THEN
022330 023727 002326 000004  CMP      CLKTYP,#4
022336 001124      BNE      50253$
2309 022340      PRINTF #OFFLIN      ;TELL OPERATOR TO PLACE PRINTERS OFFLINE
022340 012746 023534  MOV      #OFFLIN,-(SP)
022344 012746 000001  MOV      #1,-(SP)
022350 010600      MOV      SP,R0
022352 104417      TRAP     C$PNTF
022354 062706 000004  ADD      #4,SP
2310 022360      LET FLAG := #0
022360 005037 002300  CLR      FLAG
2311 022364      GMANIL READY,FLAG,100000,YES
022364 104443      TRAP     C$GMAN
022366 000404      BR      10000$
022370 002300      .WORD   FLAG
022372 000130      .WORD   T$CODE
022374 007673      .WORD   READY
022376 100000      .WORD   100000
2312 022400      10000$:
022400      PRINTF #ONLIN1,LUNIT      ;PUT LUNIT TO TEST ON LINE
022400 013746 002320  MOV      LUNIT,-(SP)
022404 012746 023574  MOV      #ONLIN1,-(SP)
022410 012746 000002  MOV      #2,-(SP)
022414 010600      MOV      SP,R0
022416 104417      TRAP     C$PNTF
022420 062706 000006  ADD      #6,SP
2313 022424      PRINTF #ONLIN2,LUNIT      ;END OF TEST.

```

## PRINTING SPEED MEASUREMENT

	022424	013746	002320		MOV	LUNIT,-(SP)	
	022430	012746	023675		MOV	#ONLIN2,-(SP)	
	022434	012746	000002		MOV	#2,-(SP)	
	022440	010600			MOV	SP,R0	
	022442	104417			TRAP	C#PNTF	
	022444	062706	000006		ADD	#6,SP	
2314	022450					PRINTF #ONLIN3,LUNIT	
	022450	013746	002320		MOV	LUNIT,-(SP)	
	022454	012746	023773		MOV	#ONLIN3,-(SP)	
	022460	012746	000002		MOV	#2,-(SP)	
	022464	010600			MOV	SP,R0	
	022466	104417			TRAP	C#PNTF	
	022470	062706	000006		ADD	#6,SP	
2315	022474					WHILE #BIT15 SETIN @LPCSR(R2) DO ; WAIT FOR LP SET ON-LINE	
	022474			50254\$:			
	022474	032772	100000	002362	BIT	#BIT15,@LPCSR(R2)	
	022502	001402			BEQ	50255\$	
2316	022504	000240				NOP	
2317	022506					ENDDO	
	022506	000772			BR	50254\$	
	022510			50255\$:			
2318	022510					LET LINCNT := #0	
	022510	005037	002302		CLR	LINCNT	
2319	022514					WHILE #BIT15 NOTSETIN @LPCSR(R2) DO ; REPEAT UNTIL LP GOES OFF-LINE	
	022514			50256\$:			
	022514	032772	100000	002362	BIT	#BIT15,@LPCSR(R2)	
	022522	001031			BNE	50257\$	
2320	022524					LET R5 := BNDPAT	
	022524	013705	023426		MOV	BNDPAT,R5	
2321	022530					LET R3 := WORK	
	022530	013703	003160		MOV	WORK,R3	
2322	022534					WHILE R3 GT #0 DO ; PRINT R3 CHARACTERS	
	022534			50260\$:			
	022534	005703			TST	R3	
	022536	003417			BLE	50261\$	
2323	022540					WHILE #BIT7 NOTSETIN @LPCSR(R2) DO ; WAIT FOR READY	
	022540			50262\$:			
	022540	032772	000200	002362	BIT	#BIT7,@LPCSR(R2)	
	022546	001007			BNE	50263\$	
2324	022550					IF #BIT15 SETIN @LPCSR(R2) THEN	
	022550	032772	100000	002362	BIT	#BIT15,@LPCSR(R2)	
	022556	001402			BEQ	50264\$	
2325	022560	000137	022722			JMP 99\$ ; EXIT LOOP IF OFF-LINE AGAIN	
2326	022564					ENDIF	
	022564			50264\$:			
	022564					ENDDO	
	022564	000765			BR	50262\$	
	022566			50263\$:			
2328	022566					LET @LPBUF(R2) :B= (R5)+ ; PUT CHAR INTO LP BUFFER	
	022566	112572	002456		MOVB	(R5)+,@LPBUF(R2)	
2329	022572					LET R3 := R3 - #1 ; DECREMENT CHAR COUNTER	
	022572	005303			DEC	R3	
2330	022574					ENDDO	
	022574	000757			BR	50260\$	
	022576			50261\$:			
2331	022576					BREAK ; ALLOW CTL-C ABORT	
	022576	104422			TRAP	C#BRK	

## PRINTING SPEED MEASUREMENT

```

2332 022600          LET LINCNT := LINCNT + #1
      022600 005237 002302      INC LINCNT
2333 022604          ENDDO
      022604 000743          BR 50256#
2334 022606          50257# :
      022606 000445          ELSE
      022610          BR 50265#
2335 022610          50253# :
      022610 023727 002326 000003      IF CLKTYP EQ #3 THEN
      022616 001003          CMP CLKTYP,#3
2336 022620          BNE 50266#
      022620 012777 003202 157506      MOV LET @CLKSET := #1666. ; 1/60 SEC.
2337 022626          @1666.,@CLKSET
      022626          ENDDO
2338 022626          50266# :
      022626 013777 002340 157476      MOV LET @CLKCSR := CLKENA ;ENABLE THE CLOCK TO DO ITS THING
2339 022634          CLKENA,@CLKCSR
      022634 005037 037640          LET TIME := #0
2340 022640          CLR TIME
      022640 005037 002302          CLR LET LINCNT := #0
2341 022644          LINCNT
      022644          WHILE TIME LT PERIOD DO ; REPEAT UNTIL TIME EXHAUSTED
2342 022652          50267# :
      022652 002023          CMP TIME,PERIOD
2343 022654          BGE 50270#
      022654 013705 023426          MOV LET R5 := BNDPAT
2344 022660          BNDPAT,R5
      022660 013703 003160          MOV LET R3 := WORK
2345 022664          WORK,R3
      022664          WHILE R3 GT #0 DO ; SEND R3 CHARACTERS
2346 022664          50271# :
      022664 005703          TST R3
2347 022666          003412          BLE 50272#
2348 022670          WHILE #BIT7 NOTSETIN @LPCSR(R2) DO ; WAIT FOR READY
      022670 032772 000200 002362      50273# :
      022676 001002          BIT #BIT7,@LPCSR(R2)
2349 022700          BNE 50274#
2349 022702          NOP
      022702 000772          ENDDO
2349 022704          BR 50273#
2349 022704          50274# :
      022704 112572 002456          MOVB LET @LPBUF(R2) :B= (R5). ; PUT DATA INTO BUFFER
2350 022710          (R5),@LPBUF(R2)
      022710 005303          DEC LET R3 := R3 - #1 ; DECREMENT CHAR COUNTER
2350 022712          R3
      022712 000764          ENDDO
2351 022714          BR 50271#
2351 022714          50272# :
      022714 005237 002302          LET LINCNT := LINCNT + #1
2352 022720          INC LINCNT
      022720 000751          ENDDO
2353 022722          BR 50267#
2353 022722          50270# :
2354 022722          ENDDO
2354 022722          50265# :
2355          99# :
          ;

```

## PRINTING SPEED MEASUREMENT

```

2356 ;IF MANUAL PRINT SPEED TESTS HAVE BEEN PERFORMED INSURE PRINTERS ARE
2357 ;BACK ON LINE WHEN DONE
2358 ;
2359 022722 IF CLKTYP EQ #4 THEN
      022722 023727 002326 000004      CMP      CLKTYP,#4
      022730 001020                      BNE      50275$
2360 022732 LET FLAG := #0                      ;CLEAR <CR> FLAG
      022732 005037 002300          CLR      FLAG
2361 022736 PRINTF #RESTOR
      022736 012746 024701          MOV      #RESTOR,-(SP)
      022742 012746 000001          MOV      #1,-(SP)
      022746 010600          MOV      SP,RO
      022750 104417          TRAP     C:PNTF
      022752 062706 000004          ADD      #4,SP
2362 022756 GMANIL READY,FLAG,100000,YES          ;WAIT FOR OPERATOR
      022756 104443          TRAP     C:GMAN
      022760 000404          BR       10001$
      022762 002300          .WORD   FLAG
      022764 000130          .WORD   T:CODE
      022766 007673          .WORD   READY
      022770 100000          .WORD   100000
      022772          10001$:
2363 022772 ENDF
      022772          50275$:
2364 022772 012777 000000 157332      MOV      #00,@CLKCSR
2365 ;
2366 ; REPORT TOTAL NUMBER OF LINES PRINTED
2367 ;
2368 023000 PRINTB #LINPER,LINCNT,LUNIT
      023000 013746 002320      MOV      LUNIT,-(SP)
      023004 013746 002302      MOV      LINCNT,-(SP)
      023010 012746 023467      MOV      #LINPER,-(SP)
      023014 012746 000003      MOV      #3,-(SP)
      023020 010600          MOV      SP,RO
      023022 104414          TRAP     C:PNTB
      023024 062706 000010      ADD      #10,SP
2369 023030 PUSH LINCNT,#OUTBUF+1          ; CONVERT LINE COUNT TO ASCII
      023030 013746 002302      MOV      LINCNT,-(SP)
      023034 012746 003165      MOV      #OUTBUF+1,-(SP)
2370 023040 004737 004562          JSR      PC,BIN2DA
2371 023044 WHILE #BIT7 NOTSETIN @LPCSR(R2) DO          ;WAIT FOR READY
      023044          50276$:
      023044 032772 000200 002362      BIT      #BIT7,@LPCSR(R2)
      023052 001001          BNE      50277$
2372 023054 ENDDO
      023054 000773          BR       50276$
      023056          50277$:
2373 023056 OUTPUTI #OUTBUF,#6,..LUNIT          ; DISPLAY LINE COUNT ON LP
2374 023120 004737 005606          JSR      PC,QUIET
2375 023124 004737 005606          JSR      PC,QUIET
2376 023130 OUTPUTI #SPED1,#19,..LUNIT          ; "LINES PRINTED"
2377 0231 2 004737 005606          JSR      PC,QUIET
2378 ; IF A CLOCK WAS USED DISPLAY THE TIME USED ALSO
2379 023176 IF CLKTYP EQ #2 OR CLKTYP EQ #3 THEN
      023176 023727 002326 000002      CMP      CLKTYP,#2
      023204 001404          BEQ      50300$
      023206 023727 002326 000003      CMP      CLKTYP,#3

```



PRINTING SPEED MEASUREMENT

```

023214 001077          BNE      50301$
023216          50300$:
2380 023216          OUTPUTI #SPED2,#4,,LUNIT      ; " IN "
2381 023260 004737 005606      JSR PC,QUIET
2382 023264          PUSH PERIOD,#OUTBUF+1      ; CONVERT TIME TO ASCII
          023264 013746 002272      MOV PERIOD,-(SP)
          023270 012746 003165      MOV #OUTBUF+1,-(SP)
2383 023274 004737 004562      JSR PC,BIN2DA
2384 023300          OUTPUTI #OUTBUF+3,#3,,LUNIT ; DISPLAY THE TIME IN SECONDS
2385 023342 004737 005606      JSR PC,QUIET
2386 023346          OUTPUTI #SPED3,#8,,LUNIT ; "SECONDS"
2387 023410 004737 005606      JSR PC,QUIET
2388 023414          ENDF
          023414          50301$:
2389 023414 000207          RTS PC ;GO BACK AND DO IT AGAIN
2390          ;
2391          ;EXPECTED ERROR HANDLER
2392          ;
2393 023416          LPERR2: LET ERRFLG := #1 ;SET ERROR FOUND
          023416 012737 000001 002344      MOV #1,ERRFLG
2394 023424 000207          RTS PC ;AND EXIT
2395          ;
2396          ;
2397 023426 000000          BNDPAT: .WORD 0 ; CONTAINS ADDRESS OF PRINT PATTERN
2398          .NLIST BEX
2399          ;
2400          ;ASSOCIATED MESSAGES
2401          ;
2402 023430          040 114 111 SPED1: .ASCII / LINES WERE PRINTED/
2403 023453          040 111 116 SPED2: .ASCII / IN /
2404 023457          040 123 105 SPED3: .ASCII / SECONDS/
2405 023467          045 116 045 LINPER: .ASCIZ /%D4%A LINES PRINTED ON LUNIT %D2%N/
2406 023534          045 116 045 OFFLIN: .ASCIZ /%AINSURE PRINTER(S) OFF LINE./
2407 023574          045 116 045 ONLIN1: .ASCIZ /%APLACE LUNIT %D2%A ON LINE TO INITIATE TIME PERIOD FOR MANUAL/
2408 023675          045 116 045 ONLIN2: .ASCIZ /%APRINTING SPEED MEASUREMENT AND BACK OFF LINE TO TERMINATE/
2409 023773          045 116 045 ONLIN3: .ASCIZ /%ATHE TIME INTERVAL.%N/
2410 024024          120 122 111 PRTSPD: .ASCIZ /PRINTING SPEED MEASUREMENT TEST 4/<12><12><12>
2411 024071          045 116 045 LPM64: .ASCIZ /%A64 CHARACTER BAND SHOULD PRINT AT 285 LPM.%N/
2412 024152          045 116 045 LPM96: .ASCIZ /%A96 CHARACTER BAND SHOULD PRINT AT 204 LPM.%N/
2413 024233          045 116 045 L26M64: .ASCIZ /%A64 CHARACTER BAND SHOULD PRINT AT 600 LPM.%N/
2414 024314          045 116 045 L26M96: .ASCIZ /%A96 CHARACTER BAND SHOULD PRINT AT 450 LPM.%N/
2415 024375          045 116 045 L07M64: .ASCIZ /%A64 CHARACTER BAND SHOULD PRINT AT 1220 LPM.%N/
2416 024456          045 116 045 L07M96: .ASCIZ /%A96 CHARACTER BAND SHOULD PRINT AT 888 LPM.%N/
2417 024537          045 116 045 L27M64: .ASCIZ /%A64 CHARACTER BAND SHOULD PRINT AT 1200 LPM.%N/
2418 024620          045 116 045 L27M96: .ASCIZ /%A96 CHARACTER BAND SHOULD PRINT AT 800 LPM.%N/
2419 024701          045 116 045 RESTOR: .ASCIZ /%ACLEAR PRINTER(S) AND PLACE ON LINE.%N/
2420          .LIST BEX
2421          .EVEN
2422          ;64 CHARACTER BAND PATTERN 285 LPM / 600 LPM.
2423          ;
2424          .SBTTL PRINT SPEED TEST PATTERNS
2425          .NLIST BEX
2426 024754          105 061 104 TABA64: .BYTE 105,061,104,075,064,041,103,136,102,060,163
2427 024767          042 062 134 .BYTE 042,062,134,054,124,101,133,101,133,043,135
2428 025002          041 105 061 .BYTE 041,105,061,100,075,077,041,056,136,074,060
2429 025015          076 042 073 .BYTE 076,042,073,042,073,134,055,124,044,133,057
2430 025030          135 054 105 .BYTE 135,054,105,072,100,050,077,052,056,051,056
    
