LP25,LP26 LP25 DIAGNOSTIC AH-E635C-MC FEB 1981
CZLPLC0 FICHE 1 OF 1

B 1

.REM &

IDENTIFICATION

PRODUCT CODE : AC-E634C-MC

PRODUCT NAME: CZLPLCO LP25, LP26 DIAG

MAINTAINER: SMALL SYSTEMS DIAGNOSTICS

PRODUCT DATE: 23-SEP-80

AUTHOR: JOHN CHATALIAN
DON RICE
RALPH SCHAUBER

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC PROGRAM VERIFIES PROPER PROPER OPERATION OF THE LP25 OR LP26 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	DAVFU ERROR DETECTION
TEST 6	DAVFU LINE COUNT PAPER CONTROL
TEST 7	DAVFU CHANNEL SELECTION PAPER CONTROL
TEST 8	DATA TRANSFER PATHS
TEST 9	PRINTABLE CHARACTERS
TEST 10	NON-PRINTABLE CHARACTERS
TEST 11	BAND PATTERN
TEST 12	SPURIOUS HAMMER FIRING
TEST 13	PRINT CONTROL
TEST 14	CRITICAL PATHS
TEST 15	MULTIPLE LINE ADVANCE
TEST 16	CHARACTER ALIGNMENT

ANY MIX OF PRINTER TYPES (LP25 OR LP26) 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 24K 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 CZLPLA0 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)

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.

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 "DDDDD".

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:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000).
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.

/EOP:DDDDD REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)

/UNITS:LIST 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: (O)  (177514) ?
INTERRUPT VECTOR : (O)  (200) ?
ENTER 0 IF LP25 OR 1 IF LP26 (O)  (O) ?
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
```

```
DAVFU OPTION INSTALLED (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
                                                           MAXIMUM IS 60 SEC.
                                                           MINIMUM IS 4 SEC.
TESTING IN U.S.A. (Y) ?          SELECTS U.S. OR BRITISH BAND PATTERN
AUTODROP ERROR COUNT (D) 5 ?     DROPS ANY UNIT FROM TEST WHICH EXCEEDS SPECIFIED NO. OF ERRORS
```

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING A FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

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

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

UNIT 2
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  1<CR>
Q-FACTOR (0) 1 ?  0<CR>

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

UNIT 4
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  3<CR>
Q-FACTOR (0) 0 ?  <CR>

UNIT 5
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  4<CR>
Q-FACTOR (0) 0 ?  <CR>

UNIT 6
CSR ADDRESS (0) ?  160000<CR>
```

```
                    SUB-DEVICE # (O) ?  5<CR>
                    Q-FACTOR (O) 0 ?  <CR>

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

                    UNIT 8
                    CSR ADDRESS (O)  160000<CR>
                    SUB-DEVICE # (O) ?  7<CR>
                    Q-FACTOR (O) 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 (D) ?  8<CR>

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

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

                    UNIT 7
                    CSR ADDRESS (O) ?  160000<CR>
                    SUB-DEVICE # (O) ?  6,7<CR>
                    Q-FACTOR (O) 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 (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0-7<CR>
Q-FACTOR (O) 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.
11	"DAVFU INCOMPLETE DATA ERROR NOT DETECTED" DAVFU FAILED TO RECOGNIZE RECEIPT OF INCOMPLETE DATA.
12	"DAVFU STOP CODE ERROR NOT DETECTED" DAVFU FAILED TO RECOGNIZE RECEIPT OF DATA THAT DID NOT INCLUDE A STOP BIT

(ONE) CHARACTER.

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.

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
DAVFU ERROR DETECTION
CHECKS FOR TWO TYPES OF DAVFU ERRORS:
1. RECEIPT OF INCOMPLETE DATA.
2. RECEIPT OF DATA NOT INCLUDING A STOP BIT (ONE) CHARACTER.

TEST 6

DAVFU LINE COUNT PAPER CONTROL
VERIFIES LINE COUNT METHOD OF PAPER CONTROL USING THE DAVFU.

TEST 7
DAVFU CHANNEL SELECTION PAPER CONTROL
CHECKS DAVFU PAPER ADVANCE BY MEANS OF STOP BITS LOADED IN
DAVFU MEMORY.

TEST 8
DATA TRANSFER PATHS
CHECKS THE DATA TRANSFER PATHS FROM THE PRINTER OUTPUT TO
THE PROCESSOR INTERFACE.

TEST 9
PRINTABLE CHARACTERS
CHECKS FOR PROPER PRINTING OF ALL PRINTABLE CHARACTERS.

TEST 10
NON-PRINTABLE CHARACTERS
CHECKS FOR PROPER DETECTION OF ALL NON-PRINTABLE CHARACTERS.

TEST 11
BAND PATTERN
PRODUCES AN IMAGE OF THE ENTIRE BAND PATTERN.

TEST 12
SPURIOUS HAMMER FIRING
CHECKS FOR SPURIOUS HAMMER FIRINGS BY TAKING NOTE OF ANY
PRINTING THAT MAY OCCUR OUTSIDE A WEDGE PATTERN.

TEST 13
PRINT CONTROL
CHECKS THAT CHARACTERS IN EXCESS OF 132 CHARACTERS ON A LINE
ARE DISREGARDED.

TEST 14
CRITICAL PATH
CHECKS FOR PROPER PRINTER OPERATION WITH A WORST CASE PATTERN.

TEST 15
CHECKS MULTIPLE LINE ADVANCE
CHECKS THE MULTIPLE LINE ADVANCE FOR PROPER PAPER MOVEMENT.

TEST 16
CHARACTER ALIGNMENT
CHECKS CHARACTER ALIGNMENT BY OVERPRINTING EACH LINE.

```
                        .TITLE  CZLPLCO LP25, LP26 DIAGNOSTIC
                        .ENABL AMA
                        .SBTTL  IDENTIFICATION

                        ; PRODUCT CODE:         AC-E634A-MC

                        ; PRODUCT NAME:         CZLPLCO LP25, LP26 DIAG

                        ; MAINTAINER:           SMALL SYSTEMS DIAGNOSTICS

                        ; AUTHORS:              JOHN CHATALIAN
                        ;                       DONALD RICE
                        ;                       RALPH SCHAUBER

                        ; DATE                  13-JUN-80

                        ;COPYRIGHT (C) 1979, 1980
                        ;DIGITAL EQUIPMENT CORPORATION, MAYNARD MASSACHUSSETTS 01754
                        ;
                        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON  A
                        ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
                        ; SION OF THE ABOVE COPYRIGHT NOTICE.  THIS  SOFTWARE,  OR  ANY
                        ; OTHER  COPIES  THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
                        ; AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON  SUCH  SYSTEM
                        ; AND  TO  ONE WHO AGREES TO THESE LICENSE TERMS.  TITLE TO AND
                        ; OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
                        ;
                        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
                        ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
                        ; EQUIPMENT CORPORATION.
                        ;
                        ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF
                        ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

```
                        ;++
                        ; FUNCTIONAL DESCRIPTION
                        ;
                        ; THIS DIAGNOSTIC PROGRAM VERIFIES PROPER OPERATION OF THE LP25 OR LP26
                        ; LINE PRINTER, AND IT'S ASSOCIATED INTERFACE MODULE.
                        ;
                        ; ANY MIX OF LP25 AND LP26 PRINTERS MAY BE TESTED, UP TO A TOTAL OF
                        ; SIXTEEN UNITS.
                        ;
                        ; THE PROGRAM CONSISTS OF SIXTEEN TESTS, THREE OF WHICH ARE FOR THE DAVFU OPTION.
                        ; THREE OF THE PRINTER TESTS INVOLVE MANUAL INTERVENTION. TWO OF THE DAVFU TESTS
                        ; REQUIRE INTERVENTION BY THE OPERATOR.
                        ;
                        ; THE PROGRAM IS COMPATIBLE TO THE PDP-11 DIAGNOSTIC SUPERVISOR, ACT/SLIDE, AND
                        ; XXDP+.
                        ;
                        ;--



                        ; VERSION       A-0    27-SEP-79        R. SCHAUBER

                        ; HISTORY       REV. A-0   INITIAL RELEASE
                        ;               REV-C SUPERVISOR / XXDP+ COMPATABLE

                        ;               REV. B-0   DOCUMENTATION CHANGE  29-NOV-79
                        ;                          CHANGE INIT CODE TO SET PRIO ON NEW PASS
                        ;
                        ;               REV. C-0        INCLUDE LP26 SUPPORT    13-JUN-80
                        ;                               INCLUDE TEST MESSAGES TO THE PRINTERS
                        ;                               INCLUDES LINE CLOCK SUPPORT FOR LSI-11
```

```
                                        .TITLE  CZLPLCO LP25, LP26 TEST
                                .SBTTL PROGRAM HEADER

                                        .MCALL  SVC
000000'                                 SVC                             ;INITIALIZE SUPERVISOR MACROS
                                        .MCALL  STRUCT
000000'                                 STRUCT                          ;STRUCTURED MACRO PACKAGE
000000                                  $LSTIN=         0               ; LIST ASSY CODE LEFT
000000                                  $LSTTAG=        0               ; LIST TAGS LEFT
177777                                  $LOCTAG=        -1

000000                          SVCINS= 0       ;LIST INSTRUCTIONS
000000                          SVCTST= 0       ;LIST TEST TAGS
000000                          SVCSUB= 0       ;LIST SUBTEST TAGS
000000                          SVCGBL= 0       ;LIST GLOBAL TAGS
000000                          SVCTAG= 0       ;LIST OTHER TAGS

                                .ENABL AMA
                                .ENABL ABS
                                .ENABL LC
002000                                  .=2000

002000                                  BGNMOD
002000                                  POINTER BGNSW,BGNSFT

002000                                  HEADER  CZLPL,C,0,60,1,340
(4) 002000                      L$NAME::                        ;DIAGNOSTIC NAME
(4) 002000  103                         .ASCII /C/
(4) 002001  132                         .ASCII /Z/
(4) 002002  114                         .ASCII /L/
(4) 002003  120                         .ASCII /P/
(4) 002004  114                         .ASCII /L/
(6) 002005  000                         .BYTE   0
(6) 002006  000                         .BYTE   0
(5) 002007  000                         .BYTE   0
(5) 002010                      L$REV::                         ;REVISION LEVEL
(4) 002010  103                         .ASCII  /C/
(5) 002011                      L$DEPO::                        ;0
(4) 002011  060                         .ASCII  /0/
(5) 002012                      L$UNIT::                        ;NUMBER OF UNITS
(4) 002012  000000                      .WORD   0
(5) 002014                      L$TIML::                        ;LONGEST TEST TIME
(4) 002014  000060                      .WORD   60
(5) 002016                      L$HPCP::                        ;POINTER TO H.W. QUES.
(4) 002016  041146                      .WORD   L$HARD
(5) 002020                      L$SPCP::                        ;POINTER TO S.W. QUES.
(4) 002020  041324                      .WORD   L$SOFT
(5) 002022                      L$HPTP::                        ;PTR. TO DEF. H.W. PTABLE
(4) 002022  002254                      .WORD   L$HW
(5) 002024                      L$SPTP::                        ;PTR. TO S.W. PTABLE
(4) 002024  002272                      .WORD   L$SW
(5) 002026                      L$LADP::                        ;DIAG. END ADDRESS
(4) 002026  041736                      .WORD   L$LAST
(5) 002030                      L$STA::                         ;RESERVED FOR APT STATS
(4) 002030  000000                      .WORD   0
(5) 002032                      L$CO::
```

```
(4)  002032  000000                        .WORD   0
(5)  002034                         L$DTYP::        ;DIAGNOSTIC TYPE
(4)  002034  000001                        .WORD   1
(5)  002036                         L$APT::         ;APT EXPANSION
(4)  002036  000000                        .WORD   0
(5)  002040                         L$DTP::         ;PTR. TO DISPATCH TABLE
(4)  002040  002132                        .WORD   L$DISPATCH
(5)  002042                         L$PRIO::        ;DIAGNOSTIC RUN PRIORITY
(4)  002042  000340                        .WORD   340
(5)  002044                         L$ENVI::        ;FLAGS DESCRIBE HOW IT WAS SETUP
(4)  002044  000000                        .WORD   0
(5)  002046                         L$EXP1::        ;EXPANSION WORD
(4)  002046  000000                        .WORD   0
(5)  002050                         L$MREV::        ;SVC REV AND EDIT #
(4)  002050     003                         .BYTE   C$REVISION
(3)  002051     003                         .BYTE   C$EDIT
(5)  002052                         L$EF::          ;DIAG. EVENT FLAGS
(4)  002052  000000                        .WORD   0
(5)  002054  000000                        .WORD   0
(5)  002056                         L$SPC::
(4)  002056  000000                        .WORD   0
(5)  002060                         L$DEVP::        ; POINTER TO DEVICE TYPE LIST
(4)  002060  002240                        .WORD   L$DVTYP
(5)  002062                         L$REPP::        ;PTR. TO REPORT CODE
(4)  002062  000000                        .WORD   0
(5)  002064                         L$EXP4::
(4)  002064  000000                        .WORD   0
(5)  002066                         L$EXP5::
(4)  002066  000000                        .WORD   0
(5)  002070                         L$AUT::         ;PTR. TO ADD UNIT CODE
(4)  002070  000000                        .WORD   0
(5)  002072                         L$DUT::         ;PTR. TO DROP UNIT CODE
(4)  002072  000000                        .WORD   0
(5)  002074                         L$LUN::         ;LUN FOR EXERCISERS TO FILL
(4)  002074  000000                        .WORD   0
(5)  002076                         L$DESP::        ;POINTER TO DIAG. DESCRIPTION
(4)  002076  002172                        .WORD   L$DESC
(5)  002100                         L$LOAD::        ;GENERATE SPECIAL AUTOLOAD EMT
(4)  002100  104035                        EMT     E$LOAD
(5)  002102                         L$ETP::         ;POINTER TO ERRTBL
(4)  002102  000000                        .WORD   0
(5)  002104                         L$ICP::         ;PTR. TO INIT CODE
(4)  002104  005676                        .WORD   L$INIT
(5)  002106                         L$CCP::         ;PTR. TO CLEAN-UP CODE
(4)  002106  007352                        .WORD   L$CLEAN
(5)  002110                         L$ACP::         ;PTR. TO AUTO CODE
(4)  002110  002264                        .WORD   L$AUTO
(5)  002112                         L$PRT::         ;PTR. TO PROTECT TABLE
(4)  002112  002122                        .WORD   L$PROT
(5)  002114                         L$TEST::        ;TEST NUMBER
(4)  002114  000000                        .WORD   0
(5)  002116                         L$DLY::         ;DELAY COUNT
(4)  002116  000000                        .WORD   0
(5)  002120                         L$HIME::        ;PTR. TO HIGH MEM
(4)  002120  000000                        .WORD   0
800
```

```
  801                                   ;
  802                                   ; THE FOLLOWING IS A LOAD PROTECTION TABLE
  803                                   ;
  804  002122                                   BGNPROT
  (3)  002122                           L$PROT::
  805  002122  000000                           .WORD   0
  806  002124  177777                           .WORD   -1
  807  002126  177777                           .WORD   -1
  808  002130                                   ENDPROT
```

```
                                .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                                  DISPATCH        16              ;X= NUMBER OF TESTS
(4) 002130  000020                      .WORD   16
(3) 002132                      L$DISPATCH::
(6) 002132  007546                      .WORD   T1
(6) 002134  010726                      .WORD   T2
(6) 002136  013704                      .WORD   T3
(6) 002140  016162                      .WORD   T4
(6) 002142  022330                      .WORD   T5
(6) 002144  023716                      .WORD   T6
(6) 002146  025422                      .WORD   T7
(6) 002150  026672                      .WORD   T8
(6) 002152  027304                      .WORD   T9
(6) 002154  030132                      .WORD   T10
(6) 002156  031464                      .WORD   T11
(6) 002160  033776                      .WORD   T12
(6) 002162  034446                      .WORD   T13
(6) 002164  035604                      .WORD   T14
(6) 002166  037306                      .WORD   T15
(6) 002170  037764                      .WORD   T16

                                ;
                                ;FOR USE ON REVISION C OF THE SUPERVISOR
                                ;
002172                                  DESCRIP         <CZLPLCO LP25, LP26 PRINTER DIAGNOSTIC>
(4) 002172                      L$DESC::
(3) 002172  055103  050114  041514      .ASCIZ  /CZLPLCO LP25, LP26 PRINTER DIAGNOSTIC/
(3) 002200  020060  050114  032462
(3) 002206  020054  050114  033062
(3) 002214  050040  044522  052116
(3) 002222  051105  042040  040511
(3) 002230  047107  051517  044524
(3) 002236  000103
(2)                                     .EVEN
002240                                  DEVTYP          <LP25,LP26>
(4) 002240                      L$DVTYP::
(3) 002240  050114  032462  046054      .ASCIZ  /LP25,LP26/
(3) 002246  031120  000066
(2)                                     .EVEN
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE RUN-TIME P-TABLE.
                                ;--

002252                                  BGNHW   DFPTBL
(3) 002252  000004                      .WORD   L10001-L$HW/2
(3) 002254                      L$HW::
(3) 002254                      DFPTBL::
002254  177514                         .WORD   177514                  ;LP25 REGISTER ADDRESS
002256  000200                         .WORD   200                     ;LP25 INTERRUPT VECTOR
002260  000000                         .WORD   0                       ; 0 IF LP25
                                                                        ; 1 IF LP26
002262  000000                         .WORD   0                       ; 0 IF 64 CHAR BAND
                                                                        ; 1 IF 96 CHAR BAND
                                ;
                                ;INTERRUPT VECTOR PRIORITY IS 4 AND CANNOT BE CHANGED


002264                                  ENDHW
(3) 002264                      L10001:


002264                                  BGNAUTO
(3) 002264                      L$AUTO::

002264  000240                         NOP                    ; NOT USED

002266                                  ENDAUTO
(3) 002266                      L10002:
(3) 002266  104461                      TRAP    C$AUTO
```

```
                        .SBTTL  SOFTWARE P-TABLE

                        ;++
                        ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                        ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                        ;--

002270                          BGNSW   SFPTBL
(3) 002270  000006              .WORD   L10003-L$SW/2
(3) 002272              L$SW::
(3) 002272              SFPTBL::

002272  000000          INHINT: .WORD   0               ;0 IF NO INTERVENTION TESTS
                                                        ;1 IF MANUAL INTERVENTION TESTS
                                                        ;DEFAULT IS NO
002274  000000          VFUOPT: .WORD   0               ;0 IF NO DAVFU OPTION
                                                        ;1 IF DAVFU OPTION INSTALLED
                                                        ;NO DAVFU DEFAULT
002276  000000          MANSPD: .WORD   0               ;0 FOR AUTOMATIC PRINT SPEED
                                                        ;1 FOR MANUAL PRINT SPEED TEST
                                                        ;AUTOMATIC DEFAULT VALUE
002300  000074          PERIOD: .WORD   60.             ;OPERATOR TO SELECT TIMING VALUE
                                                        ;FROM 4 TO 60 SECONDS. INITIAL
                                                        ;DEFAULT VALUE IS 60 SECONDS.

002302  000001          USA:    .WORD   1               ; 1 FOR TESTING IN U.S.A.
                                                        ; 0 FOR TESTING IN G.B./EUROPE
                                                        ; * DIFFERENT BAND PATTERNS *
002304  000005          MAXERR: .WORD   5               ; AUTODROP ERROR COUNT
                        ; IF ERROR COUNT EXCEEDS MAXERR THE UNIT WILL BE DROPPED FROM TEST

002306                          ENDSW
(3) 002306              L10003:
```

```
                .SBTTL  I/O MACRO DEFINITIONS

                .MACRO  OUTPUT  ADD,BFCNT,ERR,PRINTS
                        MOV     ADD,BUFADD              ;SAVE THE BUFFER ADDRESS
                        MOV     BFCNT,BUFCNT            ;BUFFER BYTE COUNT BFCNT
                        MOV     #-1,PRINTR              ; OUTPUT TO ALL UNITS
                .IF B   ERR
                        MOV     #LPERR,ERRSVC
                .ENDC
                .IF NB  ERR
                        MOV     ERR,ERRSVC
                .ENDC
                .IF B PRINTS
                        MOV     #1,BUFREP               ; PRINT ONCE DEFAULT
                .ENDC
                .IF NB PRINTS
                        MOV     PRINTS,BUFREP           ; SUPPLY PRINT COUNT
                .ENDC
                        JSR     PC,IOCTRL               ;CALL THE DRIVER
                .ENDM   OUTPUT


                .MACRO  OUTPUI  ADD,BFCNT,ERR,UNIT,PRINTS
                        MOV     ADD,BUFADD              ;SAVE BUFFER ADDRESS
                        MOV     BFCNT,BUFCNT            ;BUFFER BYTE COUNT BFCNT
                .IF B   ERR