```

PRINT SPEED TEST PATTERNS

2431	025043	051	074	046	.BYTE	051,074,046,076,071,073,045,055,053,044,137
2432	025056	057	070	054	.BYTE	057,070,054,132,072,131,072,131,050,067,052
2433	025071	130	051	127	.BYTE	130,051,127,046,066,071,126,045,125,053,065
2434	025104	137	123	137	.BYTE	137,123,137,123,070,122,132,121,131,064,067
2435	025117	120	130	117	.BYTE	120,130,117,124,063,066,116,126,115,126,115
2436	025132	125	062	065	.BYTE	125,062,065,114,123,113,122,061,121,112,064
2437	025145	111	120	110	.BYTE	111,120,110,117,060,117,060,063,107,116,106,012,015

2438  
2439  
2440  
2441 ;  
;96 CHARACTER BAND TABLE 204 LPM. / 450 LPM.  
; MINIMUM PRINT SPEED PATTERN 96 CHARACTER BAND  
TABA96: .BYTE 061,055,144,047,143,043,142,041,060,052,100  

2442	025162	061	055	144	.BYTE	061,055,144,047,143,043,142,041,060,052,100
2443	025175	075	140	174	.BYTE	075,140,174,176,041,056,054,056,054,136,042
2444	025210	176	134	173	.BYTE	176,134,173,133,175,135,055,164,047,100,043
2445	025223	077	041	074	.BYTE	077,041,074,041,074,052,062,075,076,174,073
2446	025236	041	053	054	.BYTE	041,053,054,071,042,057,134,072,133,050,133
2447	025251	050	135	051	.BYTE	050,135,051,164,070,100,046,124,045,123,044
2448	025264	122	067	064	.BYTE	122,067,064,137,073,132,073,132,053,131,071
2449	025277	066	057	130	.BYTE	066,057,130,072,127,050,120,151,125,070,065
2450	025312	046	124	046	.BYTE	046,124,046,124,045,123,044,122,067,064,137
2451	025325	121	132	120	.BYTE	121,132,120,131,117,066,063,130,116,130,116
2452	025340	127	115	126	.BYTE	127,115,126,114,125,113,065,062,124,112,123
2453	025353	111	122	110	.BYTE	111,122,110,064,061,064,061,121,107,102,106,012,015

2454  
2455  
2456 ;  
2457 ; 64 CHARACTER BAND PATTERN FOR LP07 AND LP27. 1220 LPM / 1200 LPM  

2458	025370	137	136	135	TB0764: .BYTE	137,136,135,134,133,132,131,130,127,126,125
2459	025403	124	123	122	.BYTE	124,123,122,121,120,117,116,115,114,113,112
2460	025416	111	110	107	.BYTE	111,110,107,106,105,104,103,102,101,100,077
2461	025431	076	075	074	.BYTE	076,075,074,073,072,071,070,067,066,065,064
2462	025444	063	062	061	.BYTE	063,062,061,060,057,056,055,054,053,052,051
2463	025457	050	047	046	.BYTE	050,047,046,045,044,043,042,041,137,136,135
2464	025472	134	133	132	.BYTE	134,133,132,131,130,127,126,125,124,123,122
2465	025505	121	120	117	.BYTE	121,120,117,116,115,114,113,112,111,110,107
2466	025520	106	105	104	.BYTE	106,105,104,103,102,101,100,077,076,075,074
2467	025533	073	072	071	.BYTE	073,072,071,070,067,066,065,064,063,062,061
2468	025546	060	057	056	.BYTE	060,057,056,055,054,053,052,051,050,047,046
2469	025561	045	044	043	.BYTE	045,044,043,042,041,137,136,135,134,132,131,012,015

2470  
2471  
2472 ;  
2473 ; 96 CHARACTER BAND PATTERN FOR LP07 AND LP27. 888 LPM / 800 LPM  

2474	025576	177	176	175	TB0796: .BYTE	177,176,175,174,173,172,171,170,167,166,165
2475	025611	164	163	162	.BYTE	164,163,162,161,160,157,156,155,154,153,152
2476	025624	151	150	147	.BYTE	151,150,147,146,145,144,143,142,141,140,137
2477	025637	136	135	134	.BYTE	136,135,134,133,132,131,130,127,126,125,124
2478	025652	123	122	121	.BYTE	123,122,121,120,117,116,115,114,113,112,111
2479	025665	110	107	106	.BYTE	110,107,106,105,104,103,102,101,100,077,076
2480	025700	075	074	073	.BYTE	075,074,073,072,071,070,067,066,065,064,063
2481	025713	062	061	060	.BYTE	062,061,060,057,056,055,054,053,052,051,050
2482	025726	047	046	045	.BYTE	047,046,045,044,043,042,041,177,176,175,174
2483	025741	173	172	171	.BYTE	173,172,171,170,167,166,165,164,163,162,161
2484	025754	160	157	156	.BYTE	160,157,156,155,154,153,152,151,150,147,146
2485	025767	145	144	143	.BYTE	145,144,143,142,141,140,137,136,135,134,133,012,015

2486 .EVEN  
2487 .LIST BEX

PRINT SPEED TEST PATTERNS

2488 026004  
026004  
026004 104401  
2489 026006

ENDTST  
L10013:  
TRAP C\$ETST  
ENDMOD

## DATA TRANSFER PATHS

```

2491          .SBTTL DATA TRANSFER PATHS
2492
2493 026006    BGNMOD
2494          ;**
2495          ;THIS TEST CHECKS THE DATA TRANSFER
2496          ;PATHS FROM THE PROCESSOR INTERFACE
2497          ;TO THE PRINTER OUTPUT. AN ALTERNATING
2498          ;PATTERN OF ONES AND ZEROES CORRESPONDING
2499          ;TO AN ALTERNATING STRING OF "*" AND
2500          ;"U" CHARACTERS ARE TRANSMITTED ON THE
2501          ;FULL 132 COLUMNS. AFTER 16 LINES OF
2502          ;THIS PATTERN, THE OUTPUT PATTERN IS
2503          ;SWITCHED TO AN ALTERNATING PATTERN
2504          ;OF "?" AND "a" CHARACTERS FOR ANOTHER
2505          ;16 LINES.
2506          ;--
2507
2508 026006    BGNTST 5
2509 026006    T5::
2510 026006    ;PRINT TEST IDENTIFICATION
2511          OUTPUT #DATPTH,#29.
2512 026050    ;PRINT ALTERNATING STRINGS OF CHARACTERS
026050 012737 000001 026426    INCR PATTERN FROM #1 TO #2 BY #1
026056 000402          MOV #1,PATTERN
026060          BR 50302$
026060 005237 026426    50303$:
026064          INC PATTERN
026064 023727 026426 000002    50302$:
026072 003107          CMP PATTERN,#2
026074          BGT 50304$
2513 026074          IF PATTERN EQ #1 THEN
026074 023727 026426 000001    CMP PATTERN,#1
026102 001004          BNE 50305$
2514 026104          LET CHAR :B= #'U
026104 112737 000125 026366    MOVB #'U,CHAR
2515 026112          ELSE
026112 000403          BR 50306$
2516 026114          50305$:
026114 112737 000077 026366    LET CHAR :B= #'?
2517 026122          MOVB #'?,CHAR
026122          ENDIF
2518 026122          50306$:
026122 012704 003164          LET R4 := #OUTBUF
2519 026126          MOV #OUTBUF,R4
026126 012737 000001 002310    INCR CCNT FROM #1 TO #66. BY #1
026134 000402          MOV #1,CCNT
026136          BR 50307$
026136 005237 002310    50310$:
026142          INC CCNT
026142 023727 002310 000102    50307$:
026150 003017          CMP CCNT,#66.
2520 026152          BGT 50311$
026152 113724 026366          LET (R4)+ :B= CHAR
2521 026156 105137 026366    MOVB CHAR,(R4)+
2522 026162          COMB CHAR
026162 142737 000200 026366    LET CHAR :B= CHAR CLR.BY #200
          BICB #200,CHAR

```

DATA TRANSFER PATHS

```

2523 026170          LET (R4)+ :B= CHAR
      026170 113724 026366      MOVB CHAR,(R4)+
2524 026174 105137 026366      COMB CHAR
2525 026200          LET CHAR :B= CHAR CLR.BY #200
      026200 142737 000200 026366      BICB #200,CHAR
2526 026206          ENDINC
      026206 000753          BR 50310$
      026210          50311$:
2527 026210          LET (R4)+ :B= #15
      026210 112724 000015      MOVB #15,(R4)+
2528 026214          LET (R4) :B= #12
      026214 112714 000012      MOVB #12,(R4)
2529 026220          INCR LINCNT FROM #1 TO #16. BY #1
      026220 012737 000001 002302      MOV #1,LINCNT
      026226 000402          BR 50312$
      026230          50313$:
      026230 005237 002302      INC LINCNT
      026234          50312$:
      026234 023727 002302 000020      CMP LINCNT,#16.
      026242 003022          BGT 50314$
2530 026244          OUTPUT #OUTBUF, #134.
2531 026306          ENDINC
      026306 000750          BR 50313$
      026310          50314$:
2532 026310          ENDINC
      026310 000663          BR 50303$
      026312          50304$:
2533 026312          LET OUTBUF :B= #14
      026312 112737 000014 003164      MOVB #14,OUTBUF
2534 026320          OUTPUT #OUTBUF, #1
2535 026362          EXIT TST
      026362 104432          TRAP C$EXIT
      026364 000044          .WORD L10014-.
2536          .NLIST BEX
2537 026366 000000          CHAR: .WORD 0
2538 026370 104 101 124      DATPTH: .ASCIZ /DATA TRANSFER PATHS TEST 5/ <12><12><12>
2539
2540          .EVEN
2541 026426 000000          PATTERN: .WORD 0
2542          .EVEN
2543          .EVEN
2544          .LIST BEX
2545
2546 026430          ENDTST
      026430          L10014:
      026430 104401          TRAP C$ETST
2547
2548 026432          ENDMOD
    
```

## PRINTABLE CHARACTERS

```

2550          .SBTTL PRINTABLE CHARACTERS
2551 026432   BGNMOD
2552          ;++
2553          ; THIS TEST WILL PRINT A FULL LINE OF EACH PRINTABLE CHARACTER.
2554          ; BAND TYPE IS CHECKED ON A UNIT BY UNIT BASIS.
2555          ; UNITS WITH 96 CHAR BAND WILL BE SENT THE CHARACTER CODES :
2556          ; 140(8) THRU 176(8).
2557          ;--
2558
2559 026432   BGNTST 6
          026432   T6::
2560 026432   OUTPUT #PRTCHR, #30.          ; PRINT TEST ID
2561          ;
2562          ; PRINT ALL UPPER CASE CHARACTERS ON ALL UNITS
2563          ;
2564 026474   INCR WORK FROM #40 TO #137 BY #1
          026474   012737 000040 003160   MOV #40,WORK
          026502   000402                   BR 50315$
          026504   50316$: INC WORK
          026510   023727 003160 000137   50315$: CMP WORK,#137
          026516   003045                   BGT 50317$
2565 026520   LET R4 := #OUTBUF
          026520   012704 003164   MOV #OUTBUF,R4
2566 026524   INCR COUNT FROM #1 TO #132. BY #1
          026524   012737 000001 002306   MOV #1,COUNT
          026532   000402                   BR 50320$
          026534   005237 002306   50321$: INC COUNT
          026540   023727 002306 000204   50320$: CMP COUNT,#132.
          026546   003003                   BGT 50322$
2567 026550   LET (R4)+ :B= WORK
          026550   113724 003160   MOVB WORK,(R4)+
2568 026554   ENDINC
          026554   000767   BR 50321$
          026556   50322$: LET (R4)+ :B= #LF
2569 026556   112724 000012   MOVB #LF,(R4)+
2570 026562   OUTPUT #OUTBUF,#133.
2571 026624   004737 005606   JSR PC,QUIET
2572 026630   ENDINC
          026630   000725   BR 50316$
          026632   50317$:
2573          ;
2574          ; NOW DO ALL THE LOWER CASE CHARACTERS ON THOSE UNITS
2575          ; EQUIPPED WITH 96 CHARACTER BANDS.
2576          ;
2577          ; FIRST DETERMINE IF ANY UNITS HAVE 96 CHAR BANDS
2578          ;
2579          ;
2580 026632   LET WORK := #0          ; COUNTER FOR 96 CHAR UNITS
          026632   005037 003160   CLR WORK
2581 026636   LET WORK1 := L$UNIT - #1          ; GET UNIT COUNT
          026636   013737 002012 003162   MOV L$UNIT,WORK1
          026644   005337 003162   DEC WORK1

```

## PRINTABLE CHARACTERS

```

2582 026650          INCR LUNIT FROM #0 TO WORK1 BY #1
      026650 005037 002320          CLR      LUNIT
      026654 000402          BR        50323$
      026656          50324$:
      026656 005237 002320          INC      LUNIT
      026662          50323$:
      026662 023737 002320 003162  CMP      LUNIT,WORK1
      026670 003012          BGT      50325$
2583 026672          LET R2 := LUNIT SHIFT 1
      026672 013702 002320          MOV      LUNIT,R2
      026676 006302          ASL      R2
2584 026700          IF #FLAG96 SETIN STATUS(R2) THEN ; IS THIS UNIT 96 CHAR ?
      026700 032762 010000 002516  BIT      #FLAG96,STATUS(R2)
      026706 001402          BEQ      50326$
2585 026710          LET WORK := WORK + #1 ; YES ADD 1 TO COUNT
      026710 005237 003160          INC      WORK
2586 026714          ENDF
      026714          50326$:
2587 026714          ENDINC
      026714 000760          BR        50324$
      026716          50325$:
2588 026716          IF WORK EQ #0 THEN ; ANY 96 CHAR UNITS ?
      026716 005737 003160          TST      WORK
      026722 001026          BNE      50327$
2589 026724          LET OUTBUF :B= #14
      026724 112737 000014 003164  MOVB    #14,OUTBUF
2590 026732          OUTPUT #OUTBUF,#1
2591 026774          EXIT TST ; ALL UNITS 64 CHAR...EXIT
      026774 104432          TRAP    C$EXIT
      026776 000330          .WORD   L10015-.
2592 027000          ENDF
      027000          50327$:
2593          ; SETUP FOR LOWER CASE CHARACTERS DISPLAY
2594          ;
2595 027000          INCR WORK FROM #140 TO #176 BY #1
      027000 012737 000140 003160  MOV      #140,WORK
      027006 000402          BR        50330$
      027010          50331$:
      027010 005237 003160          INC      WORK
      027014          50330$:
      027014 023727 003160 000176  CMP      WORK,#176
      027022 003073          BGT      50332$
2596 027024          LET R4 := #OUTBUF
      027024 012704 003164          MOV      #OUTBUF,R4
2597 027030          INCR COUNT FROM #1 TO #132. BY #1
      027030 012737 000001 002306  MOV      #1,COUNT
      027036 000402          BR        50333$
      027040          50334$:
      027040 005237 002306          INC      COUNT
      027044          50333$:
      027044 023727 002306 000204  CMP      COUNT,#132.
      027052 003003          BGT      50335$
2598 027054          LET (R4)+ :B= WORK
      027054 113724 003160          MOVB   WORK,(R4)+
2599 027060          ENDF
      027060 000767          BR        50334$
      027062          50335$:

```

## PRINTABLE CHARACTERS

```

2600 027062          LET (R4)+ :B= #LF
      027062 112724 000012      MOVB  #LF,(R4)+
2601 027066          LET WORK1 := L$UNIT - #1
      027066 013737 002012 003162      MOV  L$UNIT,WORK1
      027074 005337 003162      DEC  WORK1
2602 027100          INCR LUNIT FROM #0 TO WORK1 BY #1
      027100 005037 002320      CLR  LUNIT
      027104 000402          BR   50336$
      027106          50337$:      INC  LUNIT
      027106 005237 002320      50336$:
      027112          CMP  LUNIT,WORK1
      027112 023737 002320 003162      BGT  50340$
      027120 003031          LET R2 := LUNIT SHIFT 1
2603 027122          MOV  LUNIT,R2
      027122 013702 002320      ASL  R2
      027126 006302          IF #FLAG96 SETIN STATUS(R2) THEN
2604 027130          BIT  #FLAG96,STATUS(R2)
      027130 032762 010000 002516      BEQ  50341$
      027136 001421          OUTPUT #OUTBUF,#133.,,LUNIT
2605 027140          ENDIF
2606 027202          50341$:
      027202          ENDINC
2607 027202          BR   50337$
      027202 000741          50340$:
      027204          JSR PC,QUIET      ; WAIT FOR ALL DONE
2608 027204 004737 005606      ENDINC
2609 027210          BR   50331$
      027210 000677          50332$:
      027212          LET OUTBUF :B= #14
2610 027212 112737 000014 003164      MOVB  #14,OUTBUF
      027212          OUTPUT #OUTBUF,#1 ; EXECUTE TOF
2611 027220          EXIT TST
      027220 104432          TRAP  C$EXIT
2612 027262          .WORD  L10015-.
      027262 000042          .NLIST BEX
2613          .PRTCHR: .ASCIZ /PRINTABLE CHARACTERS TEST 6/ <12><12><12>
2614 027266 120 122 111
2615          .EVEN
2616
2617 027326          ENDTST
      027326          L10015:
      027326 104401          TRAP  C$ETST
2618
2619 027330          .LIST BEX
2620          ENDMOD

```



## NON-PRINTABLE CHARACTERS

```

2622          .SBTTL  NON-PRINTABLE CHARACTERS
2623
2624 027330   BGNMOD
2625          ;++
2626          ;THIS TEST CHECKS FOR DETECTION OF ALL NON-PRINTABLE CHARACTERS.
2627          ;EACH CHARACTER WILL APPEAR ON THE PRINTER OUTPUT IN THE FORM OF ITS OCTAL
2628          ;CODE ACCOMPANIED WITH ITS MNEMONIC.
2629          ;123 OF THE TESTED CODE ARE THEN SENT.
2630          ;ALSO, ON PRINTERS WITH 64 CHARACTER BANDS, IT CHECKS TO
2631          ;MAKE SURE THAT CODES 140 THRU 177 ARE CONVERTED TO CODES 100 THRU 137.
2632          ;--
2633
2634 027330   BGNTST 7
           027330   T7::
2635          ;INDICATE TEST CURRENTLY BEING DONE
2636
2637 027330   OUTPUT #NONCHR,#71.
2638 027372   LET R4 := #NONBUF
           027372   012704 030451   MOV #NONBUF,R4
2639 027376   LET WORK1 := #27.
           027376   012737 000033 003162   MOV #27.,WORK1
2640
2641          ; DO ONE LINE FOR EACH TABLE ENTRY
2642
2643 027404   INCR LINCNT FROM #0 TO WORK1 BY #1
           027404   005037 002302   CLR LINCNT
           027410   000402   BR 50342$
           027412   50343$:
           027412   005237 002302   INC LINCNT
           027416   50342$:
           027416   023737 002302 003162   CMP LINCNT,WORK1
           027424   003061   BGT 50344$
2644 027426   LET R3 := #OUTBUF
           027426   012703 003164   MOV #OUTBUF,R3
2645
2646          ; MOVE CODE AND MNEMONIC TO PRINT BUFFER
2647
2648 027432   INCR WORK FROM #1 TO #8. BY #1
           027432   012737 000001 003160   MOV #1,WORK
           027440   000402   BR 50345$
           027442   50346$:
           027442   005237 003160   INC WORK
           027446   50345$:
           027446   023727 003160 000010   CMP WORK,#8.
           027454   003002   BGT 50347$
2649 027456   LET (R3)+ :B= (R4)+
           027456   112423   MOVB (R4)+,(R3)+
2650 027460   ENDINC
           027460   000770   BR 50346$
           027462   50347$:
2651
2652          ; PUT 120 BYTES OF CODE INTO PRINT BUFFER
2653
2654
2655 027462   INCR WORK FROM #1 TO #123. BY #1
           027462   012737 000001 003160   MOV #1,WORK
           027470   000402   BR 50350$

```

## NON-PRINTABLE CHARACTERS

027472				50351\$:	INC	WORK	
027472	005237	003160		50350\$:	CMP	WORK,#123.	
027476					BGT	50352\$	
027476	023727	003160	000173			LET (R3)+ :B= (R4)	
2656	027504	003002			MOVB	(R4),(R3)+	
	027506	111423			BR	ENDINC	
2657	027510				BR	50351\$	
	027510	000770		50352\$:			
	027512						
2658							
2659							
2660							
2661							
2662	027512						
	027512	112723	000015			LET (R3)+ :B= #15	
2663	027516				MOVB	#15,(R3)+	
	027516	112723	000012			LET (R3)+ :B= #12	
2664					MOVB	#12,(R3)+	
2665							
2666							
2667	027522						
2668	027564						
	027564	005204					
2669	027566						
	027566	000711					
	027570						
2670							
2671							
2672							
2673							
2674							
2675							
2676	027570						
	027570	012703	003164				
2677	027574						
	027574	012704	000140				
	027600	000401					
	027602						
	027602	005204					
	027604						
	027604	020427	000177				
	027610	003002					
2678	027612						
	027612	110423					
2679	027614						
	027614	000772					
	027616						
2680	027616						
	027616	112723	000012				
2681	027622						
	027622	013737	002012	003160			
	027630	005337	003160				
2682	027634						
	027634	005037	002320				
	027640	000402					
	027642						

```

50351$: INC WORK
50350$: CMP WORK,#123.
BGT 50352$
LET (R3)+ :B= (R4)
MOVB (R4),(R3)+
BR ENDINC
50352$:
;
; FOLLOWED BY CRLF
;
LET (R3)+ :B= #15
MOVB #15,(R3)+
LET (R3)+ :B= #12
MOVB #12,(R3)+
;
; PRINT LINE OF OCTAL CODE, MNEMONIC, AND 120 BYTES(NONPRINTABLE CODE)
;
OUTPUT #OUTBUF,#133.
LET R4 := R4 + #1
INC R4
ENDINC
BR 50343$
50344$:
;
; UNITS WITH 64 CHAR BAND SHOULD STRIP BIT 6 OF DATA
; AND PRINT THE DATA FOR CODES 140(8) THRU 177(8)
; AS IF CODES 100(8) THRU 137(8) WERE RECIEVED.
; **NOTE** DELETE IS PRINTED AS UNDERSCORE '_'
;
LET R3 := #OUTBUF
MOV #OUTBUF,R3
INCR R4 FROM #140 TO #177 BY #1
MOV #140,R4
BR 50353$
50354$: INC R4
50353$: CMP R4,#177
BGT 50355$
LET (R3)+ :B= R4 ; FILL BUFFER WITH CODES & LF
MOVB R4,(R3)+
ENDINCR
BR 50354$
50355$: LET (R3)+ :B= #LF
MOVB #LF,(R3)+
LET WORK := L$UNIT - #1 ; SEND MSG AND BUFFER TO ALL
MOV L$UNIT,WORK
DEC WORK
INCR LUNIT FROM #0 TO WORK BY #1 ; UNITS WITH 64 CHAR BAND
CLR LUNIT
BR 50356$
50357$:

```

NON-PRINTABLE CHARACTERS