                        MOV     #LPERR,ERRSVC
                .ENDC
                .IF NB  ERR
                        MOV     ERR,ERRSVC
                .ENDC
                .IF B PRINTS
                        MOV     #1,BUFREP               ; PRINT ONCE DEFAULT
                .ENDC
                .IF NB PRINTS
                        MOV     PRINTS,BUFREP           ; SUPPLY PRINT COUNT
                .ENDC
                        MOV     UNIT,PRINTR             ; SUPPLY UNIT NUMBER
                        JSR     PC,IOCTRL               ;CALL THE DRIVER
                .ENDM


                ; PRINTS IS A PARAMETER CONTROLLING THE NUMBER IF TIMES THE DATA OR
                ; MESSAGE IS TO BE PRINTED (SENT TO THE PRINTER). DEFAULT IS 1.
                ;
                ; A TIMEOUT OF 20. SECONDS IS FURNISHED BASED ON THE FOLLOWING ASSUMPTIONS :
                ;       1   A PRINTER SPEED OF 300 LPM
                ;       2   A REPEAT COUNT OF 88 MAX. ( 1 PAGE OF LINES AT 8 LPI. )
                ;       3   AN INITIAL BAND STARTUP TIME OF 2.5 SECONDS.
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        002306          ENDMOD
```

```
                                .SBTTL  GLOBAL AREAS

002306                                  BGNMOD

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

002306                                  EQUALS
(1)                             ;
(1)                             ; BIT DIFINITIONS
(1)                             ;
(1)         100000              BIT15== 100000
(1)         040000              BIT14== 40000
(1)         020000              BIT13== 20000
(1)         010000              BIT12== 10000
(1)         004000              BIT11== 4000
(1)         002000              BIT10== 2000
(1)         001000              BIT09== 1000
(1)         000400              BIT08== 400
(1)         000200              BIT07== 200
(1)         000100              BIT06== 100
(1)         000040              BIT05== 40
(1)         000020              BIT04== 20
(1)         000010              BIT03== 10
(1)         000004              BIT02== 4
(1)         000002              BIT01== 2
(1)         000001              BIT00== 1
(1)                             ;
(1)         001000              BIT9==  BIT09
(1)         000400              BIT8==  BIT08
(1)         000200              BIT7==  BIT07
(1)         000100              BIT6==  BIT06
(1)         000040              BIT5==  BIT05
(1)         000020              BIT4==  BIT04
(1)         000010              BIT3==  BIT03
(1)         000004              BIT2==  BIT02
(1)         000002              BIT1==  BIT01
(1)         000001              BIT0==  BIT00
(1)                             ;
(1)                             ; EVENT FLAG DEFINITIONS
(1)                             ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1)                             ;
(1)         000040              EF.START==      32.                     ; START COMMAND WAS ISSUED
(1)         000037              EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
(1)         000036              EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
(1)         000035              EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
(1)         000034              EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
(1)                             ;
(1)                             ;
(1)                             ; PRIORITY LEVEL DEFINITIONS
(1)                             ;
(1)         000340              PRI07== 340
(1)         000300              PRI06== 300
(1)         000240              PRI05== 240
```

```
                                GLOBAL AREAS

(1)     000200                  PRI04== 200
(1)     000140                  PRI03== 140
(1)     000100                  PRI02== 100
(1)     000040                  PRI01== 40
(1)     000000                  PRI00== 0
(1)                             ;
(1)                             ;OPERATOR FLAG BITS
(1)                             ;
(1)     000004                  EVL==         4
(1)     000010                  LOT==        10
(1)     000020                  ADR==        20
(1)     000040                  IDU==        40
(1)     000100                  ISR==       100
(1)     000200                  UAM==       200
(1)     000400                  BOE==       400
(1)     001000                  PNT==      1000
(1)     002000                  PRI==      2000
(1)     004000                  IXE==      4000
(1)     010000                  IBE==     10000
(1)     020000                  IER==     20000
(1)     040000                  LOE==     40000
(1)     100000                  HOE==    100000
958
962     000012                  LF==12
963     000014                  FF==14
964     000015                  CR==15
965     000177                  DEL==177
966                             ;
967                             ;PRIORITY LEVEL DEFINITIONS
968                             ;
969     000340                  PRI07== 340
970     000300                  PRI06== 300
971     000240                  PRI05== 240
972     000200                  PRI04== 200
973     000140                  PRI03== 140
974     000100                  PRI02== 100
975     000040                  PRI01== 40
976     000000                  PRI00== 0
977
978                             ;
979                             ;GLOBAL ERROR CODES FOR USE BY GENERAL ERROR ROUTINE
980                             ;
981     000001                  STATER= 1                       ;TRANSMITTER STATUS ERROR IN OUTPUT
982     000002                  TIMOUT= 2                       ;TIMEOUT ERROR IN IO DRIVER MODULE
983                                                             ;THIS ERROR INDICATES THE LAST CHARACTER
984                                                             ;WAS NOT TRANSMITTED WITHIN A GIVEN TIME
985     000003                  NOINTR= 3                       ;GROSS TIME OUT ERROR. THE SPECIFIED DID NOT
986                                                             ;INTERRRUPT. THEREFORE IO DRIVER MODULE WAS
987                                                             ;NOT CALLED
988                             ;
989                             .SBTTL  GENERAL REGISTER USAGE DEFINITIONS
990                             ;
991                             ;R0     RESERVED FOR USE BY THE MACRO PACKAGES
992                             ;R1     MAXIMUM NUMBER OF UNITS TO TEST L$UNIT-1
993                             ;R2     UNIT NUMBER BY 2. USED TO CALCULATE OFFSET INTO PROPER
994                             ;       PRINTER TABLE
```

```
                                ;R3     TEMPORARY STORAGE
                                ;R4          "       "
                                ;R5
                                ;R6     STACK POINTER
                                ;R7     PROGRAM COUNTER
                                ;
                                ;
                                ;
                                ; LP STATUS TABLE BIT DEFINITIONS
                                ;
100000                          ERROR = BIT15
040000                          DROPED = BIT14
020000                          ACTIVE = BIT13
010000                          FLAG96 = BIT12  ; 96 CHAR BAND
                                 ;BIT11
002000                          FLAG27 = BIT10  ; FOR EXPANSION
001000                          FLAG26 = BIT9   ; 0 IF LP25, 1 IF LP26
000377                          LOBYTE = 377    ; BIT MASK FOR CLEARING LOBYTE (COUNTER)
```

```
1015                                    .SBTTL  GLOBAL DATA SECTION
1016
1017
1018
1019  002306  000000                    FLAG:   .WORD   0                       ;<CR> FLAG FOR USE BY SUPERVISOR
1020  002310  000000                    LINCNT: .WORD   0                       ;LINE COUNTER
1021  002312  000000                    LSTCNT: .WORD   0
1022  002314  000000                    COUNT:  .WORD   0
1023  002316  000000                    CCNT:   .WORD   0
1024  002320  000000                    STRCNT: .WORD   0
1025  002322  000000                    CHRGEN: .WORD   0
1026  002324  000000                    UNIT:   .WORD   0                       ;UNIT COUNTER FOR SINGLE UNIT TESTING
1027  002326  000000                    LUNIT:  .WORD   0                       ;UNIT COUNTER FOR ERRORS
1028                                                                            ;AND TESTS NOT USING THE OUTPUT
1029                                                                            ;MACROS.
1030  002330  000000                    PTABAD: .WORD   0                       ;P-TABLE ADDRESS RETURNED BY GPHARD
1031  002332  000000                    PRINTR: .WORD   0                       ;SELECTED LINE NO.
1032                                                                            ;MACRO
1033  002334  000000                    CLKTYP: .WORD   0                       ;CLOCK TYPE CONTROL WORD
1034                                                                            ;1= NO CLOCK AVAILABLE
1035                                                                            ;2= KW11-L LINE CLOCK
1036                                                                            ;3= KW11-P PROGRAMABLE CLOCK
1037  002336  000000                    CLOCKP: .WORD   0                       ; CLOCK P-TABLE ADDRESS
1038  002340  000000                    CLKCSR: .WORD   0                       ;CLOCK CSR ADDRESS
1039  002342  000000                    CLKSET: .WORD   0                       ; CLOCK TIME SET REG ADDRESS
1040  002344  000000                    CLKVEC: .WORD   0                       ;CLOCK VECTOR ADDRESS
1041  002346  000000                    CLKENA: .WORD   0                       ;CLOCK ENABLE BITS
1042  002350  000000                    ERRCOD: .WORD   0                       ;ERROR CODE TYPE FOR GENERAL
1043                                                                            ;ERROR ROUTINE
1044  002352  000000                    ERRFLG: .WORD 0                 ;EXPECTED ERROR INDICATOR
1045  002354  000000                    UUT:    .WORD   0               ; # UNITS ACTUALLY UNDER TEST
1046                                                                            ;EXITS BACK TO IO DRIVER EQUAL
1047                                                                            ;1 IF ERROR WAS EXPECTED.
1048
1049  002356  000000                    INDEX:  .WORD   0
1050  002360  000000                    VFUCMD: .WORD   0
1051                                    ;
1052                                    ;MACRO  VARIABLES
1053
1054  002362  000000                    BUFADD: .WORD   0                       ;BUFFER ADDRESS OF DATA TO BE SENT
1055                                                                            ;TO THE PRINTER
1056  002364  000000                    BUFCNT: .WORD   0                       ;NUMBER OF BYTES TO TRANSFER
1057
1058  002366  000000                    BUFREP: .WORD   0                       ; NUMBER OF TIMES TO PRINT
1059                                    ;
1060                                    ;
1061                                    ;
1062                                    ;LP25 PARAMETER WORD TABLES
1063                                    ;
1064  002370  000020                    LPCSR:  .REPT   16.             ; ADDRESS OF CSR FOR EACH LP11
1065                                            .WORD   0
1066                                            .ENDR
1067  002430  000016                    LPVEC:  .REPT   16              ; INTERRUPT VECTOR ADDRESS
1068                                            .WORD   0
1069                                            .ENDR
1070  002464  000020                    LPBUF:  .REPT 16.               ; DATA BUFFER REGISTER ADDRESS
```

```
1071                                            .WORD   0
1072                                            .ENDR
1073  002524  000020                  STATUS:   .REPT 16.             ; UNIT STATUS
1074                                            .WORD   0
1075                                            .ENDR
1076  002564  000020                  CURADD:   .REPT 16.             ; CURRENT ADDRESS OF OUTPUT DATA BYTE
1077                                            .WORD   0
1078                                            .ENDR
1079  002624  000020                  MSGCNT:   .REPT 16.             ; INITIAL BYTE COUNT OF MSG FOR REPEAT RESTORE
1080                                            .WORD 0
1081                                            .ENDR
1082  002664  000020                  REPCNT:   .REPT   16.           ; NO. OF TIMES TO REPEAT MESSAGE
1083                                            .WORD   0
1084                                            .ENDR
1085  002724  000020                  MSGADR:   .REPT   16.           ; ADDRESS OF DATA TO PRINT START OF DATA
1086                                            .WORD   0
1087                                            .ENDR
1088  002764  000020                  CURCNT:   .REPT 16.             ; CURRENT COUNT REMAINING TO OUTPUT
1089                                            .WORD   -1
1090                                            .ENDR
1091  003024  000020                  LPINTR:   .REPT 16.             ; INTERRUPT ROUTINE ADDRESS
1092                                            .WORD   0
1093                                            .ENDR
1094  003064  000020                  DELCNT:   .REPT 16.
1095                                            .WORD   0             ; TIMEOUT DELAY COUNTER
1096                                            .ENDR
1097  003124  000000                  ERRSVC:   .WORD   0             ; ERROR ROUTINE DISPATCH ADDRESS
1098  003126  000020                  ERRTBL::          .REPT 16.     ; ERROR COUNT FOR EACH UNIT
1099                                            .WORD   0
1100                                            .ENDR
1101
1102  003166  000000                  WORK::    .WORD   0             ; WORK AREA
1103  003170  000000                  WORK1:    .WORD   0
1104
1105
1106
1107                                  .SBTTL  OUTPUT BUFFER
1108                                  ;
1109                                  ;150 BYTES IS RESERVED FOR THE OUTPUT BUFFER AREA
1110                                  ;
1111
1112                                            .EVEN
1113  003172  000226                  OUTBUF:   .REPT   150.
1114                                            .BYTE   0
1115                                            .ENDR
1116
```

```
                                .SBTTL  GLOBAL TEXT SECTION

                                .NLIST BEX
                                ;++
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--
003420  051120  047111  042524  CSRERR: .ASCIZ /PRINTER ERROR/
003436  051120  047111  042524  RDYERR: .ASCIZ /PRINTER NOT READY/
003460  040520  042520  020122  PAPSWI: .ASCIZ /PAPER LOW INTERLOCK SWITCH FAILURE/
003523     110  046501  042515  BNKSWI: .ASCIZ /HAMMER BANK INTERLOCK SWITCH FAILURE/
003570  044103  051101  041501  BNDSWI: .ASCIZ  /CHARACTER BAND INTERLOCK SWITCH FAILURE/
003640  051124  047101  046523  INTER1: .ASCIZ  /TRANSMIT INTERRUPT TIMEOUT/
003673     120  044522  052116  TXERR:  .ASCIZ  /PRINTER STATUS ERROR/
003720  052517  050124  052125  OUTTIM: .ASCIZ  /OUTPUT TIMEOUT ERROR/
003745     125  044516  020124  TXNOIN: .ASCIZ  /UNIT FAILED TO INTERRUPT/
003776  046101  020114  047125  UUTEQ0: .ASCIZ  /ALL UNITS HAVE BEEN DROPPED..RESTART../
004045     045  022516  044501  VFUSEL: .ASCII /%N%AINSURE THAT VFU-FLS SWITCH ON EACH UNIT IS IN THE /
004133     045  022516  021101  VFUSE1: .ASCIZ /%N%A''VFU'' POSITION.%N/
004161     116  020117  046103  NOCLK:  .ASCIZ  /NO CLOCK AVAILABLE FOR TIMING TESTS/<7><7>
        004230                  .EVEN


                                ;
                                ;

                                .LIST BEX
                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

004230  040445  050114  030461  LPDROP: .ASCIZ  /%ALP11 UNIT %D2%A DROPPED FROM TEST%N/
004236  052440  044516  020124
004244  042045  022462  020101
004252  051104  050117  042520
004260  020104  051106  046517
004266  052040  051505  022524
004274  000116
```

```
                                .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.
                                ;