```

027642 005237 002320          INC      LUNIT
027646          50356$: CMP      LUNIT,WORK
027646 023737 002320 003160  BGT      50360$
027654 003073          LET R2 := LUNIT SHIFT 1
2683 027656          MOV      LUNIT,R2
027656 013702 002320          ASL      R2
027662 006302          IF #FLAG96 NOTSETIN STATUS(R2) THEN
2684 027664          BIT      #FLAG96,STATUS(R2)
027664 032762 010000 002516  BNE      50361$
027672 001063          OUTPUT #AUTSEC,#77...,LUNIT
2685 027674          OUTPUT #AUTCON,#61...,LUNIT
2686 027736          OUTPUT #OUTBUF,#33...,LUNIT
2687 030000          ENDIF
2688 030042          50361$: ENDINCR
030042          BR      50357$
2689 030042          50360$: LET OUTBUF :B= #14
030042 000677          MOVB   #14,OUTBUF
030044          OUTPUT #OUTBUF,#1
2690 030044          EXIT   TST                ;AND EXIT TEST
030044 112737 000014 003164  TRAP  C$EXIT
2691 030052          .WORD  L10016-.
2692 030114          ;CHARACTER BUFFER AND TEST HEADER MESSAGE
030114 104432          ;
030116 000730          .NLIST BEX
2693          NONCHR: .ASCII /NON-PRINTABLE CHARACTERS TEST 07/<12>
2694          .ASCIZ /A FULL LINE OF EACH CODE WILL BE SENT/<12>
2695          AUTSEC: .ASCIZ /THIS SECTION CHECKS AUTOMATIC CONVERSION ON PRINTERS WITH 64 CHARACTER BAND
2696          AUTCON: .ASCIZ /CODES 140(8) ..177(8) SHOULD BE CONVERTED TO 100(8)..137(8)/<12><12>
2697 030120          116      117      116      NONCHR: .ASCIZ <15><12><12><12>
2698 030161          101      040      106
2699 030230          124      110      111
S 2700 030346          103      117      104      AUTCON: .ASCIZ /CODES 140(8) ..177(8) SHOULD BE CONVERTED TO 100(8)..137(8)/<12><12>
2701 030444          015      012      012      SKIP3: .ASCIZ <15><12><12><12>
2702          NONBUF: .ASCII / 000 NUL/<0>
2703 030451          040      060      060      .ASCII / 001 SOH/<1>
2704 030462          040      060      060      .ASCII / 002 STX/<2>
2705 030473          040      060      060      .ASCII / 003 ETX/<3>
2706 030504          040      060      060      .ASCII / 004 EOT/<4>
2707 030515          040      060      060      .ASCII / 005 ENQ/<5>
2708 030526          040      060      060      .ASCII / 006 ACK/<6>
2709 030537          040      060      060      .ASCII / 007 BEL/<7>
2710 030550          040      060      060      .ASCII / 010 BS /<10>
2711 030561          040      060      061      .ASCII / 011 HT /<11>
2712 030572          040      060      061      .ASCII / 016 SO /<16>
2713 030603          040      060      061      .ASCII / 017 SI /<17>
2714 030614          040      060      061      .ASCII / 020 DLE/<20>
2715 030625          040      060      062      .ASCII / 021 XON/<21>
2716 030636          040      060      062      .ASCII / 022 DC2/<22>
2717 030647          040      060      062      .ASCII / 023 XOF/<23>
2718 030660          040      060      062      .ASCII / 024 DC4/<24>
2719 030671          040      060      062      .ASCII / 025 NAK/<25>
2720 030702          040      060      062      .ASCII / 026 SYN/<26>
2721 030713          040      060      062      .ASCII / 027 ETB/<27>
2722 030724          040      060      062      .ASCII / 030 CAN/<30>
2723 030735          040      060      063      .ASCII / 031 EM /<31>
2724 030746          040      060      063      .ASCII / 032 SUB/<32>
2725 030757          040      060      063

```

NON-PRINTABLE CHARACTERS

2726	030770	040	060	063	.ASCII / 033 ESC/<33>
2727	031001	040	060	063	.ASCII / 034 FS /<34>
2728	031012	040	060	063	.ASCII / 035 GS /<35>
2729	031023	040	060	063	.ASCII / 036 RS /<36>
2730	031034	040	060	063	.ASCII / 037 US /<37>
2731					.EVEN
2732					
2733					.LIST BEX
2734	031046				ENDTST
	031046				L10016:
	031046	104401			TRAP C#ETST
2735					
2736	031050				ENDMOD
2737					

BAND PATTERN

```

2739          .SBTTL BAND PATTERN
2740 031050   BGNMOD
2741          : **
2742          ;BAND PATTERN TEST
2743          :
2744          ;THIS TEST ONLY EXECUTES ON LP25 / LP26 / LP27. IT DOES NOTHING ON LP07.
2745          ;THIS TEST PRODUCES AN IMAGE OF THE ENTIRE BAND PATTERN. THE PRINT-OUT
2746          ;IS ORGANIZED TO LOCATE THE FOUR QUADRANTS OF THE BAND IN THE FOLLOWING
2747          ;FORMAT:
2748          :
2749          :           QUADRANT NO.1           QUADRANT NO. 2
2750          :           QUADRANT NO.3           QUADRANT NO.4
2751          :
2752          :           QUADRANT NO.1           ETC.
2753          :
2754          ;THE REASON FOR THIS ARRANGEMENT IS TO FACILITATE VISUAL INSPECTION
2755          ;OF THE PRINTOUT AS WELL AS TO ACCOMODATE THE 208 CHARACTERS OF THE BAND
2756          ;IN 132 COLUMNS.
2757          :
2758          :           STATUS(R2) FLAG96           =1 FOR 96 CHAR BAND
2759          :                                           =0 FOR 64 CHAR BAND
2760          :           SOFTWARE SWITCH USA           =1 FOR AMERICAN PRINT SET
2761          :                                           =0 FOR BRITISH PRINT SET
2762          :
2763          :--
2764 031050   BGNTST 8.
           031050   T8::
2765          000014   TOF = 014
2766          :
2767          ; PRINT TEST IDENTIFICATION ON ALL UNITS
2768          :
2769 031050   PRINTF #N007
           031050   012746 032600   MOV #N007,-(SP)
           031054   012746 000001   MOV #1,-(SP)
           031060   010600   MOV SP,R0
           031062   104417   TRAP C$PNTF
           031064   062706 000004   ADD #4,SP
2770          :
2771          ; SETUP PATTERNS FOR EUROPEAN OR AMERICAN PRINTERS
2772          :
2773 031070   IF USA NE #0 THEN ; AMERICAN, PRINT SHARP SIGN '#'
           031070   005737 002274   TST USA
           031074   001432   BEQ 50362$
2774 031076   LET BP64Q2+18. :B= #0
           031076   105037 033123   CLRB BP64Q2+18.
2775 031102   LET BP64Q2+22. :B= #43
           031102   112737 000043 033127   MOVB #43,BP64Q2+22.
2776 031110   LET BP64Q3+35. :B= #0
           031110   105037 033241   CLRB BP64Q3+35.
2777 031114   LET BP64Q3+40. :B= 43
           031114   113737 000043 033246   MOVB 43,BP64Q3+40.
2778 031122   LET BP64Q4+53. :B= #0
           031122   105037 033357   CLRB BP64Q4+53.
2779 031126   LET BP64Q4+57. :B= #43
           031126   112737 000043 033363   MOVB #43,BP64Q4+57.
2780 031134   LET BP96Q2+13. :B= #0
           031134   105037 033501   CLRB BP96Q2+13.

```

BAND PATTERN

2781	031140				LET BP96Q2+18. :B= #43
	031140	112737	000043	033506	MOVB #43, BP96Q2+18.
2782	031146				LET BP96Q4+13. :B= #0
	031146	105037	033672		CLRB BP96Q4+13.
2783	031152				LET BP96Q4+18. :B= #43
	031152	112737	000043	033677	MOVB #43, BP96Q4+18.
2784	031160				ELSE ; EUROPEAN, PRINT POUND STERLING SIGN
	031160	000431			BR 50363\$
	031162				50362\$:
2785	031162	112737	000043	033123	LET BP64Q2+18. :B= #43
	031162				MOVB #43, BP64Q2+18.
2786	031170				LET BP64Q2+22. :B= #0
	031170	105037	033127		CLRB BP64Q2+22.
2787	031174				LET BP64Q3+35. :B= #43
	031174	112737	000043	033241	MOVB #43, BP64Q3+35.
2788	031202				LET BP64Q3+40. :B= #0
	031202	105037	033246		CLRB BP64Q3+40.
2789	031206				LET BP64Q4+53. :B= #43
	031206	112737	000043	033357	MOVB #43, BP64Q4+53.
2790	031214				LET BP64Q4+57. :B= #0
	031214	105037	033363		CLRB BP64Q4+57.
2791	031220				LET BP96Q2+13. :B= #43
	031220	112737	000043	033501	MOVB #43, BP96Q2+13.
2792	031226				LET BP96Q2+18. :B= #0
	031226	105037	033506		CLRB BP96Q2+18.
2793	031232				LET BP96Q4+13. :B= #43
	031232	112737	000043	033672	MOVB #43, BP96Q4+13.
2794	031240				LET BP96Q4+18. :B= #0
	031240	105037	033677		CLRB BP96Q4+18.
2795	031244				ENDIF
	031244				50363\$:
2796					;
2797					; PRINT PROPER BAND IDENTIFICATION MSG. ON EACH PRINTER
2798					;
2802	031244				LET R1 := L\$UNIT - #1
	031244	013701	002012		MOV L\$UNIT, R1
	031250	005301			DEC R1
2803	031252				INCR LUNIT FROM #0 TO R1 BY #1
	031252	005037	002320		CLR LUNIT
	031256	000402			BR 50365\$
	031260				50364\$:
	031260	005237	002320		INC LUNIT
	031264				50365\$:
	031264	023701	002320		CMP LUNIT, R1
	031270	003402			BLE 50366\$
	031272	000137	031764		JMP 50367\$
	031276				50366\$:
2804	031276				LET R2 := LUNIT SHIFT 1
	031276	013702	002320		MOV LUNIT, R2
	031302	006302			ASL R2
2805	031304	032762	002000	002516	BIT #FLAG07, STATUS(R2)
2806	031312	001406			BEQ 1000\$
2807	031314	032762	001000	002516	BIT #FLAG26, STATUS(R2)
2808	031322	001002			BNE 1000\$
2809	031324	000137	031760		JMP 1010\$
2810	031330				1000\$:
2811	031330				OUTPUT #BNDTST, #23..., LUNIT

## BAND PATTERN

```

2812 031372                                IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG07 NOTSETIN STATUS(R2) THEN
      031372 032762 001000 002516          BIT #FLAG26,STATUS(R2)
      031400 001004                                BNE 50370$
      031402 032762 002000 002516          BIT #FLAG07,STATUS(R2)
      031410 001402                                BEQ .+6
      031412                                50370$:
      031412 000137 031460                    JMP 50371$
2813 031416                                OUTPUTI #BPID25,#6,,LUNIT ; PRINTER IS LP25
2814 031460                                ENDIF
      031460                                50371$:
2815 031460                                IF #FLAG26 SETIN STATUS(R2) AND #FLAG07 NOTSETIN STATUS(R2) THEN
      031460 032762 001000 002516          BIT #FLAG26,STATUS(R2)
      031466 001404                                BEQ 50372$
      031470 032762 002000 002516          BIT #FLAG07,STATUS(R2)
      031476 001402                                BEQ .+6
      031500                                50372$:
      031500 000137 031546                    JMP 50373$
2816 031504                                OUTPUTI #BPID26,#6,,LUNIT ; PRINTER IS LP26
2817 031546                                ENDIF
      031546                                50373$:
2818 031546                                IF #FLAG07 SETIN STATUS(R2) AND #FLAG26 SETIN STATUS(R2) THEN
      031546 032762 002000 002516          BIT #FLAG07,STATUS(R2)
      031554 001404                                BEQ 50374$
      031556 032762 001000 002516          BIT #FLAG26,STATUS(R2)
      031564 001002                                BNE .+6
      031566                                50374$:
      031566 000137 031634                    JMP 50375$
2819 031572                                OUTPUTI #BPID27,#6,,LUNIT ; PRINTER IS LP27
2820 031634                                ENDIF
      031634                                50375$:
2821 031634                                IF #FLAG96 SETIN STATUS(R2) THEN
      031634 032762 010000 002516          BIT #FLAG96,STATUS(R2)
      031642 001002                                BNE .+6
      031644 000137 031716                    JMP 50376$
2822 031650                                OUTPUTI #BP96ID,#22,,LUNIT ; 96 CHAR BAND
2823 031712                                ELSE
      031712 000137 031760                    JMP 50377$
      031716                                50376$:
2824 031716                                OUTPUTI #BP64ID,#22,,LUNIT ; 64 CHAR BAND
2825 031760                                ENDIF
      031760                                50377$:
2826 031760                                1010$:
2827 031760                                ENDINC
      031760 000137 031260                    JMP 50364$
      031764                                50367$:
2831 ;
2832 ; NOW PRINT 2_LINE PATTERN 15. TIMES, WITH BLANK LINE BETWEEN PATTERNS
2833 ;
2834 031764                                LET LINCNT := #14.
      031764 012737 000016 002302          MOV #14.,LINCNT
2835 031772                                2$:
2836 031772                                INCR LUNIT FROM #0 TO R1 BY #1 ; PRINT QUADRANTS 1 & 2
      031772 005037 002320                    CLR LUNIT
      031776 000402                                BR 50400$
      032000                                50401$:
      032000 005237 002320                    INC LUNIT
      032004                                50400$:

```

## BAND PATTERN