                                ;
004276                          LPERR:  SELECT ERRCOD OF 3 VERIFY       ;SELECT PROPER MESSAGE FORMAT
(2)  004276  013746  002350             MOV     ERRCOD,-(SP)
(4)  004302  003403                     BLE     50000$
(3)  004304  021627  000003             CMP     (SP),#3
(6)  004310  003402                     BLE     50001$
(3)  004312                     50000$:
(3)  004312  012716  000004             MOV     #4,(SP)
(3)  004316                     50001$:
(2)  004316  006316                     ASL     (SP)
(2)  004320  060716                     ADD     PC,(SP)
(2)  004322  063607                     ADD     @(SP)+,PC
(3)  004324                     50002$:
(4)  004324  000010                     .WORD   50006$-50002$
(4)  004326  000036                     .WORD   50005$-50002$
(4)  004330  000064                     .WORD   50004$-50002$
(3)  004332  000110                     .WORD   50003$-50002$
1189
1190 004334                             CASE 1                  ;STATUS ERROR
(4)  004334                     50006$:
1191 004334                             LET ERRTBL(R2) := ERRTBL(R2) + #1
(6)  004334  005262  003126             INC     ERRTBL(R2)
1192 004340                             LET L$LUN := R2 SHIFT -1
(4)  004340  010237  002074             MOV     R2,L$LUN
(7)  004344  006237  002074             ASR     L$LUN
```

```
1193  004350                                        ERRHRD  14,TXERR
 (4)  004350  104456                                TRAP    C$ERHRD
 (5)  004352  000016                                .WORD   14
 (5)  004354  003673                                .WORD   TXERR
 (5)  004356  000000                                .WORD   0
1194
1195  004360                                        CASE 2                  ;OUTPUT TIMEOUT ERROR
 (4)  004360  000425                                BR      50003$
 (4)  004362                                50005$:
1196  004362                                        LET ERRTBL(R2) := ERRTBL(R2) + #1
 (6)  004362  005262  003126                        INC     ERRTBL(R2)
1197  004366                                        LET L$LUN := R2 SHIFT -1
 (4)  004366  010237  002074                        MOV     R2,L$LUN
 (7)  004372  006237  002074                        ASR     L$LUN
1198  004376                                        ERRHRD  15,OUTTIM       ;
 (4)  004376  104456                                TRAP    C$ERHRD
 (5)  004400  000017                                .WORD   15
 (5)  004402  003720                                .WORD   OUTTIM
 (5)  004404  000000                                .WORD   0
1199
1200  004406                                        CASE 3
 (4)  004406  000412                                BR      50003$
 (4)  004410                                50004$:
1201                                                ;  NEVER RECIEVED THE INTERRUPT
1202  004410                                        LET ERRTBL(R2) := ERRTBL(R2) + #1
 (6)  004410  005262  003126                        INC     ERRTBL(R2)
1203  004414                                        LET L$LUN := R2 SHIFT -1
 (4)  004414  010237  002074                        MOV     R2,L$LUN
 (7)  004420  006237  002074                        ASR     L$LUN
1204  004424                                        ERRHRD 16,TXNOIN
 (4)  004424  104456                                TRAP    C$ERHRD
 (5)  004426  000020                                .WORD   16
 (5)  004430  003745                                .WORD   TXNOIN
 (5)  004432  000000                                .WORD   0
1205
1206
1207
1208  004434                                 ENDSELECT
 (3)  004434                                50003$:
1209
1210  004434                                        IF ERRTBL(R2) GT MAXERR THEN
 (6)  004434  026237  003126  002304                 CMP     ERRTBL(R2),MAXERR
 (9)  004442  003402                                BLE     50007$
1211  004444  004737  005534                            JSR PC,DROPIT       ; MAXIMUM ERROR COUNT EXCEEDED !
1212  004450                                        ENDIF
 (4)  004450                                50007$:
1213  004450                                        LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR
 (6)  004450  042762  100000  002524                BIC     #ERROR,STATUS(R2)
1214  004456                                        LET ERRCOD := #0
 (4)  004456  005037  002350                        CLR     ERRCOD
1215  004462                                        LET @LPCSR(R2) := #100          ; CLEAR THE ERROR BIT AND ENABLE INTERRUPTS
 (4)  004462  012772  000100  002370                MOV     #100,@LPCSR(R2)
1216  004470  000207                                RTS     PC                      ;AND EXIT
1217
1218                                       ;=============================================================
1219                                       ; BIN2DA        BINARY TO DECIMAL ASCII CONVERSION ROUTINE
```

```
1220                                ;               ENTER WITH NUMBER TO BE CONVERTED ON THE STACK
1221                                ;               FOLLOWED BY THE ADDRESS OF A 5 BYTE BUFFER
1222                                ;               FOR THE ASCII STRING. 5 DIGITS WILL BE CONVERTED
1223                                ;               LEADING ZEROES WILL BE CONVERTED TO SPACES.
1224                                ;               CALL BY JSR  PC,BIN2DA
1225                                ;=============================================================
1226
1227  004472                        BIN2DA: PUSH R4,R5
 (2)  004472  010446                        MOV     R4,-(SP)
 (3)  004474  010546                        MOV     R5,-(SP)
1228  004476                                LET R4 := 6(SP)         ; GET ADDRESS FOR ASCII STRING
 (4)  004476  016604  000006               MOV     6(SP),R4
1229  004502                                LET R5 := #TABLDA       ; GET ADDRESS OF DECIMAL TABLE
 (4)  004502  012705  004664               MOV     #TABLDA,R5
1230  004506                                LET FLAGDA := #0        ; LEADING ZERO FLAG
 (4)  004506  005037  004676               CLR     FLAGDA
1231  004512                                LET COUNTD := #0
 (4)  004512  005037  004700               CLR     COUNTD
1232                                       ; 8.(SP) HAS NUMBER TO BE CONVERTED
1233  004516                                DECR DIGITS FROM #4 TO #0 BY #1 ; DO 5 DIGITS
 (4)  004516  012737  000004  004702       MOV     #4,DIGITS
 (5)  004524  000402                       BR      50010$
 (4)  004526                        50011$:
 (7)  004526  005337  004702               DEC     DIGITS
 (5)  004532                        50010$:
 (5)  004532  005737  004702               TST     DIGITS
 (7)  004536  002435                       BLT     50012$
1234  004540                                   WHILE 8.(SP) GE (R5) DO     ; CREATE A DIGIT
 (4)  004540                        50013$:
 (6)  004540  026615  000010               CMP     8.(SP),(R5)
 (9)  004544  002405                       BLT     50014$
1235  004546                                   LET 8.(SP) := 8.(SP) - (R5)
 (6)  004546  161566  000010               SUB     (R5),8.(SP)
1236  004552                                   LET COUNTD := COUNTD + #1
 (6)  004552  005237  004700               INC     COUNTD
1237  004556                                ENDDO
 (4)  004556  000770                       BR      50013$
 (3)  004560                        50014$:
1238                                       ; CONVERT DIGIT TO ASCII OR SUPPLY A SPACE
1239  004560                                IF COUNTD GT #0 OR FLAGDA GT #0 THEN
 (6)  004560  005737  004700               TST     COUNTD
 (8)  004564  003003                       BGT     50015$
 (6)  004566  005737  004676               TST     FLAGDA
 (9)  004572  003410                       BLE     50016$
 (6)  004574                        50015$:
1240  004574                                   LET COUNTD := COUNTD SET.BY #60
 (6)  004574  052737  000060  004700       BIS     #60,COUNTD
1241  004602                                   LET (R4)+ :B= COUNTD
 (4)  004602  113724  004700               MOVB    COUNTD,(R4)+
1242  004606                                   LET FLAGDA := FLAGDA + #1
 (6)  004606  005237  004676               INC     FLAGDA
1243  004612                                ELSE
 (4)  004612  000402                       BR      50017$
 (3)  004614                        50016$:
1244  004614                                   LET (R4)+ :B= #40
 (4)  004614  112724  000040               MOVB    #40,(R4)+
```

```
 1245  004620                                          ENDIF
  (4)  004620                               50017$:
 1246                                                       ; DO THE NEXT DIGIT
 1247  004620                                          LET R5 := R5 + #2
  (6)  004620  062705  000002                       ADD     #2,R5
 1248  004624                                          LET COUNTD := #0
  (4)  004624  005037  004700                       CLR     COUNTD
 1249  004630                                       ENDDECR
  (4)  004630  000736                               BR      50011$
  (3)  004632                               50012$:
 1250                                               ; IF NUMBER WAS A ZERO PRINT A '0'
 1251  004632                                       IF FLAGDA EQ #0 THEN
  (6)  004632  005737  004676                       TST     FLAGDA
  (9)  004636  001002                               BNE     50020$
 1252  004640                                          LET -(R4) :B= #60
  (4)  004640  112744  000060                       MOVB    #60,-(R4)
 1253  004644                                       ENDIF
  (4)  004644                               50020$:
 1254                                               ; CLEAN UP THE STACK AND EXIT
 1255  004644                                       LET 8.(SP) := 4(SP)
  (4)  004644  016666  000004  000010               MOV     4(SP),8.(SP)
 1256  004652                                       POP     R5,R4
  (2)  004652  012605                               MOV     (SP)+,R5
  (3)  004654  012604                               MOV     (SP)+,R4
 1257  004656                                       LET SP := SP + #4
  (6)  004656  062706  000004                       ADD     #4,SP
 1258  004662  000207                               RTS     PC
 1259
 1260
 1261  004664  023420  001750  000144  TABLDA: .WORD   10000.,1000.,100.,10.,1
       004672  000012  000001
 1262  004676  000000                      FLAGDA: .WORD   0
 1263  004700  000000                      COUNTD: .WORD   0
 1264  004702  000000                      DIGITS: .WORD   0
 1265
```

```
                                .SBTTL  I/O DRIVER

                                ;
                                ;
                                ;++
                                ;THE I/O DRIVER ROUTINE IS INVOKED BY MEANS OF THE INTERRUPT SYSTEM.
                                ;CALL TO IT IS JMP  IODRV.
                                ;RETURN   RTI.
                                ;ENTER ROUTINE WITH R2 SET UP TO DESIRED UNIT *2. R2 IS USED
                                ;TO CALCULATE OFFSET INTO PROPER TABLE.
                                ;R1 EQUALS MAXIMUM NUMBER OF UNITS ON SYSTEM UNDER TEST.
                                ;
                                ;--
                                ;
                                ; CHECK FOR ERROR FLAG IN STATUS REG.
                                ;
004704                          IODRV:  IF #BIT15 NOTSETIN @LPCSR(R2) THEN
(6) 004704  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
(9) 004712  001061                          BNE     50021$
                                ;
                                ; IF COUNT NOT ZERO SEND NEXT BYTE
                                ;
004714                                          IF CURCNT(R2) GT #0 THEN
(6) 004714  005762  002764                  TST     CURCNT(R2)
(9) 004720  003416                          BLE     50022$
004722                                          LET @LPBUF(R2) :B= @CURADD(R2)
(4) 004722  117272  002564  002464           MOVB    @CURADD(R2),@LPBUF(R2)
004730                                          LET CURADD(R2) := CURADD(R2) + #1
(6) 004730  005262  002564                  INC     CURADD(R2)
                                ;
                                ; ENABLE INTERRUPT FOR NEXT BYTE
                                ;
004734                                          LET STATUS(R2) := STATUS(R2) SET.BY #ACTIVE
(6) 004734  052762  020000  002524          BIS     #ACTIVE,STATUS(R2)
004742                                          LET CURCNT(R2) := CURCNT(R2) - #1
(6) 004742  005362  002764                  DEC     CURCNT(R2)
004746                                          LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100
(6) 004746  052772  000100  002370          BIS     #100,@LPCSR(R2)
                                                ELSE
(4) 004754  000437                          BR      50023$
(3) 004756                      50022$:
                                ; CURRENT MSG DONE, IF PRINT COUNT NOT ZERO SEND AGAIN
004756                                              LET REPCNT(R2) := REPCNT(R2) - #1
(6) 004756  005362  002664                  DEC     REPCNT(R2)
004762                                              IF REPCNT(R2) GT #0 THEN
(6) 004762  005762  002664                  TST     REPCNT(R2)
(9) 004766  003424                          BLE     50024$
004770                                                  LET CURADD(R2) := MSGADR(R2)  ; RESTORE THE MSG ADDR
(4) 004770  016262  002724  002564           MOV     MSGADR(R2),CURADD(R2)
004776                                                  LET CURCNT(R2) := MSGCNT(R2)  ; RESTORE THE BYTE COUNT
(4) 004776  016262  002624  002764           MOV     MSGCNT(R2),CURCNT(R2)
005004                                                  LET @LPBUF(R2) :B= @CURADD(R2)  ; RESEND THE MESSAGE
(4) 005004  117272  002564  002464           MOVB    @CURADD(R2),@LPBUF(R2)
005012                                                  LET CURADD(R2) := CURADD(R2) + #1 ; BUMP THE POINTER
(6) 005012  005262  002564                  INC     CURADD(R2)
005016                                                  LET CURCNT(R2) := CURCNT(R2) - #1 ; DROP BYTE COUNT
```

```
  (6)  005016  005362  002764                    DEC     CURCNT(R2)
1305   005022                                                 LET STATUS(R2) := STATUS(R2) SET.BY #ACTIVE
  (6)  005022  052762  020000  002524            BIS     #ACTIVE,STATUS(R2)
1306   005030                                                 LET @LPCSR(R2) := #100  ; RE-ENABLE INTERRUPTS
  (4)  005030  012772  000100  002370            MOV     #100,@LPCSR(R2)
1307   005036                                                     ELSE
  (4)  005036  000406                            BR      50025$
  (3)  005040                                50024$:
1308                                         ; CURRENT MSG DONE, REPEAT COUNT =0
1309                                         ; CLEAR ACTIVE AND DISABLE INTERRUPTS.
1310   005040                                                 LET STATUS(R2) := STATUS(R2) CLR.BY #ACTIVE
  (6)  005040  042762  020000  002524            BIC     #ACTIVE,STATUS(R2)
1311   005046                                                 LET @LPCSR(R2) := #00
  (4)  005046  012772  000000  002370            MOV     #00,@LPCSR(R2)
1312   005054                                                     ENDIF
  (4)  005054                                50025$:
1313   005054                                             ENDIF
  (4)  005054                                50023$:
1314   005054                                         ELSE
  (4)  005054  000410                            BR      50026$
  (3)  005056                                50021$:
1315                                          ; CLEAR ERROR CONDITION, ENABLE INTERRUPTS
1316                                          ; SET ERROR FLAG
1317   005056                                             LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
  (6)  005056  052762  100000  002524            BIS     #ERROR,STATUS(R2)
1318   005064                                             LET ERRCOD := #STATER       ; STATUS ERROR