```

032004 023701 002320      CMP      LUNIT,R1
032010 003114      BGT      50402$
2837 032012      LET R2 := LUNIT SHIFT 1      ; INDEX INTO STATUS TABLES
032012 013702 002320      MOV      LUNIT,R2
032016 006302      ASL      R2
2838 032020      IF #FLAG07 SETIN STATUS(R2) AND #FLAG26 NOTSETIN STATUS(R2) THEN
032020 032762 002000 002516  BIT      #FLAG07,STATUS(R2)
032026 001407      BEQ      50403$
032030 032762 001000 002516  BIT      #FLAG26,STATUS(R2)
032036 001003      BNE      50403$
2839 032040      LET COUNT := #0
032040 005037 002306      CLR      COUNT
2840 032044      ELSE
032044 000475      BR      50404$
032046      50403$:
2841 032046      IF #FLAG96 NOTSETIN STATUS(R2) THEN
032046 032762 010000 002516  BIT      #FLAG96,STATUS(R2)
032054 001022      BNE      50405$
2842 032056      OUTPUTI #BP64Q1,#121...,LUNIT      ; 64 CHAR PATTERN
2843 032120      ELSE
032120 000447      BR      50406$
032122      50405$:
2844 032122      IF #FLAG26!FLAG07 NOTSETIN STATUS(R2) THEN
032122 032762 003000 002516  BIT      #FLAG26!FLAG07,STATUS(R2)
032130 001022      BNE      50407$
2845 032132      OUTPUTI #BP96Q3,#121...,LUNIT      ; LP25 96 CHAR PATTERN
2846 032174      ELSE
032174 000421      BR      50410$
032176      50407$:
2847 032176      OUTPUTI #BP96Q1,#121...,LUNIT      ; LP26 96 CHAR PATTERN
2848 032240      ENDIF
032240      50410$:
2849 032240      ENDIF
032240      50406$:
2850 032240      ENDIF
032240      50404$:
2851 032240      ENDINC
032240 000657      BR      50401$
032242      50402$:
2852      ; NOW DO QUADRANTS 3 & 4
2853 032242      INCR LUNIT FROM #0 TO R1 BY #1      ; REPEAT FOR ALL UNITS
032242 005037 002320      CLR      LUNIT
032246 000402      BR      50411$
032250      50412$:
032250 005237 002320      INC      LUNIT
032254      50411$:
032254 023701 002320      CMP      LUNIT,R1
032260 003065      BGT      50413$
2854 032262      LET R2 := LUNIT SHIFT 1      ; INDEX INTO STATUS TABLES
032262 013702 002320      MOV      LUNIT,R2
032266 006302      ASL      R2
2855 032270 032762 002000 002516  BIT      #FLAG07,STATUS(R2)
2856 032276 001406      BEQ      1100$
2857 032300 032762 001000 002516  BIT      #FLAG26,STATUS(R2)
2858 032306 001002      BNE      1100$
2859 032310 000137 032432      JMP      1110$
2860 032314      1100$:

```



## BAND PATTERN

```

2861 032314          IF #FLAG96 NOTSETIN STATUS(R2) THEN
      032314 032762 010000 002516      BIT #FLAG96,STATUS(R2)
      032322 001022          BNE 50414$
2862 032324          OUTPUTI #BP64Q3,#122...,LUNIT ; 64 CHAR PATTERN
2863 032366          ELSE
      032366 000421          BR 50415$
      032370          50414$:
2864 032370          OUTPUTI #BP96Q3,#122...,LUNIT ; 96 CHAR PATTERN
2865 032432          ENDIF
      032432          50415$:
2866 032432          1110$:
2867 032432          ENDINC
      032432 000706          BR 50412$
      032434          50413$:
2868 032434          LET LINCNT := LINCNT - #1
      032434 005337 002302          DEC LINCNT
2869 032440          BEQ 3$
2870 032442          000137 031772          JMP 2$
2871 032446          3$:
2872          ;
2873          ; DO TOF THEN EXIT ON ALL UNITS EXCEPT LP07'S
2874 032446          LET #OUTBUF :B= #14
      032446 112727 000014 003164          MOVB #14,#OUTBUF
2875 032454          INCR LUNIT FROM #0 TO R1 BY #1
      032454 005037 002320          CLR LUNIT
      032460 000402          BR 50416$
      032462          50417$:
      032462 005237 002320          INC LUNIT
      032466          50416$:
      032466 023701 002320          CMP LUNIT,R1
      032472 003040          BGT 50420$
2876 032474          LET R2 := LUNIT SHIFT 1
      032474 013702 002320          MOV LUNIT,R2
      032500 006302          ASL R2
2877 032502          IF #FLAG07 SETIN STATUS(R2) AND #FLAG26 NOTSETIN STATUS(R2) THEN
      032502 032762 002000 002516      BIT #FLAG07,STATUS(R2)
      032510 001407          BEQ 50421$
      032512 032762 001000 002516      BIT #FLAG26,STATUS(R2)
      032520 001003          BNE 50421$
2878 032522          LET COUNT := #0
      032522 005037 002306          CLR COUNT
2879 032526          ELSE
      032526 000421          BR 50422$
      032530          50421$:
2880 032530          OUTPUTI #OUTBUF,#1,,LUNIT
2881 032572          ENDIF
      032572          50422$:
2882 032572          ENDINC
      032572 000733          BR 50417$
      032574          50420$:
2883          ;
2884 032574          EXIT TST
      032574 104432          TRAP C$EXIT
      032576 001156          .WORD L10017-.
2885          .NLIST BEX
2886 032600          045 116 045 NO07: .ASCII /*N*ABAND PATTERN TEST DOES NOTHING ON LP07'S.*N/
2887 032660          102 101 116 BNDTST: .ASCII /*BAND PATTERN TEST 08 /

```

BAND PATTERN

```

2888 032707    114    120    062  BPID25: .ASCII /LP25 /
2889 032715    114    120    062  BPID26: .ASCII /LP26 /
2890 032723    114    120    062  BPID27: .ASCII /LP27 /
2891 032731    066    064    040  BP64ID: .ASCII /64 CHAR BAND PATTERN/<12><12>
2892 032757    071    066    040  BP96ID: .ASCII /96 CHAR BAND PATTERN/<12><12>
2893
2894           ; 64 CHAR BAND PATTERN LP25 & LP26 & LP27
2895           ;
2896 033005    040    040    040  BP64Q1: .ASCII / /
2897 033015    101    102    105    .ASCII /ABECDTFGOHIJ1KL2MN3OP4QRS5/
2898 033047    125    126    066    .ASCII /UV6WX7YZ8_+*9&)*(:,/<57>/$-;><./
2899
2900 033101    040    040    040  BP64Q2: .ASCII / /
2901 033111    077    100    105    .ASCII /?@E][T\"0+#+1!#2' A3BC4DFG5/
2902 033143    110    111    066    .ASCII /HI6JK7LM8NOP9QR*SU,VW-XYZ./<12>
2903
2904 033176    040    040    040  BP64Q3: .ASCII / /
2905 033206    137    053    105    .ASCII /_+E#&T)(0:/<57>/$1;>2<?3@]4[\"5/
2906 033240    136    043    066    .ASCII /+*6=!7#' 8ABC9DF*GH,IJ-KLM./
2907
2908 033272    040    040    040  BP64Q4: .ASCII / /
2909 033302    116    117    105    .ASCII /NOEPQTRSOUVW1XY2Z_3+*4&)(5/
2910 033334    072    057    066    .ASCII _:/6$;7><8?@]9[\"*'+,#=-!#' _<12><12>
2911
2912           ; 96 CHAR BAND      LP25 = Q3..Q4      LP26 & LP27 = Q1..Q4
2913           ;
2914 033370    040    040    040  BP96Q1: .ASCII / /
2915 033400    101    102    103    .ASCII /ABCD0EFG1HIJ2KLMN3OPQ4RST5/
2916 033432    125    126    127    .ASCII /UVWX6YZ_7$*8&)(:/<57>/9+;>/
2917 033457    145                    .BYTE 145
2918 033460    074    077    100    .ASCII /<?@/
2919 033463    164                    .BYTE 164
2920
2921 033464    040    040    040  BP96Q2: .ASCII / /
2922 033474    135    133    134    .ASCII /][\" ,#/
2923 033502    174                    .BYTE 174
2924 033503    075    052    041    .ASCII /=*!#' -/
2925 033511    175    173    176    .BYTE 175,173,176,136,56,177,140,141,60,142,143,144,61
2926 033526    146    147    150    .BYTE 146,147,150,151,62,152,153,154,63,155,156
2927 033541    157    064    160    .BYTE 157,64,160,161,162,163,65,165,166,167,66
2928 033554    170    171    172    .BYTE 170,171,172,67,12
2929
2930 033561    040    040    040  BP96Q3: .ASCII / /
2931 033571    101    102    103    .ASCII /ABCD8EFG9HIJ/
2932 033605    145                    .BYTE 145
2933 033606    113    114    115    .ASCII /KLMN/
2934 033612    164                    .BYTE 164
2935 033613    117    120    121    .ASCII /OPQ,RST*/
2936 033623    125    126    127    .ASCII /UVWX-YZ_.$*80)(:/<57>/1+;>2<?@3/
2937
2938 033655    040    040    040  BP96Q4: .ASCII / /
2939 033665    135    133    134    .ASCII /][\"4#/
2940 033673    174                    .BYTE 174
2941 033674    075    065    041    .ASCII /=5!#' 6/
2942 033702    175    173    176    .BYTE 175,173,176,136,67,177,140,141,70,142,143,144,71
2943 033717    146    147    150    .BYTE 146,147,150,151,145,152,153,154,164,155,156,157
2944 033733    054    160    161    .BYTE 54,160,161,162,163,52,165,166,167,55,170

```

BAND PATTERN

2945	033746	171	172	056	.BYTE	171,172,56,12,12
2946					.LIST BEX	
2947					.EVEN	
2948	033754				ENDTST	
	033754				L10017:	
	033754	104401			TRAP	C\$ETST
2949	033756				ENDMOD	

SPURIOUS HAMMER FIRING

```

2951 .SBTTL SPURIOUS HAMMER FIRING
2952
2953 033756 BGNMOD
2954
2955 ;++
2956 ;THE PURPOSE OF THIS TEST IS TO DETECT SPURIOUS HAMMER FIRINGS AND DEFECTIVE
2957 ;HAMMER DRIVERS DURING THE OPERATION OF THE LINE PRINTER. THE PROGRAM
2958 ;PRODUCES A LEFT WEDGE PATTERN CONSISTING OF 132 LINES OF PRINT WITH EACH
2959 ;LINE BEGINNING WITH A "?" CHARACTER. ANY POINT OUTSIDE THE WEDGE
2960 ;BOUNDARIES IS CAUSED BY HAMMER MISFIRES OR BY HAMMER BOUNCE.
2961 ;--
2962
2963 033756 BGNTST 9.
      033756 T9::

2964
2965 ;PRINT THE TEST HEADER
2966
2967 033756 OUTPUT #HAMFIR,#33.
2968
2969 ;OUTPUT THE ACTUAL WEDGE AT THIS POINT
2970
2971 034020 INCR WORK FROM #1 TO #132. BY #1 ;NUMBER OF LINES TO OUTPUT
      034020 012737 000001 003160 MOV #1,WORK
      034026 000402 BR 50423$
      034030 50424$:
      034030 005237 003160 INC WORK
      034034 50423$:
      034034 023727 003160 000204 CMP WORK,#132.
      034042 003123 BGT 50425$

2972 ;ALSO NUMBER OF PRINTING CHARACTERS
2973
2974 034044 LET R4 := #OUTBUF ;OUTPUT BUFFER POINTER
      034044 012704 003164 MOV #OUTBUF,R4
2975 034050 LET SPCCNT := #132. - WORK ;NUMBER OF SPACES TO FILL IN
      034050 012737 000204 034366 MOV #132.,SPCCNT
      034056 163737 003160 034366 SUB WORK,SPCCNT

2976 ;FILL THE OUTPUT BUFFER WITH THE REQUIRED NUMBER OF SPACES
2977
2978
2979 034064 WHILE SPCCNT NE #0 DO
      034064 50426$:
      034064 005737 034366 TST SPCCNT
      034070 001405 BEQ 50427$
2980 034072 LET (R4)+ :B= #40 ;SPACE FILL
      034072 112724 000040 MOVB #40,(R4)+
2981 034076 LET SPCCNT := SPCCNT - #1 ;UPDATE FILLER COUNTER
      034076 005337 034366 DEC SPCCNT
2982 034102 ENDDO
      034102 000770 BR 50426$
      034104 50427$:
2983 034104 LET CCNT := #0
      034104 005037 002310 CLR CCNT
2984 034110 LET CHRGEN := #77 ;FIRST CHARACTER A "?"
      034110 012737 000077 002314 MOV #77,CHRGEN
2985 034116 LET STRCNT := #33. ;# OF CHARACTERS IN GROUP
      034116 012737 000041 002312 MOV #33.,STRCNT
2986 034124 WHILE CCNT LT WORK DO ;NOW FILL IN REST OF BUFFER

```

## SPURIOUS HAMMER FIRING