  (4)  005064  012737  000001  002350            MOV     #STATER,ERRCOD
1319   005072  004777  176026                         JSR PC,@ERRSVC
1320                                                   ; ERROR SERVICE SHOULD CLEAR ERROR BIT AND ENABLE INTR
1321   005076                                         ENDIF
  (4)  005076                                50026$:
1322   005076                                         POP R2
  (2)  005076  012602                            MOV     (SP)+,R2
1323   005100  000002                            RTI
```

```
                        .SBTTL I/O CONTROL
                        ;++
                        ;
                        ; THE I/O CONTROL SUBROUTINE IS A SINGLE ENTRY QUEUE MANAGER.
                        ; THIS ROUTINE IS INVOKED BY A JSR FROM AN I/O CALL.
                        ; INPUTS:       PRINTR  -1 FOR ALL TERMINALS
                        ;                       N FOR PRINTER NUMBER 'N'
                        ;               BUFADD  ADDRESS OF MESSAGE TO PRINT
                        ;               BUFCNT  BYTE COUNT TO TRANSMIT TO PRINTER
                        ;
                        ;               ERRSVC  ADDRESS OF ERROR SERVICE SUBROUTINE
                        ;               BUFREP IS NO. OF TIMES TO PRINT THE MSG
                        ;--
                        
005102                  IOCTRL: PUSH R2,R3
(2) 005102  010246              MOV     R2,-(SP)
(3) 005104  010346              MOV     R3,-(SP)
                        ;
                        ; IF PRINTR IS -1 QUE OUTPUT TO ALL PRINTERS SELECTED
                        ; OTHERWISE TO UNIT NUMBER IN PRINTR.
                        ;
005106                          IF PRINTR EQ #-1 THEN
(6) 005106  023727  002332  177777      CMP     PRINTR,#-1
(9) 005114  001005              BNE     50027$
005116                              LET R3 := L$UNIT
(4) 005116  013703  002012      MOV     L$UNIT,R3
005122                              LET L$LUN := #0
(4) 005122  005037  002074      CLR     L$LUN
005126                          ELSE
(4) 005126  000405              BR      50030$
(3) 005130              50027$:
005130                              LET R3 := #1
(4) 005130  012703  000001      MOV     #1,R3
005134                              LET L$LUN := PRINTR
(4) 005134  013737  002332  002074  MOV     PRINTR,L$LUN
005142                          ENDIF
(4) 005142              50030$:
                        ;
                        ; REPEAT TILL R3 = 0
                        ;
005142                  CTLLOP:
005142                          IF R3 EQ #0 THEN
(6) 005142  005703              TST     R3
(9) 005144  001002              BNE     50031$
005146                              INLINE <JMP CTLEND>
(2) 005146  000137  005460      JMP     CTLEND
005152                          ENDIF
(4) 005152              50031$:
                        ;
                        ; USE R2 AS AN INDEX INTO THE UNIT TABLES
                        ;
005152                              LET R2 := L$LUN SHIFT 1
(4) 005152  013702  002074      MOV     L$LUN,R2
(7) 005156  006302              ASL     R2
005160                              LET ERRCOD := #0
(4) 005160  005037  002350      CLR     ERRCOD
```

```
1363                                    ;
1364                                    ; IF THE UNIT HAS BEEN DROPPED SELECT THE NEXT UNIT
1365                                    ;
1366  005164                                        IF #DROPED NOTSETIN STATUS(R2) THEN
 (6)  005164  032762  040000  002524          BIT     #DROPED,STATUS(R2)
 (9)  005172  001123                          BNE     50032$
1367                                    ;
1368                                    ;TEST FOR DVC ERROR BIT SET
1369                                    ;
1370  005174                                            IF #BIT15 SETIN @LPCSR(R2) THEN
 (6)  005174  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
 (9)  005202  001407                          BEQ     50033$
1371  005204                                                LET ERRCOD := #STATER       ; STATUS REG ERROR BIT 15 SET IN CSR
 (4)  005204  012737  000001  002350          MOV     #STATER,ERRCOD
1372  005212                                                LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
 (6)  005212  052762  100000  002524          BIS     #ERROR,STATUS(R2)
1373  005220                                            ELSE
 (4)  005220  000455                          BR      50034$
 (3)  005222                            50033$:
1374                                    ;
1375                                    ; MAKE SURE PREVIOUS MSG IS DONE
1376                                    ;
1377  005222                                            IF CURCNT(R2) GT #0 THEN
 (6)  005222  005762  002764                  TST     CURCNT(R2)
 (9)  005226  003452                          BLE     50035$
1378  005230                                                IF #ACTIVE NOTSETIN STATUS(R2) THEN
 (6)  005230  032762  020000  002524          BIT     #ACTIVE,STATUS(R2)
 (9)  005236  001004                          BNE     50036$
1379                                    ;
1380                                    ; OUTPUT WAS QUEUED BUT I/O DRIVER WAS NEVER INVOKED (VIA INTERRUPT)
1381                                    ;
1382  005240                                                LET ERRCOD := #NOINTR       ; NO INTERRUPT
 (4)  005240  012737  000003  002350          MOV     #NOINTR,ERRCOD
1383  005246                                                ELSE
 (4)  005246  000442                          BR      50037$
 (3)  005250                            50036$:
1384  005250                                                WHILE #ACTIVE SETIN STATUS(R2) DO
 (4)  005250                            50040$:
 (6)  005250  032762  020000  002524          BIT     #ACTIVE,STATUS(R2)
 (9)  005256  001436                          BEQ     50041$
1385
1386  005260                                                LET DELCNT(R2) := #100.       ; 20 SEC. DELAY MAX
 (4)  005260  012762  000144  003064          MOV     #100.,DELCNT(R2)
1387  005266                                                DELAY 2.        ; 200MS LOOPS
 (2)  005266  012727  000002                  MOV     #2.,(PC)+
 (2)  005272  000000                          .WORD   0
 (2)  005274  013727  002116                  MOV     L$DLY,(PC)+
 (2)  005300  000000                          .WORD   0
 (2)  005302  005367  177772                  DEC     -6(PC)
 (2)  005306  001375                          BNE     .-4
 (2)  005310  005367  177756                  DEC     -22(PC)
 (2)  005314  001367                          BNE     .-20
1388  005316                                                LET DELCNT(R2) := DELCNT(R2) - #1
 (6)  005316  005362  003064                  DEC     DELCNT(R2)
1389  005322                                                IF DELCNT(R2) EQ #0 THEN
 (6)  005322  005762  003064                  TST     DELCNT(R2)
```

```
  (9)  005326  001011                          BNE     50042$
1390   005330                                                  LET ERRCOD := #TIMOUT
  (4)  005330  012737  000002  002350          MOV     #TIMOUT,ERRCOD
1391   005336                                                  LET STATUS(R2) := STATUS(R2) CLR.BY #ACTIVE
  (6)  005336  042762  020000  002524          BIC     #ACTIVE,STATUS(R2)
1392   005344                                                  LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
  (6)  005344  052762  100000  002524          BIS     #ERROR,STATUS(R2)
1393   005352                                                  ENDIF
  (4)  005352                          50042$:
1394   005352                                                  ENDDO
  (4)  005352  000736                          BR      50040$
  (3)  005354                          50041$:
1395   005354                                                  ENDIF
  (4)  005354                          50037$:
1396   005354                                              ENDIF
  (4)  005354                          50035$:
1397   005354                                              ENDIF
  (4)  005354                          50034$:
1398   005354                                              IF ERRCOD NE #0 THEN
  (6)  005354  005737  002350                  TST     ERRCOD
  (9)  005360  001403                          BEQ     50043$
1399                                           ;
1400                                           ; REPORT THE ERROR
1401                                           ;
1402   005362  004777  175536                                  JSR     PC,@ERRSVC
1403   005366                                                  ELSE
  (4)  005366  000425                          BR      50044$
  (3)  005370                          50043$:
1404                                           ;
1405                                           ; Q UP THE MESSAGE AND ENABLE INTERRUPTS
1406                                           ; THE I/O DRIVER WILL PICK UP FROM HERE.
1407                                           ;
1408   005370                                                  LET CURADD(R2) := BUFADD         ; BYTE ADDRESS
  (4)  005370  013762  002362  002564          MOV     BUFADD,CURADD(R2)
1409   005376                                                  LET MSGADR(R2) := BUFADD         ; MESSAGE ADDRESS
  (4)  005376  013762  002362  002724          MOV     BUFADD,MSGADR(R2)
1410   005404                                                  LET CURCNT(R2) := BUFCNT         ; OUTPUT COUNT
  (4)  005404  013762  002364  002764          MOV     BUFCNT,CURCNT(R2)
1411   005412                                                  LET MSGCNT(R2) := BUFCNT         ; BYTE COUNT
  (4)  005412  013762  002364  002624          MOV     BUFCNT,MSGCNT(R2)
1412   005420                                                  LET REPCNT(R2) := BUFREP         ; PRINT COUNT
  (4)  005420  013762  002366  002664          MOV     BUFREP,REPCNT(R2)
1413   005426                                                  IF CURCNT(R2) GT #0 THEN
  (6)  005426  005762  002764                  TST     CURCNT(R2)
  (9)  005432  003403                          BLE     50045$
1414   005434                                                          LET @LPCSR(R2) := #100   ; ENABLE INTERRUPTS
  (4)  005434  012772  000100  002370          MOV     #100,@LPCSR(R2)
1415   005442                                                  ENDIF
  (4)  005442                          50045$:
1416   005442                                              ENDIF
  (4)  005442                          50044$:
1417   005442                                          ENDIF
  (4)  005442                          50032$:
1418                                           ;
1419                                           ; CLEAR OUT ANY TIMEOUT COUNT
1420                                           ;
```

```
 1421  005442                                        LET DELCNT(R2) := #0
  (4)  005442  005062  003064                  CLR     DELCNT(R2)
 1422                                          ;
 1423                                          ; SELECT THE NEXT UNIT AND DECRIMENT THE LINECOUNT
 1424                                          ;
 1425  005446                                        LET R3 := R3 - #1
  (6)  005446  005303                          DEC     R3
 1426  005450                                        LET L$LUN := L$LUN + #1
  (6)  005450  005237  002074                  INC     L$LUN
 1427  005454  000137  005142                  JMP     CTLLOP
 1428  005460                                CTLEND:
 1429  005460                                        POP  R3,R2
  (2)  005460  012603                          MOV     (SP)+,R3
  (3)  005462  012602                          MOV     (SP)+,R2
 1430  005464  000207                          RTS     PC
 1431
 1432                                          ;++++
 1433                                          ; SUBROUTINE    QUIET
 1434                                          ;
 1435                                          ; THIS SUBROUTINE WILL EFFECTIVLY DELAY UNTIL ALL QUEUED OUTPUT
 1436                                          ; IS FINISHED. THE DELAY IS ACCOMPLISHED BY QUEUEING A NULL
 1437                                          ; MESSAGE TO ALL LINES.
 1438                                          ;-----
 1439
 1440  005466                                QUIET:  OUTPUT  #0,#0           ; NULL MESSAGE OUTPUT
 1441  005530  000240                                NOP
 1442  005532  000207                                RTS     PC
 1443
```

```
 1445                                ;=================================================================
 1446                                ; DROPIT        FUNCTIONAL DESCRIPTION :
 1447                                ;
 1448                                ; THIS SUBROUTINE IS USED TO DROP A BAD PRINTER FROM THE TEST
 1449                                ; DISABLE ANY INTERRUPTS FROM THE PRINTER, AND NOTIFY THE
 1450                                ; OPERATOR THAT THE PRINTER WAS DROPPED.
 1451                                ;=================================================================
 1452
 1453  005534                        DROPIT: LET STATUS(R2) := STATUS(R2) SET.BY #DROPED
  (6)  005534  052762  040000  002524         BIS     #DROPED,STATUS(R2)
 1454  005542                                LET CURCNT(R2) := #-1
  (4)  005542  012762  177777  002764         MOV     #-1,CURCNT(R2)
 1455  005550                                LET @LPCSR(R2) := #0
  (4)  005550  005072  002370                 CLR     @LPCSR(R2)
 1456  005554                                PRINTF "LPDROP, L$LUN
  (8)  005554  013746  002074                 MOV     L$LUN,-(SP)
  (7)  005560  012746  004230                 MOV     #LPDROP,-(SP)
  (6)  005564  012746  000002                 MOV     #2,-(SP)
  (3)  005570  010600                         MOV     SP,R0
  (4)  005572  104417                         TRAP    C$PNTF
  (4)  005574  062706  000006                 ADD     #6,SP
 1457  005600                                LET ERRTBL(R2) := #0
  (4)  005600  005062  003126                 CLR     ERRTBL(R2)
 1458  005604                                LET UUT := UUT - #1
  (6)  005604  005337  002354                 DEC     UUT
 1459  005610                                IF UUT EQ #0 THEN
  (6)  005610  005737  002354                 TST     UUT
  (9)  005614  001011                         BNE     50046$
 1460  005616                                PRINTF #UUTEQ0
  (7)  005616  012746  003776                 MOV     #UUTEQ0,-(SP)
  (6)  005622  012746  000001                 MOV     #1,-(SP)
  (3)  005626  010600                         MOV     SP,R0
  (4)  005630  104417                         TRAP    C$PNTF
  (4)  005632  062706  000004                 ADD     #4,SP
 1461  005636                                    DOCLN       ; NOTHING TO TEST
  (3)  005636  104444                         TRAP    C$DCLN
 1462  005640                                ENDIF
  (4)  005640                        50046$:
 1463  005640  000207                         RTS     PC
 1464
 1465
 1466
 1467                                ;=================================================================
 1468                                ; FAKE          FUNCTIONAL DESCRIPTION:
 1469                                ;
 1470                                ; THIS SUBROUTINE IS REQUIRED TO INSURE PROPER PASS COUNT REPORTS
 1471                                ; IN A MULTI UNIT MODE OF OPERATION.
 1472                                ;=================================================================
 1473
 1474  005642                        FAKE:   LET L$LUN := #0
  (4)  005642  005037  002074                 CLR     L$LUN
 1475  005646                                WHILE L$LUN LT L$UNIT DO
  (4)  005646                        50047$:
  (6)  005646  023737  002074  002012         CMP     L$LUN,L$UNIT
  (9)  005654  002007                         BGE     50050$
 1476  005656                                    GPHARD      L$LUN, R3
```

```
   (3)  005656  013700  002074                    MOV     L$LUN,R0
   (3)  005662  104442                            TRAP    C$GPHRD
   (3)  005664  010003                            MOV     R0,R3