```

034124
034124 023737 002310 003160 50430$:
034132 002022          CMP      CCNT,WORK
2987 034134          BGE      50431$
034134 005737 002312          IF STRCNT EQ #0 THEN
034140 001006          TST      STRCNT
          BNE      50432$
2988
2989          ;RESET GROUP POINTERS AND COUNTERS
2990
2991 034142          LET STRCNT := #33.
034142 012737 000041 002312          MOV      #33.,STRCNT
2992 034150          LET CHRGEN := #77
034150 012737 000077 002314          MOV      #77,CHRGEN
2993 034156          ENDF
034156          50432$:
2994 034156          LET (R4)+ :B= CHRGEN
034156 113724 002314          MOV      CHRGEN,(R4)+
2995 034162          LET CHRGEN := CHRGEN + #1
034162 005237 002314          INC      CHRGEN
2996 034166          LET CCNT := CCNT + #1
034166 005237 002310          INC      CCNT
2997 034172          LET STRCNT := STRCNT - #1
034172 005337 002312          DEC      STRCNT          ;UPDATE POINTERS AND COUNTERS
2998 034176          ENDDO
034176 000752          BR      50430$
034200          50431$:
2999
3000          ;NOW SET UP LINE TERMINATOR AND OUTPUT THE LINE.
3001
3002 034200          LET (R4)+ :B= #12
034200 112724 000012          MOV      #12,(R4)+
3003          ;OUTPUT THE LINE
3004
3005          OUTPUT #OUTBUF,#132.          ; SEND THE DATA, NO LF YET
3006 034246          OUTPUT #OUTBUF+132.,#1          ; THIS MAKES SURE OUTPUT IS SENT
3007          ; BEFORE CHANGING OUTBUF DATA !
3008
3009 034310          ENDINC
034310 000647          BR      50424$
034312          50425$:
3010 034312          LET OUTBUF :B= #14
034312 112737 000014 003164          MOV      #14,OUTBUF
3011 034320          OUTPUT #OUTBUF,#1
3012 034362          EXIT TST
034362 104432          TRAP   C$EXIT
034364 000046          .WORD  L10020-.
3013
3014          ;COUNTERS, POINTERS, TEXT BUFFER, AND HEADER FOR TEST PRINTOUT
3015
3016 034366 000000          SPCCNT: .WORD 0
3017
3018          ;TEST HEADER MESSAGE
3019          .NLIST BEX
3020 034370 123 120 125 HAMFIR: .ASCIZ /SPURIOUS HAMMER FIRING TEST 09/<12><12><12>
3021
3022          ;
3023          .LIST BEX

```

SPURIOUS HAMMER FIRING

3024

3025

3026

3027 034432

034432

034432 104401

3028

3029 034434

.EVEN

ENDTST

L10020:

TRAP

C\$ETST

ENDMOD

PRINT CONTROL

```

3031
3032
3033 034434
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048 034434
      034434
3049 034434
      034434 012746 035451
      034440 012746 000001
      034444 010600
      034446 104417
      034450 062706 000004
3050 034454
      034454 013701 002012
      034460 005301
3051      000001
3052 034462
      034462 005037 002320
      034466 000402
      034470
      034470 005237 002320
      034474
      034474 023701 002320
      034500 003402
      034502 000137 035354
      034506
3053 034506
      034506 013702 002320
      034512 006302
3054 034514 032762 002000 002516
3055 034522 001406
3056 034524 032762 001000 002516
3057 034532 001002
3058 034534 000137 035350
3059 034540
3060 034540
3061 034602
      034602 012737 000015 002306
3062 034610
3063 034610
      034610 012705 035710
3064 034614
      034614
      034614 005715
    
```

```

.SBTTL PRINT CONTROL
BGNMOD
; **
; THIS TEST CHECKS THE PRINT CONTROL BY SENDING MORE THAN 132 CHARACTERS
; BEFORE SENDING A PRINT COMMAND. ALL CHARACTERS IN EXCESS OF 132 CHARACTERS
; SHOULD BE DISREGARDED. THIS TEST DOES NOTHING ON ALL LP07'S.
; THREE LINES ARE PRINTED PER ITERATION, THESE LINES WILL IDENTIFY THE
; COLUMN NUMBERS ACROSS THE PAGE. EXAMPLE :
;      0      0      0.....          1
;      1      2      3.....          3
; 123456789012345678901234567890..... 012
;
; NOTICE THAT THE PRINTOUT SHOULD IDENTIFY 132 COLUMNS ACROSS THE PAGE.
;
; THIS OUTPUT IS REPEATED 13 TIMES.
; --
BGNTST 10.
T10:
PRINTF #NOLP07
      MOV      #NOLP07, -(SP)
      MOV      #1, -(SP)
      MOV      SP, R0
      TRAP     C$PNTF
      ADD      #4, SP
LET R1 := L$UNIT - #1
      MOV      L$UNIT, R1
      DEC      R1
$BRJMP=1
INCR LUNIT FROM #0 TO R1 BY #1
      CLR      LUNIT
      BR       50434$
50433$:
      INC      LUNIT
50434$:
      CMP      LUNIT, R1
      BLE      50435$
      JMP      50436$
50435$:
LET R2 := LUNIT SHIFT 1
      MOV      LUNIT, R2
      ASL      R2
      BIT      #FLAG07, STATUS(R2)
      BEQ      1200$
      BIT      #FLAG26, STATUS(R2)
      BNE      1200$
      JMP      1210$
1200$:
      OUTPUTI #PRTCTL, #56..., LUNIT
      LET COUNT := #13.
      MOV      #13., COUNT
1$:
LET R5 := #TABLE1
      MOV      #TABLE1, R5
      WHILE (R5) NE #0 DO
50437$:
      TST      (R5)
    
```

## PRINT CONTROL

```

034616 001002          BNE      .+6
034620 000137 034666  JMP      50440$
3065 034624          OUTPUTI (R5)+,#10...LUNIT
3066 034664          ENDDO
034664 000753          BR      50437$
034666          50440$:
3067 034666          LET OUTBUF :B= #12
034666 112737 000012 003164  MOVB   #12,OUTBUF
3068 034674          OUTPUTI #OUTBUF,#1,,LUNIT
3069
3070 034736          LET R5 := #TABLE2
034736 012705 035744  MOV     #TABLE2,R5
3071 034742          WHILE (R5) NE #0 DO
034742          50441$:
034742 005715          TST     (R5)
034744 001002          BNE     .+6
034746 000137 035014  JMP     50442$
3072 034752          OUTPUTI (R5)+,#10...LUNIT
3073 035012          ENDDO
035012 000753          BR      50441$
035014          50442$:
3074 035014          OUTPUTI #OUTBUF,#1,,LUNIT
3075
3076 035056          DECR LINCNT FROM #14. TO #1 BY #1
035056 012737 000016 002302  MOV     #14,,LINCNT
035064 000402          BR      50444$
035066          50443$:
035066 005337 002302  DEC     LINCNT
035072          50444$:
035072 023727 002302 000001  CMP     LINCNT,#1
035100 002002          BGE     50445$
035102 000137 035152  JMP     50446$
035106          50445$:
3077 035106          OUTPUTI #X11,#10...LUNIT
3078 035150          ENDDO
035150 000746          BR      50443$
035152          50446$:
3079 035152          OUTPUTI #OUTBUF,#1,,LUNIT
3080 035214          OUTPUTI #OUTBUF,#1,,LUNIT
3081 035256          LET COUNT := COUNT - #1
035256 005337 002306  DEC     COUNT
3082 035262          IF COUNT GT #0 THEN
035262 005737 002306  TST     COUNT
035266 003002          BGT     .+6
035270 000137 035300  JMP     50447$
3083 035274          JMP 1$
3084 035300          ENDDO
035300          50447$:
3085 035300          LET OUTBUF :B= #14
035300 112737 000014 003164  MOVB   #14,OUTBUF
3086 035306          OUTPUTI #OUTBUF,#1,,LUNIT
3087 035350          1210$:
3088 035350          ENDINCR
035350 000137 034470  JMP     50433$
035354          50436$:
3089 177777          $BRJMP--1
3090 035354          EXIT TST

```



PRINT CONTROL

```

035354 104432          TRAP      C$EXIT
035356 000422          .WORD    L10021-.
3091          .NLIST BEX
3092 035360          120          122          111 PRTCTL: .ASCII /PRINT CONTROL TEST 10/ <12>
3093 035406          123          110          117          .ASCIZ  /SHOULD SHOW 132 COLUMNS PRINTED/<12><12><15>
3094 035451          045          116          045 NOLP07: .ASCIZ  /N$APRINT CONTROL TEST DOES NOTHING ON LP07'S.$N/
3095
3096 035532          040          040          040 X0:      .ASCII  /          0/
3097 035544          040          040          040 X1:      .ASCII  /          1/
3098 035556          040          040          040 X2:      .ASCII  /          2/
3099 035570          040          040          040 X3:      .ASCII  /          3/
3100 035602          040          040          040 X4:      .ASCII  /          4/
3101 035614          040          040          040 X5:      .ASCII  /          5/
3102 035626          040          040          040 X6:      .ASCII  /          6/
3103 035640          040          040          040 X7:      .ASCII  /          7/
3104 035652          040          040          040 X8:      .ASCII  /          8/
3105 035664          040          040          040 X9:      .ASCII  /          9/
3106
3107 035676          061          062          063 X11:     .ASCII  /1234567890/
3108
3109          .EVEN
3110 035710 035532 035532 035532 TABLE1: .WORD  X0,X0,X0,X0,X0,X0,X0,X0,X0,X1,X1,X1,X1,0
3111 035744 035544 035556 035570 TABLE2: .WORD  X1,X2,X3,X4,X5,X6,X7,X8,X9,X0,X1,X2,X3,0
3112          .EVEN
3113
3114          .LIST BEX
3115 036000          ENDTST
036000          L10021:
036000 104401          TRAP      C$ETST
3116 036002          ENDMOD
    
```

## MULTIPLE LINE ADVANCE

```

3119          .SBTTL  MULTIPLE LINE ADVANCE
3120
3121 036002    BGNMOD
3122          ;**
3123          ;THIS TEST CHECKS THE MULTIPLE LINE ADVANCE OF THE LINE PRINTER. A LINE OF
3124          ;NUMBERS IS PRINTED AND THEN THE PAPER IS ADVANCED THAT NUMBER OF LINES.  THUS THE
3125          ;NUMBER PRINTED WILL INDICATE THE NUMBER OF BLANK LINES FOLLOWING THAT
3126          ;LINE.  THE NUMBER OF LINES IS VARIED BETWEEN 2 AND 7 AND A LINE OF
3127          ;ALL 0'S WILL INDICATE THE END OF THE TEST SEQUENCE.
3128          ;--
3129
3130
3131 036002    BGNTST 11.
3132          T11::
3133
3134          ;PRINT TEST IDENTIFICATION
3135 036002    OUTPUT #MULINE,#86.
3136
3137 036044    LET STACHR := #TABSTR          ;OUTPUT CHARACTERS
3138          036044 012737 036312 036310    MOV      #TABSTR,STACHR
3139 036052    REPEAT
3140          036052 117737 000232 002302    50450$:
3141          036060          LET LINCNT := #STACHR          ;GET A CHARACTER TO OUTPUT
3142          036064          MOV      #STACHR,LINCNT
3143          036070          LET LINCNT := LINCNT AND #7 ;MAKE THE ASCII TO OCTAL
3144          036074          MOV      LINCNT,-(SP)
3145          036074          BIC     #7,(SP)
3146          036074          BIC     (SP)+,LINCNT
3147          036100          LET R3 := #OUTBUF          ;SET UP OUTPUT BUFFER
3148          036100 012737 000001 002310    MOV      #OUTBUF,R3
3149          036106          INCR CCNT FROM #1 TO #132. BY #1
3150          036110          MOV      #1,CCNT
3151          036110          BR      50451$
3152          036110          50452$:
3153          036114          I'IC    CCNT
3154          036122          50451$:
3155          036124          CMP     CCNT,#132.
3156          036124          BGT     50453$
3157          036124          LET (R3)+ := #STACHR          ;PUT CHARACTER IN OUTPUT BUFFER
3158          036124          MOV     #STACHR,(R3)+
3159          036130          ENDINC
3160          036130          BR      50452$
3161          036132          50453$:
3162          036132          LET R4 := #0
3163          036132          CLR     R4
3164          036134          WHILE R4 NE LINCNT DO
3165          036134          50454$:
3166          036134          020437 002302    CMP     R4,LINCNT
3167          036140          001404          BEQ     50455$
3168          036142          LET (R3)+ := #12
3169          036142          112723 000012    MOV     #12,(R3)+
3170          036146          LET R4 := R4 + #1
3171          036146          005204          INC     R4
3172          036150          ENDDO
3173          036150          000771          BR      50454$

```

## MULTIPLE LINE ADVANCE