  1477  005666                                        LET L$LUN := L$LUN + #1
   (6)  005666  005237  002074                    INC     L$LUN
  1478  005672                                    ENDDO
   (4)  005672  000765                            BR      50047$
   (3)  005674                          50050$:
  1479  005674  000207                            RTS     PC
  1480
  1481
  1482  005676                                    ENDMOD
```

```
                                .SBTTL  INITIALIZATION SECTION
                                ;++
                                ;THE INITIALIZE ROUTINE IS EXECUTED AT THE BEGINNING OF EACH SUB-PASS AND IS
                                ;PRIMARILY USED FOR REQUESTING P-TABLE PARAMETERS. INFORMATION REQUESTED FROM
                                ;THE OPERATOR INCLUDE THE NUMBER OF UNITS UNDER TEST, DEVICE ADDRESSES, VECTORS,
                                ;CLOCK TYPE, AUTO OR MANUAL PRINTING SPEED MEASUREMENT, AND WHETHER A DAVFU
                                ;OPTION IS INSTALLED IN THE SYSTEM.
                                ;--
005676                          BGNMOD
005676                          BGNINIT
 (3) 005676                     L$INIT::
                                ;RESET EXTERNAL BUS IF START EVENT FLAG IS SET
                                ;OR POWER FAIL RESTART
005676                                  READEF  #EF.START               ;TEST START EF INDICATOR
 (3) 005676  012700  000040             MOV     #EF.START,R0
 (3) 005702  104447                     TRAP    C$REFG
005704                                  BCOMPLETE 1$                    ;BRANCH IF FROM START UP
 (2) 005704  103413                     BCS     1$
005706                                  READEF  #EF.RESTART             ;NOW THE RESTARTFLAG
 (3) 005706  012700  000037             MOV     #EF.RESTART,R0
 (3) 005712  104447                     TRAP    C$REFG
005714                                  BCOMPLETE 1$                    ;IF EITHER START OR POWER FAIL RESTART
 (2) 005714  103407                     BCS     1$
                                                                        ;DO A BUS RESET
005716  004737  005642                 JSR     PC,FAKE                 ; UPDATE PASS COUNT
005722                                  SETPRI  #PRI00                  ; PRIORITY ZERO
 (3) 005722  012700  000000             MOV     #PRI00,R0
 (3) 005726  104441                     TRAP    C$SPRI
005730                                  EXIT INIT                       ; ELSE EXIT INIT CODE
 (3) 005730  104432                     TRAP    C$EXIT
 (3) 005732  001332                     .WORD   L10004-.
                                ;
                                ;POWER UP RESTART OR START COMMAND ISSUED
                                ;
005734                          1$:     BRESET                          ;RESET THE BUS
 (3) 005734  104433                     TRAP    C$RESET
005736                                  IF L$UNIT GT #16. THEN
 (6) 005736  023727  002012  000020     CMP     L$UNIT,#16.
 (9) 005744  003420                     BLE     50051$
                                            PRINTF #NRGT16
 (7) 005746  012746  006654             MOV     #NRGT16,-(SP)
 (6) 005752  012746  000001             MOV     #1,-(SP)
 (3) 005756  010600                     MOV     SP,R0
 (4) 005760  104417                     TRAP    C$PNTF
 (4) 005762  062706  000004             ADD     #4,SP
                                            PRINTF #NRGT17
 (7) 005766  012746  006737             MOV     #NRGT17,-(SP)
 (6) 005772  012746  000001             MOV     #1,-(SP)
 (3) 005776  010600                     MOV     SP,R0
 (4) 006000  104417                     TRAP    C$PNTF
 (4) 006002  062706  000004             ADD     #4,SP
006006                                  ENDIF
 (4) 006006                     50051$:
006006                                  MANUAL          ; WAIT FOR CR IF IN MANUAL MODE
 (3) 006006  104450                     TRAP    C$MANI
006010                                  BNCOMPLETE      100$
```

```
                        INITIALIZATION SECTION

  (2)  006010  103016                       BCC     100$

1514
1515  006012                                PRINTF  #MRESET                 ;PRINT RESET MESSAGE
  (7)  006012  012746  006773               MOV     #MRESET,-(SP)
  (6)  006016  012746  000001               MOV     #1,-(SP)
  (3)  006022  010600                       MOV     SP,R0
  (4)  006024  104417                       TRAP    C$PNTF
  (4)  006026  062706  000004               ADD     #4,SP
1516                                    ;
1517                                    ;WAIT FOR A "CR" BEFORE GOING ON
1518                                    ;
1519  006032                                LET FLAG := #0
  (4)  006032  005037  002306               CLR     FLAG
1520  006036                                LET ERRCOD := #0
  (4)  006036  005037  002350               CLR     ERRCOD
1521  006042                                LET UUT := #0
  (4)  006042  005037  002354               CLR     UUT
1522  006046                            100$:
1523  006046                                GMANIL  READY,FLAG,100000,YES   ;GET MANUAL PARAMETERS
  (3)  006046  104443                       TRAP    C$GMAN
  (3)  006050  000404                       BR      10000$
  (4)  006052  002306                       .WORD   FLAG
  (5)  006054  000130                       .WORD   T$CODE
  (5)  006056  007072                       .WORD   READY
  (5)  006060  100000                       .WORD   100000
  (3)  006062                           10000$:
1524                                    ;
1525                                    ;REQUEST P-TABLE FOR PRINTERS UNDER TEST
1526                                    ;
1527  006062                            2$:     LET R1 := L$UNIT - #1           ;MAXIMUM NUMBER OF UNITS
  (4)  006062  013701  002012               MOV     L$UNIT,R1
  (6)  006066  005301                       DEC     R1
1528  006070                                INCR L$LUN FROM #0 TO R1 BY #1
  (4)  006070  005037  002074               CLR     L$LUN
  (5)  006074  000402                       BR      50052$
  (4)  006076                           50053$:
  (7)  006076  005237  002074               INC     L$LUN
  (5)  006102                           50052$:
  (5)  006102  023701  002074               CMP     L$LUN,R1
  (7)  006106  003132                       BGT     50054$
1529  006110                                    GPHARD L$LUN,R3             ;REQUEST P-TABLE ADDRESS
  (3)  006110  013700  002074               MOV     L$LUN,R0
  (3)  006114  104442                       TRAP    C$GPHRD
  (3)  006116  010003                       MOV     R0,R3
1530  006120                                    INCOMPLETE 3$               ;BRANCH IF DEVICE NOT PRESENT
  (2)  006120  103121                       BCC     3$
1531  006122                                    LET R2 := L$LUN SHIFT 1
  (4)  006122  013702  002074               MOV     L$LUN,R2
  (7)  006126  006302                       ASL     R2
1532                                    ;
1533                                    ; CLEAR ERROR COUNT, OUTPUT COUNT, GET DEVICE TYPE TO STATUS.
1534                                    ;
1535  006130                                IF 4(R3) EQ #0 THEN
  (6)  006130  005763  000004               TST     4(R3)
  (9)  006134  001004                       BNE     50055$
1536  006136                                    LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG26!FLAG27
```

```
   (6)  006136  042762  003000  002524          BIC     #FLAG26!FLAG27,STATUS(R2)
 1537   006144                                   ELSE
   (4)  006144  000416                          BR      50056$
   (3)  006146                              50055$:
 1538   006146                                       IF 4(R3) EQ #1 THEN
   (6)  006146  026327  000004  000001          CMP     4(R3),#1
   (9)  006154  001007                          BNE     50057$
 1539   006156                                       LET STATUS(R2) := STATUS(R2) SET.BY #FLAG26
   (6)  006156  052762  001000  002524          BIS     #FLAG26,STATUS(R2)
 1540   006164                                       LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG27
   (6)  006164  042762  002000  002524          BIC     #FLAG27,STATUS(R2)
 1541   006172                                   ELSE
   (4)  006172  000403                          BR      50060$
   (3)  006174                              50057$:
 1542   006174                                       LET STATUS(R2) := STATUS(R2) SET.BY #FLAG26!FLAG27
   (6)  006174  052762  003000  002524          BIS     #FLAG26!FLAG27,STATUS(R2)
 1543   006202                                   ENDIF
   (4)  006202                              50060$:
 1544   006202                                   ENDIF
   (4)  006202                              50056$:
 1545                                       ;
 1546                                       ; NOW GET THE BAND TYPE  64 OR 96 CHARACTER
 1547                                       ;
 1548   006202                                       IF 6(R3) EQ #0 THEN
   (6)  006202  005763  000006                  TST     6(R3)
   (9)  006206  001004                          BNE     50061$
 1549   006210                                       LET STATUS(R2) := STATUS(R2) CLR.BY #FLAG96
   (6)  006210  042762  010000  002524          BIC     #FLAG96,STATUS(R2)
 1550   006216                                   ELSE
   (4)  006216  000403                          BR      50062$
   (3)  006220                              50061$:
 1551   006220                                       LET STATUS(R2) := STATUS(R2) SET.BY #FLAG96
   (6)  006220  052762  010000  002524          BIS     #FLAG96,STATUS(R2)
 1552   006226                                   ENDIF
   (4)  006226                              50062$:
 1553   006226                                       LET ERRTBL(R2) := #0
   (4)  006226  005062  003126                  CLR     ERRTBL(R2)
 1554   006232                                       LET CURCNT(R2) := #-1
   (4)  006232  012762  177777  002764          MOV     #-1,CURCNT(R2)
 1555   006240                                       LET DELCNT(R2) := #0
   (4)  006240  005062  003064                  CLR     DELCNT(R2)
 1556   006244                                       LET REPCNT(R2) := #0
   (4)  006244  005062  002664                  CLR     REPCNT(R2)
 1557                                       ;
 1558                                       ;LOAD CSR ADDRESS INTO TABLE
 1559                                       ;
 1560   006250                                       LET LPCSR(R2) := (R3)+      ;SET UP CSR ADDRESS FOR DEVICE
   (4)  006250  012362  002370                  MOV     (R3)+,LPCSR(R2)
 1561   006254                                       LET LPBUF(R2) := LPCSR(R2) + #2
   (4)  006254  016262  002370  002464          MOV     LPCSR(R2),LPBUF(R2)
   (6)  006262  062762  000002  002464          ADD     #2,LPBUF(R2)
 1562                                       ;
 1563                                       ;SET UP VECTOR ADDRESS INTO GIVEN TABLE
 1564                                       ;
 1565   006270                                       LET LPVEC(R2) := (R3)+
   (4)  006270  012362  002430                  MOV     (R3)+,LPVEC(R2)
```

```
 1566                                          ;
 1567                                          ;SET UP DEVICE INTERRUPT VECTOR INFORMATION
 1568                                          ;
 1569   006274                                         LET WORK := R2 SHIFT 3
  (4)   006274  010237  003166                         MOV     R2,WORK
  (7)   006300  006337  003166                         ASL     WORK
  (7)   006304  006337  003166                         ASL     WORK
  (7)   006310  006337  003166                         ASL     WORK
 1570   006314                                         LET WORK := WORK + #INT00
  (6)   006314  062737  040466  003166                 ADD     #INT00,WORK
 1571   006322                                         LET LPINTR(R2) := WORK
  (4)   006322  013762  003166  003024                 MOV     WORK,LPINTR(R2)
 1572   006330                                         SETVEC LPVEC(R2), LPINTR(R2), #PRI04
  (7)   006330  012746  000200                         MOV     #PRI04,-(SP)
  (6)   006334  016246  003024                         MOV     LPINTR(R2),-(SP)
  (5)   006340  016246  002430                         MOV     LPVEC(R2),-(SP)
  (4)   006344  012746  000003                         MOV     #3,-(SP)
  (3)   006350  104437                                 TRAP    C$SVEC
  (2)   006352  062706  000010                         ADD     #10,SP
 1573                                          ;
 1574                                          ; ADD ONE TO UNIT UNDER TEST COUNT
 1575                                          ;
 1576   006356                                         LET UUT := UUT + #1
  (6)   006356  005237  002354                         INC     UUT
 1577   006362  000403                                         BR  4$
 1578                                          ;
 1579                                          ;INDICATE L$LUN NOT AVAILABLE FOR TESTING
 1580                                          ;
 1581   006364                                 3$:             LET STATUS(R2) := STATUS(R2) SET.BY #DROPED
  (6)   006364  052762  040000  002524                 BIS     #DROPED,STATUS(R2)
 1582   006372                                 4$:     ENDINC                          ;GO BACK AND DO IT AGAIN
  (4)   006372  000641                                 BR      50053$
  (5)   006374                                 50054$:
 1583                                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 1584                                          ; SETUP TO HANDLE CLOCK INTERRUPTS
 1585                                          ; IF AN L-CLOCK IS ON THE SYSTEM THEN SETUP A NOOP INTERRUPT
 1586                                          ; HANDLER BECAUSE LSI SYSTEMS MAY HAVE THE CLOCK ENABLED AT ALL TIMES.
 1587   006374                                         LET CLKTYP := #1        ; DEFAULT FOR NO CLOCK ON SYSTEM
  (4)   006374  012737  000001  002334                 MOV     #1,CLKTYP
 1588   006402                                         CLOCK   L,R4            ; TEST FOR L-CLOCK
  (3)   006402  012700  000114                         MOV     #'L,R0
  (3)   006406  104462                                 TRAP    C$CLCK
  (3)   006410  010004                                 MOV     R0,R4
 1589   006412                                         IFCOND CS THEN          ; WE HAVE AN L-CLOCK
  (6)   006412  103031                                 BCC     50063$
 1590   006414                                                 LET CLKTYP := #2
  (4)   006414  012737  000002  002334                 MOV     #2,CLKTYP
 1591   006422                                                 LET CLOCKP := R4
  (4)   006422  010437  002336                         MOV     R4,CLOCKP
 1592   006426                                                 LET CLKCSR := @CLOCKP
  (4)   006426  017737  173704  002340                 MOV     @CLOCKP,CLKCSR
 1593   006434                                                 LET @CLKCSR := #00  ; TRY TO DISABLE INTERRUPTS
  (4)   006434  012777  000000  173676                 MOV     #00,@CLKCSR
 1594                                                          ; SETUP THE NOOP HANDLER
 1595   006442                                                 LET CLKVEC := 4(R4)
  (4)   006442  016437  000004  002344                 MOV     4(R4),CLKVEC
```

```
1596  006450                                        SETVEC  CLKVEC,#IGNORE,#PRI06
 (7)  006450  012746  000300                        MOV     #PRI06,-(SP)
 (6)  006454  012746  007266                        MOV     #IGNORE,-(SP)
 (5)  006460  013746  002344                        MOV     CLKVEC,-(SP)
 (4)  006464  012746  000003                        MOV     #3,-(SP)
 (3)  006470  104437                                TRAP    C$SVEC
 (2)  006472  062706  000010                        ADD     #10,SP
1597  006476                                        ENDIF
 (4)  006476                                50063$:
1598                                        ; IF THE OPERATOR WANTS MANUAL SPEED TEST SET CLOCK TYPE = 4
1599  006476  005737  002276                        TST MANSPD
1600  006502  001410                                BEQ     CK1
1601  006504                                        LET CLKTYP := #4
 (4)  006504  012737  000004  002334                MOV     #4,CLKTYP
1602  006512                                        SETPRI #PRI00           ; START TEST AT PRI 0
 (3)  006512  012700  000000                        MOV     #PRI00,R0
 (3)  006516  104441                                TRAP    C$SPRI
1603  006520                                        EXIT INIT
 (3)  006520  104432                                TRAP    C$EXIT
 (3)  006522  000542                                .WORD   L10004-.
1604                                        ; IF A P-CLOCK IS ON THE SYSTEM UPGRADE CLOCK TYPE TO 3
1605  006524                                CK1:    CLOCK   P,R4
 (3)  006524  012700  000120                        MOV     #'P,R0
 (3)  006530  104462                                TRAP    C$CLCK
 (3)  006532  010004                                MOV     R0,R4
1606  006534                                        IFCOND CS THEN          ; WE HAVE A P-CLOCK
 (6)  006534  103016                                BCC     50064$
1607  006536                                            LET CLKTYP := #4
 (4)  006536  012737  000004  002334                MOV     #4,CLKTYP
1608  006544                                            LET CLOCKP := R4
 (4)  006544  010437  002336                        MOV     R4,CLOCKP
1609  006550                                            LET CLKCSR := @CLOCKP
 (4)  006550  017737  173562  002340                MOV     @CLOCKP,CLKCSR
1610  006556                                            LET CLKVEC := 4(R4)
 (4)  006556  016437  000004  002344                MOV     4(R4),CLKVEC
1611                                                ; TRY TO DISABLE THE P-CLOCK
1612  006564                                            LET @CLKCSR := #00
 (4)  006564  012777  000000  173546                MOV     #00,@CLKCSR
1613  006572                                        ENDIF
 (4)  006572                                50064$:
1614                                        ; IF NO CLOCKS ON THE SYSTEM NOTIFY THE OPERATOR
1615  006572                                        IF CLKTYP EQ #1 THEN
 (6)  006572  023727  002334  000001                CMP     CLKTYP,#1
 (9)  006600  001020                                BNE     50065$
1616  006602                                            PRINTF #NOCLCK
 (7)  006602  012746  007127                        MOV     #NOCLCK,-(SP)
 (6)  006606  012746  000001                        MOV     #1,-(SP)
 (3)  006612  010600                                MOV     SP,R0
 (4)  006614  104417                                TRAP    C$PNTF
 (4)  006616  062706  000004                        ADD     #4,SP
1617  006622                                            PRINTF #NOTIM
 (7)  006622  012746  007171                        MOV     #NOTIM,-(SP)
 (6)  006626  012746  000001                        MOV     #1,-(SP)
 (3)  006632  010600                                MOV     SP,R0
 (4)  006634  104417                                TRAP    C$PNTF
 (4)  006636  062706  000004                        ADD     #4,SP
```

```
 1618  006642                                              ENDIF
  (4)  006642                                      50065$:
 1619  006642                                              SETPRI #PRI00
  (3)  006642  012700  000000                              MOV     #PRI00,R0
  (3)  006646  104441                                      TRAP    C$SPRI
 1620  006650                                      EXIT INIT
  (3)  006650  104432                                      TRAP    C$EXIT
  (3)  006652  000412                                      .WORD   L10004-.
 1621                                              .NLIST BEX
 1622
 1623  006654  047045  040445  052516  NRGT16: .ASCIZ  /%N%ANUMBER OF LINE PRINTERS UNDER TEST EXCEEDS 16./
 1624  006737     045  022516  047501  NRGT17: .ASCIZ  /%N%AONLY 16 WILL BE TESTED./
 1625  006773     045  022516  051101  MRESET: .ASCIZ  /%N%ARESET LINE PRINTER(S), DO FORM FEED,  AND PLACE ON LINE.%N/
 1626
 1627  007072  042504  051120  051505  READY:  .ASCIZ  /DEPRESS ''RETURN'' WHEN READY./
 1628  007127     045  022516  044101  NOCLCK: .ASCIZ  /%N%AHARDWARE CLOCK NOT AVAILABLE./
 1629  007171     045  022516  040501  NOTIM:  .ASCIZ  /%N%AAUTO PRINTING SPEED MEASUREMENT CANNOT BE PERFORMED./
 1630                                      .EVEN
 1631  007262  000000                      PLOC:   .WORD   0
 1632
 1633                                      .LIST BEX
 1634  007264                              ENDINIT
  (3)  007264                              L10004:
  (3)  007264  104411                              TRAP    C$INIT
 1635
 1636                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 1637                                      ; IGNORE        AN INTERRUPT CATCHER FOR THE L-CLOCK
 1638                                      ;               THAT IGNORES THE INTERRUPT.
 1639                                      ;               USED FOR SYSTEMS WHERE CLOCK CANNOT BE TURNED OFF.
 1640                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 1641
 1642  007266                              IGNORE:         ; NOOP
 1643  007266  000002                              RTI
 1644
 1645
 1646
 1647                                      ;-----------------------------------------------------------------
 1648                                      ;
 1649                                      ; RESVEC        FUNCTIONAL DESCRIPTION
 1650                                      ;
 1651                                      ;       THIS SUBROUTINE WILL SETUP ALL UNITS VECTOR AREAS
 1652                                      ;       TO THE 'NORMAL' INTERRUPT ROUTINES STARTING AT INT00.
 1653                                      ;-----------------------------------------------------------------
 1654
 1655  007270                              RESVEC::        PUSH    R3,R4
  (2)  007270  010346                              MOV     R3,-(SP)
  (3)  007272  010446                              MOV     R4,-(SP)
 1656  007274                                              LET  R4 := #0
  (4)  007274  005004                              CLR     R4
 1657  007276                                              LET R3 := L$UNIT
  (4)  007276  013703  002012                      MOV     L$UNIT,R3
 1658  007302                                              WHILE R3 GT #0 DO
  (4)  007302                              50066$:
  (6)  007302  005703                              TST     R3
  (9)  007304  003417                              BLE     50067$
 1659  007306                                                   SETVEC LPVEC(R4), LPINTR(R4), #PRI04
```

```
   (7)  007306  012746  000200                    MOV     #PRI04,-(SP)
   (6)  007312  016446  003024                    MOV     LPINTR(R4),-(SP)
   (5)  007316  016446  002430                    MOV     LPVEC(R4),-(SP)
   (4)  007322  012746  000003                    MOV     #3,-(SP)
   (3)  007326  104437                            TRAP    C$SVEC
   (2)  007330  062706  000010                    ADD     #10,SP
1660    007334                                              LET R4 := R4 + #2
   (6)  007334  062704  000002                    ADD     #2,R4
1661    007340                                              LET R3 := R3 - #1
   (6)  007340  005303                            DEC     R3
1662    007342                                              ENDDO
   (4)  007342  000757                            BR      50066$
   (3)  007344                              50067$:
1663    007344                                              POP R4,R3
   (2)  007344  012604                            MOV     (SP)+,R4
   (3)  007346  012603                            MOV     (SP)+,R3
1664    007350  000207                                      RTS     PC
1665
```

```
                                        CLEANUP CODING SECTION

                                        .SBTTL  CLEANUP CODING SECTION
007352                                  STARS
 (2)                                    ;;*****************************************************************
                                        ;++
                                        ;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.
                                        ;--
007352                                  STARS
 (2)                                    ;;*****************************************************************
007352                                  BGNCLN
 (3)  007352                            L$CLEAN::
007352                                          SETPRI #PRI07
 (3)  007352  012700  000340                    MOV     #PRI07,R0
 (3)  007356  104441                            TRAP    .C$SPRI
007360                                      BRESET
 (3)  007360  104433                            TRAP    C$RESET

007362                                  CLEAN:  LET R1 := L$UNIT - #1   ;NUMBER OF UNITS-1
 (4)  007362  013701  002012                    MOV     L$UNIT,R1
 (6)  007366  005301                            DEC     R1
007370                                          INCR L$LUN FROM #0 TO R1 BY #1
 (4)  007370  005037  002074                    CLR     L$LUN
 (5)  007374  000402                            BR      50070$
 (4)  007376                            50071$:
 (7)  007376  005237  002074                    INC     L$LUN
 (5)  007402                            50070$:
 (5)  007402  023701  002074                    CMP     L$LUN,R1
 (7)  007406  003020                            BGT     50072$
                                                ; DISABLE ALL INTERRUPTS, SELECT ALL LINES
                                                ; ZERO ALL ERROR COUNTS
007410                                              LET R2 := L$LUN SHIFT 1
 (4)  007410  013702  002074                    MOV     L$LUN,R2
 (7)  007414  006302                            ASL     R2
                                                    ; CLEAR ALL BITS IN STATUS EXCEPT DEVICE TYPE
007416                                              LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR!DROPED!ACTIVE!LOBYTE
 (6)  007416  042762  160377  002524                BIC     #ERROR!DROPED!ACTIVE!LOBYTE,STATUS(R2)
007424                                              LET CURCNT(R2) := #-1
 (4)  007424  012762  177777  002764                MOV     #-1,CURCNT(R2)
007432                                              LET ERRTBL(R2) := #0
 (4)  007432  005062  003126                        CLR     ERRTBL(R2)
007436                                              LET DELCNT(R2) := #0
 (4)  007436  005062  003064                        CLR     DELCNT(R2)
007442                                              LET REPCNT(R2) := #0
 (4)  007442  005062  002664                        CLR     REPCNT(R2)
007446                                          ENDINC
 (4)  007446  000753                            BR      50071$
 (3)  007450                            50072$:
007450  004737  007270                          JSR     PC,RESVEC               ; RESET THE VECTORS
007454                                          IF CLKTYP EQ #3 THEN
 (6)  007454  023727  002334  000003            CMP     CLKTYP,#3
 (9)  007462  001006                            BNE     50073$
```

```
                                        CLEANUP CODING SECTION

 1697  007464                                       CLRVEC @CLKVEC
  (3)  007464  017700  172654                       MOV     @CLKVEC,R0
  (3)  007470  104436                               TRAP    C$CVEC
 1698  007472                                       LET @CLKCSR := #00
  (4)  007472  012777  000000  172640               MOV     #00,@CLKCSR
 1699  007500                                       ENDIF
  (4)  007500                                  50073$:
 1700  007500                                       IF CLKTYP EQ #2 THEN
  (6)  007500  023727  002334  000002               CMP     CLKTYP,#2
  (9)  007506  001013                               BNE     50074$
 1701  007510                                       SETVEC CLKVEC,#IGNORE,#PRI06
  (7)  007510  012746  000300                       MOV     #PRI06,-(SP)
  (6)  007514  012746  007266                       MOV     #IGNORE,-(SP)
  (5)  007520  013746  002344                       MOV     CLKVEC,-(SP)
  (4)  007524  012746  000003                       MOV     #3,-(SP)
  (3)  007530  104437                               TRAP    C$SVEC
  (2)  007532  062706  000010                       ADD     #10,SP
 1702  007536                                       ENDIF
  (4)  007536                                  50074$:
 1703  007536                                       SETPRI #PRI00
  (3)  007536  012700  000000                       MOV     #PRI00,R0
  (3)  007542  104441                               TRAP    C$SPRI
 1704  007544                                  ENDCLN
  (3)  007544                                  L10005:
  (3)  007544  104412                               TRAP    C$CLEAN
 1705
 1706  007546                                  ENDMOD
```

```
                                .SBTTL  INTERFACE LOGIC

007546                          BGNMOD
                                ;++
                                ;THIS TEST VERIFIES THE OPERATION OF THE INTERFACE LOGIC. TESTS ARE
                                ;PERFORMED FOR PRINTER ERROR, PRINTER READY, AND CLEARING PRINTER READY
                                ;BY LOADING A CHARACTER INTO THE OUTPUT BUFFER. ALSO IT IS VERIFIED
                                ;THAT THE PRINTER WILL NOT INTERRUPT IF IT IS AT THE SAME PRIORITY LEVEL
                                ;AS THE PROCESSOR, BUT WILL INTERRUPT IF THE PROCESSOR IS AT A LOWER
                                ;PRIORITY LEVEL. THE PRINTER IS AT PRIORITY LEVEL 4.
                                ;
                                ;
                                ;--
007546                          BGNTST  1
007546                          T1::
007546                          LET R1 := L$UNIT - #1                   ;MAX NUMBER OF UNITS ON SYSTEM
007546  013701  002012                  MOV     L$UNIT,R1
007552  005301                          DEC     R1
                                ;
                                ;HARD CODED INCREMEMNT LOOP
                                ;INCR LUNIT FROM #0 TO R1 BY #1          ;START LOOP
                                ;
007554  005037  002326                  CLR     LUNIT                   ;UNIT TO 0
007560  000402                          BR      T1C                     ;DO COMPARE
007562                          T1A:
007562  005237  002326                  INC     LUNIT                   ;UPDATE UNIT NUMBER
007566                          T1C:
007566  023701  002326                  CMP     LUNIT,R1                ;DO COMPARISON OF UNIT NUMBER
007572  003402                          BLE     1$                      ;ONTO NEXT UNIT
007574  000137  010264                  JMP     T1B                     ;EXIT LOOP
007600                          1$:
007600                                  LET R2 := LUNIT SHIFT 1
007600  013702  002326                  MOV     LUNIT,R2
007604  006302                          ASL     R2
007606                              IF #BIT15 SETIN @LPCSR(R2) THEN
007606  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
007614  001416                          BEQ     50075$
007616                                  LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
007616  052762  100000  002524          BIS     #ERROR,STATUS(R2)
007624                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
007624  005262  003126                  INC     ERRTBL(R2)
007630                                  LET L$LUN := LUNIT
007630  013737  002326  002074          MOV     LUNIT,L$LUN
007636                                  ERRHRD 1,CSRERR         ;ERROR BIT WAS SET. SAY SO
007636  104456                          TRAP    C$ERHRD
007640  000001                          .WORD   1
007642  003420                          .WORD   CSRERR
007644  000000                          .WORD   0