```

036152          50455$:
3151          ;NOW OUTPUT THE ACTUAL LINE
3152
3153
3154 036152          LET R4 := LINCNT * #132.          ;NUMBER OF CHARACTERS TO OUTPUT
          036152 013704 002302          MOV      LINCNT,R4
          036156 062704 000204          ADD      #132.,R4
3155 036162          LET STACHR := STACHR * #1          ; UPDATE CHARACTER COUNT
          036162 005237 036310          INC      STACHR
3156 036166          OUTPUT #OUTBUF,R4          ;OUTPUT THE LINE
3157
3158 036226          UNTIL LINCNT EQ #0
          036226 005737 002302          TST      LINCNT
          036232 001307          BNE      50450$
3159 036234          LET OUTBUF :B= #14
          036234 112737 000014 003164          MOV      #14,OUTBUF
3160 036242          OUTPUT #OUTBUF,#1
3161
3162 036304          EXIT TST
          036304 104432          TRAP     C$EXIT
          036306 000150          .WORD   L10022-.
3163
3164
3165 036310 000000          STACHR: .WORD 0
3166          .NLIST BEX
3167 036312          062      067      062  TABSTR: .ASCIZ /272637463540/
3168 036327          115      125      114  MULINE: .ASCII /MULTIPLE LINE ADVANCE TEST 11/<12>
3169 036365          116      125      115  .ASCIZ  /NUMBERS PRINTED REPRESENT # LINES TO NEXT LINE PRINTED/<12><12>
3170
3171
3172
3173          .EVEN
3174          .LIST BEX
3175
3176 036456          ENDTST
          036456          L10022:
          036456 104401          TRAP     C$ETST
3177 036460          ENDMOD
3178

```

## CHARACTER ALIGNMENT

```

3180 .SBTTL CHARACTER ALIGNMENT
3181 036460 BGNMOD
3182 ;**
3183 ;THIS TEST CHECKS CHARACTER ALIGNMENT BY OVERPRINTING LINES OF ALTERNATING
3184 ;H'S AND SPACES WITH SPACES AND H'S.
3185 ;--
3186 036460 BGNTST 12.
      036460 T12::
3187 ;PRINT TEST IDENTIFICATION
3188 036460 OUTPUT #CHRALN,#30. ; PRINT TEST NAME ON LP
3189 ;PRINT 24 LINES OF ALTERNATING "H"'S AND "SPACE"'S
3190 036522 012737 000062 002302 1$: LET LINCNT := #50.
      036522 MOV #50.,LINCNT
3191 036530 005737 002302 2$: IF LINCNT LE #0 THEN
      036530 TST LINCNT
      036534 003002 BGT 50456$
3192 036536 000137 037044 JMP INLINE <JMP 3$>
      036536 JMP 3$
3193 036542 ENDIF
      036542 50456$:
3194 ;LOAD BUFFER WITH ALTERNATING STRING OF "H"'S AND "SPACE"'S
3195 036542 012704 003164 LET R4 := #OUTBUF
      036542 MOV #OUTBUF,R4
3196 036546 012737 000001 003160 INCR WORK FROM #1 TO #66. BY #1 ; 132 CHARACTERS
      036546 MOV #1,WORK
      036554 000402 BR 50457$
      036556 50460$:
      036556 005237 003160 INC WORK
      036562 023727 003160 000102 50457$:
      036562 CMP WORK,#66.
      036570 003005 BGT 50461$
3197 036572 112724 000110 LET (R4)+ :B= #110 ; PUT PATTERN INTO BUFFER
      036572 MOVB #110,(R4)+
3198 036576 112724 000040 LET (R4)+ :B= #40
      036576 MOVB #40,(R4)+
3199 036602 000765 ENDINCR
      036602 BR 50460$
3200 036604 112724 000015 50461$:
      036604 LET (R4)+ :B= #CR ; FOLLOWED BY CR
      MOVB #CR,(R4)+
3201 ;
3202 ; SEND BASIC PATTERN
3203 ;
3204 036610 OUTPUT #OUTBUF,#132.
3205 036652 OUTPUT #OUTBUF+132.,#1
3206 ;
3207 ; OVERPRINT WITH LINE OF ALTERNATING SPACE AND 'H'
3208 ;
3209 036714 012704 003370 LET R4 := #OUTBUF+132. ; FILL BUFFER WITH REVERSE PATTERN
      036714 MOV #OUTBUF+132.,R4
3210 036720 112724 000110 LET (R4)+ :B= #110 ; H
      036720 MOVB #110,(R4)+
3211 036724 112724 000012 LET (R4)+ :B= #LF ; FOLLOWED BY A LINEFEED
      036724 MOVB #LF,(R4)+
3212 ;
3213 036730 OUTPUT #OUTBUF+1,#132. ; OVERPRINT
3214 036772 OUTPUT #OUTBUF+133.,#1

```

## CHARACTER ALIGNMENT

```

3215 037034          LET LINCNT := LINCNT - #1
      037034 005337 002302      DEC LINCNT
3216 037040          INLINE <JMP 2$>
      037040 000137 036530      JMP 2$
3217 037044          3$:
3218 037044          LET OUTBUF :B= #14
      037044 112737 000014 003164      MOVB #14,OUTBUF
3219 037052          OUTPUT #OUTBUF,#1
3220 037114 004737 005606      JSR PC,QUIET
3221 037120          EXIT TST
      037120 104432          TRAP C$EXIT
      037122 000042          .WORD L10023-.
3222          .NLIST BEX
3223 037124          103          110          101 CHRALN: .ASCIZ /CHARACTER ALIGNMENT TEST 12/<12><12><12>
3224          .EVEN
3225          .LIST BEX
3226 037164          ENDTST
      037164          L10023:
      037164 104401          TRAP C$ETST
3227 037166          ENDMOD

```

## INTERRUPT SERVICE ROUTINES

```

3229          .SBTTL  INTERRUPT SERVICE ROUTINES
3230 037166   BGNSRV
3231          ;
3232          ;++
3233          ;INTERRUPT VECTORS ARE ESTABLISHED DURING INITIALIZATION
3234          ;POINTING TO THE BASIC ROUTINES WHICH
3235          ;SET UP THE UNIT NUMBER CAUSING THE INTERRUPTS.
3236          ;LINE NUMBER IS RETURNED IN R2
3237          ;
3238          ;--
3239          X=0
3240 037166   000000
3241          000020
3242          INT00: .REPT 16.
3243          SETPRI #PRI04
3244          PUSH  R2
3245          LET R2 := #X
3246          X=X+2
                 INLINE <JMP  IODRV>
                 .ENDR
037166 012700 000200   MOV  #PRI04,R0
037172 104441         TRAP  C$SPRI
037174 010246         MOV  R2,-(SP)
037176 012702 000000   MOV  #X,R2
037202 000137 004774   JMP  IODRV
037206 012700 000200   MOV  #PRI04,R0
037212 104441         TRAP  C$SPRI
037214 010246         MOV  R2,-(SP)
037216 012702 000002   MOV  #X,R2
037222 000137 004774   JMP  IODRV
037226 012700 000200   MOV  #PRI04,R0
037232 104441         TRAP  C$SPRI
037234 010246         MOV  R2,-(SP)
037236 012702 000004   MOV  #X,R2
037242 000137 004774   JMP  IODRV
037246 012700 000200   MOV  #PRI04,R0
037252 104441         TRAP  C$SPRI
037254 010246         MOV  R2,-(SP)
037256 012702 000006   MOV  #X,R2
037262 000137 004774   JMP  IODRV
037266 012700 000200   MOV  #PRI04,R0
037272 104441         TRAP  C$SPRI
037274 010246         MOV  R2,-(SP)
037276 012702 000010   MOV  #X,R2
037302 000137 004774   JMP  IODRV
037306 012700 000200   MOV  #PRI04,R0
037312 104441         TRAP  C$SPRI
037314 010246         MOV  R2,-(SP)
037316 012702 000012   MOV  #X,R2
037322 000137 004774   JMP  IODRV
037326 012700 000200   MOV  #PRI04,R0
037332 104441         TRAP  C$SPRI
037334 010246         MOV  R2,-(SP)
037336 012702 000014   MOV  #X,R2
037342 000137 004774   JMP  IODRV
037346 012700 000200   MOV  #PRI04,R0
037352 104441         TRAP  C$SPRI
037354 010246         MOV  R2,-(SP)
037356 012702 000016   MOV  #X,R2

```

## INTERRUPT SERVICE ROUTINES

037362	000137	004774	JMP	IODRV
037366	012700	000200	MOV	#PRI04,R0
037372	104441		TRAP	C\$SPRI
037374	010246		MOV	R2,-(SP)
037376	012702	000020	MOV	#X,R2
037402	000137	004774	JMP	IODRV
037406	012700	000200	MOV	#PRI04,R0
037412	104441		TRAP	C\$SPRI
037414	010246		MOV	R2,-(SP)
037416	012702	000022	MOV	#X,R2
037422	000137	004774	JMP	IODRV
037426	012700	000200	MOV	#PRI04,R0
037432	104441		TRAP	C\$SPRI
037434	010246		MOV	R2,-(SP)
037436	012702	000024	MOV	#X,R2
037442	000137	004774	JMP	IODRV
037446	012700	000200	MOV	#PRI04,R0
037452	104441		TRAP	C\$SPRI
037454	010246		MOV	R2,-(SP)
037456	012702	000026	MOV	#X,R2
037462	000137	004774	JMP	IODRV
037466	012700	000200	MOV	#PRI04,R0
037472	104441		TRAP	C\$SPRI
037474	010246		MOV	R2,-(SP)
037476	012702	000030	MOV	#X,R2
037502	000137	004774	JMP	IODRV
037506	012700	000200	MOV	#PRI04,R0
037512	104441		TRAP	C\$SPRI
037514	010246		MOV	R2,-(SP)
037516	012702	000032	MOV	#X,R2
037522	000137	004774	JMP	IODRV
037526	012700	000200	MOV	#PRI04,R0
037532	104441		TRAP	C\$SPRI
037534	010246		MOV	R2,-(SP)
037536	012702	000034	MOV	#X,R2
037542	000137	004774	JMP	IODRV
037546	012700	000200	MOV	#PRI04,R0
037552	104441		TRAP	C\$SPRI
037554	010246		MOV	R2,-(SP)
037556	012702	000036	MOV	#X,R2
037562	000137	004774	JMP	IODRV

3247

## CLOCK SERVICE ROUTINE

```

3249          .SBTTL CLOCK SERVICE ROUTINE
3250          ;**
3251          ;UPDATES THE COUNTER AT A RATE OF 16.67 MILLISECONDS PER TICK
3252          ;AND UPDATES A SECOND COUNTER WHEN THE FIRST OVERFLOWS.
3253          ;--
3254
3255 037566     BGNSRV
3256 037566     CLKTCK: SETPRI #PRI06
                   MOV      #PRI06,R0
                   TRAP    C$SPRI
3257 037574     IF TICK EQ #0 THEN
                   TST     TICK
                   BNE     50462$
3258 037602     LET TICK := #60.                ;60 TICKS PER SECOND
                   MOV     #60.,TICK
3259 037610     LET TIME := TIME + #1
                   INC     TIME
3260 037614     ENDF
                   50462$:
3261 037614     LET TICK := TICK - #1          ;BACK UP SECOND TIMER
                   DEC     TICK
3262 037620     IF CLKTYP EQ #2 THEN
                   CMP     CLKTYP,#2
                   BNE     50463$
3263 037630     LET @CLKCSR := #100
                   MOV     #100,@CLKCSR
3264 037636     ENDF
                   50463$:
3265
3266 037636     ENDSRV          ;AND EXIT
                   L10025:
                   RTI
3267
3268 037640     ;
3269 037642     TIME:  .WORD  0
                   TICK: .WORD  0

```



HARDWARE PARAMETER SECTION

```

3271          .SBTTL  HARDWARE PARAMETER SECTION
3272 037644   BGNMOD
3273
3274          ;++
3275          ;THIS SECTION INCLUDES THE QUESTIONS WHICH REQUEST THE OPERATOR TO
3276          ;FURNISH THE HARDWARE INFORMATION NECESSARY TO BUILD THE HARDWARE
3277          ;P-TABLES.
3278          ;
3279          ;--
3280 037644   BGNHRD
3281          .WORD  L10026-L$HARD/2
3282          L$HARD::
3282 037646   GPRMA  GETADR,0,0,160000,177516,YES
3283          .WORD  T$CODE
3284          037650   .WORD  GETADR
3285          037652   .WORD  T$LLOLIM
3286          037654   .WORD  T$HILIM
3287 037656   GPRMA  GETVEC,2,0,110,770,YES
3288          .WORD  T$CODE
3289          037660   .WORD  GETVEC
3290          037662   .WORD  T$LLOLIM
3291          037664   .WORD  T$HILIM
3292 037666   GPRMD  GETTYP,4,0,3,0,3,YES
3293          .WORD  T$CODE
3294          037670   .WORD  GETTYP
3295          037672   .WORD  3
3296          037674   .WORD  T$LLOLIM
3297          037676   .WORD  T$HILIM
3298 037700   GPRML  GETBND,6,1,YES
3299          .WORD  T$CODE
3300          037702   .WORD  GETBND
3301          037704   .WORD  1
3302 037706   ENDHRD
3303          .EVEN
3304          L10026:
3305          .NLIST BEX
3306 037706   061  GETADR: .ASCIZ  .LP11/LPV11 ADDRESS.
3307 037731   111  124  GETVEC: .ASCIZ  /INTERRUPT VECTOR/
3308 037752   105  124  GETTYP: .ASCIZ  .ENTER 0 IF LP25, 1 IF LP26, 2 IF LP07, 3 IF LP27 .
3309 040034   071  066  040  GETBND: .ASCIZ  /96 CHARACTER BAND/
3310          ; ENTER 3 IF LPYY   FOR FUTURE EXPANSION
3311          .LIST BEX
3312          .EVEN
3294

```

## SOFTWARE PARAMETER SECTION

```

3296      .SBTTL  SOFTWARE PARAMETER SECTION
3297      ;
3298      ;++
3299      ;THIS SECTION INCLUDES THE QUESTIONS WHICH REQUEST THE OPERATOR TO FURNISH
3300      ;THE SOFTWARE INFORMATION NECESSARY TO BUILD THE SOFTWARE P-TABLES.
3301      ;--
3302      ;
3303 040056 BGNSFT
      040056 000023      .WORD L10027-L$SOFT/2
      040060      L$SOFT::
3304 040060      GPRML  MGTINT,0,1,YES
      040060 000130      .WORD  T$CODE
      040062 040126      .WORD  MGTINT
      040064 000001      .WORD  1
3305 040066      GPRML  GETMAN,2,1,YES
      040066 001130      .WORD  T$CODE
      040070 040164      .WORD  GETMAN
      040072 000001      .WORD  1
3306 040074      GPRMD  GETTIM,4,D,377,4,60.,YES
      040074 002052      .WORD  T$CODE
      040076 040236      .WORD  GETTIM
      040100 000377      .WORD  377
      040102 000004      .WORD  T$LOLIM
      040104 000074      .WORD  T$HILIM
3307 040106      GPRML  GETPLA,6,1,YES
      040106 003130      .WORD  T$CODE
      040110 040323      .WORD  GETPLA
      040112 000001      .WORD  1
3308 040114      GPRMD  GETMAX,10,D,377,1,255.,YES
      040114 004052      .WORD  T$CODE
      040116 040344      .WORD  GETMAX
      040120 000377      .WORD  377
      040122 000001      .WORD  T$LOLIM
      040124 000377      .WORD  T$HILIM
3309 040126      ENDSFT
      040126      .EVEN
3310      L10027:
3311 040126      122 125 116 MGTINT: .ASCIZ  /RUN MANUAL INTERVENTION TESTS/
3312 040164      120 105 122 GETMAN: .ASCIZ  /PERFORM MANUAL PRINTING SPEED MEASUREMENT/
3313 040236      104 105 123 GETTIM: .ASCIZ  /DESIRED TIME INTERVAL FOR PRINTING SPEED CALCULATION/
3314 040323      124 105 123 GETPLA: .ASCIZ  /TESTING IN U.S.A/
3315 040344      101 125 124 GETMAX: .ASCIZ  /AUTODROP ERROR COUNT/
3316      .LIST BEX
3317      .EVEN
3318      ;
3319
3320 040372      PATCH: .BLKW  20
3321 040432      LASTAD
      040432 000000      .EVEN
      040434 000000      .WORD  0
      040436      .WORD  0
3322 040436      L$LAST::
3323      ENDMOD
      000001      .END

```



## Symbol table

L\$AUTO	002260	G	L10011	014610	PAPSW1	013536	TB0764	025370	T7	027330	G
L\$CCP	002106	G	L10012	017716	PAPTST	013674	TB0796	025576	T8	031050	G
L\$CLEA	010154	G	L10013	026004	PATCH	040372	TICK	037642	T9	033756	G
L\$CO	002032	G	L10014	026430	PATTER	026426	TIME	037640	UAM	= 000200	G
L\$DEPO	002011	G	L10015	027326	PERIOD	002272	TIMOUT=	000002	UNIT	002316	
L\$DESC	002162	G	L10016	031046	PLOC	010064	TOF	= 000014	USA	002274	
L\$DESP	002076	G	L10017	033754	PNT	= 001000	TXERR	003665	UUT	002346	
L\$DEVP	002060	G	L10020	034432	PRI	= 002000	TXNOIN	003737	UUTEQO	003770	
L\$DISP	002132	G	L10021	036000	PRINTR	002324	T\$ARGC=	000001	VFUCMD	002352	
L\$DLY	002116	G	L10022	036456	PRI00	= 000000	T\$CODE=	004052	VFUINF	004263	
L\$DTP	002040	G	L10023	037164	PRI01	= 000040	T\$ERRN=	000012	VFUINT	014425	
L\$DTYP	002034	G	L10025	037636	PRI02	= 000100	T\$EXCP=	000000	VFURDY	014517	
L\$DUT	002072	G	L10026	037706	PRI03	= 000140	T\$FLAG=	000040	VFUSEL	004037	
L\$DVTY	002222	G	L10027	040126	PRI04	= 000200	T\$GMAN=	000000	VFUSE1	004125	
L\$EF	002052	G	L26M64	024233	PRI05	= 000240	T\$HILI=	000377	WORK	003160	G
L\$ENVI	002044	G	L26M96	024314	PRI06	= 000300	T\$LAST=	000001	WORK1	003162	
L\$ETP	002102	G	L27M64	024537	PRI07	= 000340	T\$LOLI=	000001	X	= 000040	
L\$EXP1	002046	G	L27M96	024620	PRTCHR	027266	T\$LSYM=	010000	X\$ALWA=	000000	
L\$EXP4	002064	G	MANSPD	002270	PRTCTL	035360	T\$LTNO=	000014	X\$FALS=	000040	
L\$EXP5	002066	G	MAXERR	002276	PRTSPD	024024	T\$NEST=	000000	X\$OFFS=	000400	
L\$HARD	037646	G	MGTINT	040126	PTABAD	002322	T\$NSO	= 000010	X\$TRUE=	000020	
L\$HIME	002120	G	MOVMSG	017616	QUIET	005606	T\$NS1	= 000000	X0	035532	
L\$HPCP	002016	G	MRESET	007365	RDYERR	003430	T\$NS2	= 000005	X1	035544	
L\$HPTP	002022	G	MSGADR	002716	READY	007673	T\$PTNU=	000000	X11	035676	
L\$HW	002250	G	MSGCNT	002616	REFLIN	017104	T\$SAVL=	177777	X2	035556	
L\$ICP	002104	G	MULINE	036327	REPCNT	002656	T\$SEGL=	177777	X3	035570	
L\$INIT	006016	G	NMLFLS	017203	REPLUP	022330	T\$SUBN=	000000	X4	035602	
L\$LADP	002026	G	NOCLCK	007730	RESET1	007464	T\$TAGL=	177777	X5	035614	
L\$LAST	040436	G	NOCLK	004153	RESET2	007557	T\$TAGN=	010030	X6	035626	
L\$LOAD	002100	G	NOINTR=	000003	RESTOR	024701	T\$TEMP=	000000	X7	035640	
L\$LUN	002074	G	NOLP07	035451	RESVEC	010072	T\$TEST=	000014	X8	035652	
L\$MREV	002050	G	NONBUF	030451	SFPTBL	002266	T\$TSTM=	177777	X9	035664	
L\$NAME	002000	G	NONCHR	030120	SKIP3	030444	T\$TSTS=	000001	\$AAFLA=	000403	
L\$PRIO	002042	G	NOTIM	007772	SPCCNT	034366	T\$AUT=	010002	\$AFLAG=	000403	
L\$PROT	002122	G	NO07	032600	SPED1	023430	T\$CLE=	010005	\$BGNLE=	177777	
L\$PRT	002112	G	NRGT16	007246	SPED2	023453	T\$HAR=	010026	\$BRJMP=	177777	
L\$REPP	002062	G	NRGT17	007331	SPED3	023457	T\$HW	= 010001	\$ERFLG=	000400	
L\$REV	002010	G	OFFLIN	023534	STACHR	036310	T\$INI=	010004	\$FLAG	= 000330	
L\$SOFT	040060	G	ONEFIL=	000001	STATER=	000001	T\$PRO=	010000	\$FAND=	000310	
L\$SPC	002056	G	ONLIN1	023574	STATUS	002516	T\$SOF=	010027	\$FBAD=	000401	
L\$SPCP	002020	G	ONLIN2	023675	STRCNT	002312	T\$SRV=	010025	\$FBLA=	000170	
L\$SPTP	002024	G	ONLIN3	023773	SVCGBL=	000000	T\$SW	= 010003	\$FCAS=	000150	
L\$STA	002030	G	OUTBUF	003164	SVCINS=	000000	T\$TES=	010023	\$FDEC=	000220	
L\$SW	002266	G	OUTTIM	003712	SVCSUB=	000000	T1	010350	\$FDO	= 000340	
L\$TEST	002114	G	O\$APTS=	000000	SVCTAG=	000000	T1A	010364	\$FEXI=	000101	
L\$TIML	002014	G	O\$AU	= 000000	SVCTST=	000000	T1B	011142	\$FFAL=	000405	
L\$UNIT	002012	G	O\$BGNR=	000000	SYMD	= 000007	T1C	010370	\$FGOO=	000400	
L07M64	024375		O\$BGNS=	000001	SYML	= 000005	T10	034434	\$FIF	= 000110	
L07M96	024456		O\$DU	= 000000	SYMS	= 000007	T11	036002	\$FINC=	000210	
L10001	002260		O\$ERRT=	000000	SYX	= 050215	T12	036460	\$FLOO=	000200	
L10002	002262		O\$GNSW=	000001	S\$LSYM=	010000	T2	011640	\$FNAM=	000160	
L10003	002300		O\$POIN=	000001	TABA64	024754	T3	014612	\$FNO	= 000403	
L10004	010066		O\$SETU=	000000	TABA96	025162	T3MOV	016444	\$FOR	= 000320	
L10005	010346		PAPCHK	017042	TABLDA	004754	T3SET	016442	\$FRTI=	000350	
L10006	011636		PAPRDY	013605	TABLE1	035710	T4	017720	\$FRTN=	000300	
L10007	011332		PAPRSW	013466	TABLE2	035744	T5	026006	\$FRTT=	000360	
L10010	011340		PAPSWI	003452	TABSTR	036312	T6	026432	\$FSEL=	000140	

## Symbol table

\$F\$THE= 000330	\$LSI11= 177777	\$SAVLE= 177777	\$TS4 = 050432	\$\$INH = 000403
\$F\$TRU= 000404	\$LSKO = 000000	\$SELLE= 000001	\$TS5 = 050273	\$\$LOC = 037626
\$F\$UNT= 000130	\$LSK1 = 000000	\$SS0 = 050460	\$TS6 = 050274	\$\$LOCN= 000000
\$F\$WHI= 000120	\$LSTIN= 000000	\$SS1 = 000402	\$TS7 = 050264	\$\$REG = 177777
\$F\$YES= 000402	\$LSTTA= 000000	\$SS2 = 050216	\$U = 000403	\$\$RETU= 000000
\$IFLEV= 177777	\$NESTL= 177777	\$SYMD = 000070	\$\$ARGC= 000000	\$\$RTN1= 000000
\$INH = 000403	\$NSKO = 000110	\$SYMS = 000007	\$\$BYTE= 000000	\$\$RTN2= 000000
\$ISK0 = 000001	\$NSK1 = 000120	\$TAGLE= 177777	\$\$CASE= 000404	\$\$SRC = 000027
\$ISK1 = 000001	\$NSK2 = 000110	\$TAGNU= 050464	\$\$DST = 000037	\$\$TGSV= 050217
\$ISK2 = 000001	\$NSK3 = 000110	\$TEMP = 050463	\$\$ELOC= 000402	\$\$TGS1= 050216
\$ISK3 = 000001	\$NSK4 = 000110	\$TS0 = 050463	\$\$ERFL= 000000	\$\$TGS2= 000000
\$ISK4 = 000001	\$NSK5 = 000110	\$TS1 = 050460	\$\$FLAG= 000001	\$\$TO = 000000
\$LO = 177777	\$SAVE = 050215	\$TS2 = 050455	\$\$FROM= 000000	\$\$TAG= 050000
\$LOCTA= 177777	\$SAVE2= 050216	\$TS3 = 050446		

. ABS. 040436 000 (RW,I,GBL,ABS,OVR)  
 000000 001 (RW,I,LCL,REL,CON)

Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 12716  
 Work file writes: 6950  
 Size of work file: 58520 Words ( 229 Pages)  
 Size of core pool: 19714 Words ( 75 Pages)  
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:38:50.19  
 CZLPLFO,CZLPLFO/NL:TOC/-SP=SVC34/ML,SPMACJ/ML,CZLPLFO.P11