007646                                  LET @LPCSR(R2) := #0
007646  005072  002370                  CLR     @LPCSR(R2)
007652                              ENDIF
007652                          50075$:
                                ;TIME DELAY
                                        ; IF NOT READY ALLOW 3 SECONDS TO COME UP
007652                                  IF #BIT7 NOTSETIN @LPCSR(R2) THEN
007652  032772  000200  002370          BIT     #BIT7,@LPCSR(R2)
```

```
  (9)  007660  001014                          BNE     50076$
1747   007662                                      DELAY 30.
  (2)  007662  012727  000036                  MOV     #30.,(PC)+
  (2)  007666  000000                          .WORD   0
  (2)  007670  013727  002116                  MOV     L$DLY,(PC)+
  (2)  007674  000000                          .WORD   0
  (2)  007676  005367  177772                  DEC     -6(PC)
  (2)  007702  001375                          BNE     .-4
  (2)  007704  005367  177756                  DEC     -22(PC)
  (2)  007710  001367                          BNE     .-20
1748   007712                                  ENDIF
  (4)  007712                          50076$:
1749                                   ;
1750                                   ;NOW TEST FOR PRINTER READY
1751                                   ;
1752   007712                              IF #BIT07 NOTSETIN @LPCSR(R2) THEN          ;TEST FOR THE READY BIT
  (6)  007712  032772  000200  002370          BIT     #BIT07,@LPCSR(R2)
  (9)  007720  001014                          BNE     50077$
1753   007722                                  LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
  (6)  007722  052762  100000  002524          BIS     #ERROR,STATUS(R2)
1754   007730                                  LET L$LUN := LUNIT
  (4)  007730  013737  002326  002074          MOV     LUNIT,L$LUN
1755   007736                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  007736  005262  003126                  INC     ERRTBL(R2)
1756   007742                                  ERRHRD 2,RDYERR         ;REPORT AN ERROR
  (4)  007742  104456                          TRAP    C$ERHRD
  (5)  007744  000002                          .WORD   2
  (5)  007746  003436                          .WORD   RDYERR
  (5)  007750  000000                          .WORD   0
1757   007752                              ENDIF
  (4)  007752                          50077$:
1758                                   ;
1759                                   ;INSURE LOADING CHARACTER CAUSES PRINTER READY TO GO AWAY
1760                                   ;
1761   007752                              LET @LPBUF(R2) := #12
  (4)  007752  012772  000012  002464          MOV     #12,@LPBUF(R2)
1762   007760                              IF #BIT07 SETIN @LPCSR(R2) THEN
  (6)  007760  032772  000200  002370          BIT     #BIT07,@LPCSR(R2)
  (9)  007766  001416                          BEQ     50100$
1763   007770                                  LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
  (6)  007770  052762  100000  002524          BIS     #ERROR,STATUS(R2)
1764   007776                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  007776  005262  003126                  INC     ERRTBL(R2)
1765   010002                                  LET L$LUN := LUNIT
  (4)  010002  013737  002326  002074          MOV     LUNIT,L$LUN
1766   010010                                  ERRHRD 3,ERR11          ;REPORT AN ERROR
  (4)  010010  104456                          TRAP    C$ERHRD
  (5)  010012  000003                          .WORD   3
  (5)  010014  010511                          .WORD   ERR11
  (5)  010016  000000                          .WORD   0
1767   010020                                  LET @LPCSR(R2) := #0
  (4)  010020  005072  002370                  CLR     @LPCSR(R2)
1768   010024                              ENDIF
  (4)  010024                          50100$:
1769                                   ;
1770                                   ;VERIFY THAT THE PRINTER WILL NOT INTERRUPT IF IT IS AT A PRIORITY LEVEL
```

```
1771                                    ;THE SAME AS THE CPU
1772                                    ;
1773  010024                                SETPRI #PRI04                     ;CPU TO PRIORITY 4
 (3)  010024  012700  000200                    MOV     #PRI04,R0
 (3)  010030  104441                            TRAP    C$SPRI
1774  010032                                SETVEC LPVEC(R2),#INTERR,#PRI04   ;LP VECTOR SET UP
 (7)  010032  012746  000200                    MOV     #PRI04,-(SP)
 (6)  010036  012746  010422                    MOV     #INTERR,-(SP)
 (5)  010042  016246  002430                    MOV     LPVEC(R2),-(SP)
 (4)  010046  012746  000003                    MOV     #3,-(SP)
 (3)  010052  104437                            TRAP    C$SVEC
 (2)  010054  062706  000010                    ADD     #10,SP
1775  010060                                LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100    ;INTERRUPT ENABLE
 (6)  010060  052772  000100  002370            BIS     #100,@LPCSR(R2)
1776  010066                                    DELAY   30.                ; ALLOW 3 SEC FOR DELAY
 (2)  010066  012727  000036                    MOV     #30.,(PC)+
 (2)  010072  000000                            .WORD   0
 (2)  010074  013727  002116                    MOV     L$DLY,(PC)+
 (2)  010100  000000                            .WORD   0
 (2)  010102  005367  177772                    DEC     -6(PC)
 (2)  010106  001375                            BNE     .-4
 (2)  010110  005367  177756                    DEC     -22(PC)
 (2)  010114  001367                            BNE     .-20
1777                                    ;
1778                                    ;NOW TEST THAT THE PRINTER WILL INTERRUPT IF THE CPU PRIORITY IS LOWER THAN
1779                                    ;THE PRINTER PRIORITY
1780                                    ;
1781  010116                                LET @LPCSR(R2) := @LPCSR(R2) CLR.BY #100    ;CLEAR INTERRUPT ENABLE
 (6)  010116  042772  000100  002370            BIC     #100,@LPCSR(R2)
1782  010124                                SETPRI #PRI03                     ;CPU TO PRIORITY 3
 (3)  010124  012700  000140                    MOV     #PRI03,R0
 (3)  010130  104441                            TRAP    C$SPRI
1783  010132                                SETVEC LPVEC(R2),#INTHDL,#PRI04
 (7)  010132  012746  000200                    MOV     #PRI04,-(SP)
 (6)  010136  012746  010452                    MOV     #INTHDL,-(SP)
 (5)  010142  016246  002430                    MOV     LPVEC(R2),-(SP)
 (4)  010146  012746  000003                    MOV     #3,-(SP)
 (3)  010152  104437                            TRAP    C$SVEC
 (2)  010154  062706  000010                    ADD     #10,SP
1784  010160                                LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100    ;INTERRUPT ENABLE
 (6)  010160  052772  000100  002370            BIS     #100,@LPCSR(R2)
1785  010166                                    DELAY 30                   ; ALLOW 3 SEC DELAY
 (2)  010166  012727  000030                    MOV     #30,(PC)+
 (2)  010172  000000                            .WORD   0
 (2)  010174  013727  002116                    MOV     L$DLY,(PC)+
 (2)  010200  000000                            .WORD   0
 (2)  010202  005367  177772                    DEC     -6(PC)
 (2)  010206  001375                            BNE     .-4
 (2)  010210  005367  177756                    DEC     -22(PC)
 (2)  010214  001367                            BNE     .-20
1786  010216                                    LET ERRTBL(R2) := ERRTBL(R2) + #1
 (6)  010216  005262  003126                    INC     ERRTBL(R2)
1787  010222                                    LET L$LUN := LUNIT
 (4)  010222  013737  002326  002074            MOV     LUNIT,L$LUN
1788  010230                                ERRHRD 4,ERR13
 (4)  010230  104456                            TRAP    C$ERHRD
```

```
  (5)  010232  000004                          .WORD   4
  (5)  010234  010650                          .WORD   ERR13
  (5)  010236  000000                          .WORD   0
 1789  010240                        END2:     LET @LPCSR(R2) := #00              ; CLEAR THE LPCSR
  (4)  010240  012772  000000  002370          MOV     #00,@LPCSR(R2)
 1790  010246                                  LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR!DROPED!ACTIVE
  (6)  010246  042762  160000  002524          BIC     #ERROR!DROPED!ACTIVE,STATUS(R2)
 1791  010254                                  LET DELCNT(R2) := #0
  (4)  010254  005062  003064                  CLR     DELCNT(R2)
 1792                                         ;
 1793                                         ;END OF HARD CODED INCREMENT LOOP
 1794                                         ;ENDINC
 1795                                         ;
 1796  010260  000137  007562                  JMP     T1A                        ;UPDATE UNIT #
 1797  010264  004737  007270          T1B:    JSR     PC,RESVEC          ; RESET STANDARD VECTORS
 1798  010270                                  SETPRI #PRI00
  (3)  010270  012700  000000                  MOV     #PRI00,R0
  (3)  010274  104441                          TRAP    C$SPRI
 1799  010276                                  OUTPUT #INTFAC,#24.
 1800  010340                                  DECR WORK FROM #12. TO #1 BY #1
  (4)  010340  012737  000014  003166          MOV     #12.,WORK
  (5)  010346  000402                          BR      50101$
  (4)  010350                        50102$:
  (7)  010350  005337  003166                  DEC     WORK
  (5)  010354                        50101$:
  (5)  010354  023727  003166  000001          CMP     WORK,#1
  (7)  010362  002415                          BLT     50103$
 1801  010364                                     DELAY 250
  (2)  010364  012727  000250                  MOV     #250,(PC)+
  (2)  010370  000000                          .WORD   0
  (2)  010372  013727  002116                  MOV     L$DLY,(PC)+
  (2)  010376  000000                          .WORD   0
  (2)  010400  005367  177772                  DEC     -6(PC)
  (2)  010404  001375                          BNE     .-4
  (2)  010406  005367  177756                  DEC     -22(PC)
  (2)  010412  001367                          BNE     .-20
 1802  010414                                  ENDDEC
  (4)  010414  000755                          BR      50102$
  (3)  010416                        50103$:
 1803  010416                                  EXIT TST                                   ;EXIT THE TEST
  (3)  010416  104432                          TRAP    C$EXIT
  (3)  010420  000304                          .WORD   L10006-.
 1804                                         ;
 1805                                         ;INTERRUPT HANDLER TO SERVICE FAULTY INTERRUPT FROM LP INTERFACE.
 1806                                         ;THIS ROUTINE IS ENTERED ONLY WHEN THE LP INTERRUPTS AT THE SAME LEVEL AS
 1807                                         ;THE CPU AND IS CONSIDERED AN ERROR.
 1808                                         ;
 1809  010422                                  BGNSRV
 1810  010422                                  INTERR: LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  010422  005262  003126                  INC     ERRTBL(R2)
 1811  010426                                  LET L$LUN := LUNIT
  (4)  010426  013737  002326  002074          MOV     LUNIT,L$LUN
 1812  010434                                  ERRHRD 5,ERR12
  (4)  010434  104456                          TRAP    C$ERHRD
  (5)  010436  000005                          .WORD   5
  (5)  010440  010565                          .WORD   ERR12
```

```
  (5)  010442  000000                        .WORD   0
1813   010444                               LET (SP) := #END2
  (4)  010444  012716  010240                      MOV     #END2,(SP)
1814   010450                               ENDSRV
  (3)  010450                               L10007:
  (2)  010450  000002                              RTI
1815                                       ;
1816                                       ;INTERRUPT HANDLER FOR EXPECTED INTERRUPT
1817                                       ;
1818   010452                              BGNSRV
1819                                       ;
1820   010452                              INTHDL: LET (SP) := #END2
  (4)  010452  012716  010240                      MOV     #END2,(SP)
1821   010456                               ENDSRV
  (3)  010456                               L10010:
  (2)  010456  000002                              RTI
1822                                       ;
1823   010460  047111  042524  043122      INTFAC: .ASCIZ  /INTERFACE LOGIC TEST 1/<12><12>
       010466  041501  020105  047514
       010474  044507  020103  042524
       010502  052123  030440  005012
       010510     000
1824                                       ;
1825                                       ;ERROR MESSAGES ASSOCIATED WITH THIS TEST
1826                                       ;
1827                                       .NLIST BEX
1828   010511     114  040517  044504      ERR11:  .ASCIZ  /LOADING PRINTER BUFFER DOES NOT CLEAR READY/
1829   010565     120  044522  052116      ERR12:  .ASCIZ  /PRINTER INTERRUPTED AT SAME LEVEL AS THE PROCESSOR/
1830   010650  051120  047111  042524      ERR13:  .ASCIZ  /PRINTER DID NOT INTERRUPT AT CPU PRIORITY 3/
1831                                       .EVEN
1832   010724                              ENDTST
  (3)  010724                               L10006:
  (5)  010724  104401                              TRAP    C$ETST
1833                                       .LIST BEX
1834   010726                              ENDMOD
1835
```

```
                                .SBTTL  READY LINE INTERLOCKS   TEST 2

010726                          BGNMOD
                                ;++
                                ;THIS TEST CHECKS THE OPERATION OF THE
                                ;PRINTER READY INTERLOCK SWITCHES.
                                ;MANUAL INTERVENTION IS USED TO
                                ;OPEN THE INTERLOCKS TO PRODUCE FAULTS
                                ;IN THE PRINTER AFTER WHICH THE RESULTANT ERROR
                                ;INDICATION IS VERIFIED.
                                ;--

010726                          BGNTST  2
010726                          T2::
                                ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
010726                                  MANUAL
010726  104450                          TRAP    C$MANI
010730                                  BCOMPLETE 11$
010730  103402                          BCS     11$
010732                                  EXIT TST
010732  104432                          TRAP    C$EXIT
010734  002746                          .WORD   L10011-.
                                ;EXIT TEST IF MANUAL INTERVENTION TESTS ARE NOT SPECIFIED
010736                          11$:    IF INHINT EQ #0 THEN
010736  005737  002272                  TST     INHINT
010742  001002                          BNE     50104$
010744                                  EXIT TST
010744  104432                          TRAP    C$EXIT
010746  002734                          .WORD   L10011-.

010750                          ENDIF
010750                          50104$:
010750                                  LET FLAG := #0
010750  005037  002306                  CLR     FLAG
010754                                  LET R1 := L$UNIT - #1
010754  013701  002012                  MOV     L$UNIT,R1
010760  005301                          DEC     R1

                                ;CHECK FOR ERROR IN EACH PRINTER UNDER TEST
010762                              INCR LUNIT FROM #0 TO R1 BY #1
010762  005037  002326                  CLR     LUNIT
010766  000402                          BR      50105$
010770                          50106$:
010770  005237  002326                  INC     LUNIT
010774                          50105$:
010774  023701  002326                  CMP     LUNIT,R1
011000  003020                          BGT     50107$
011002                                  LET R2 := LUNIT SHIFT 1
011002  013702  002326                  MOV     LUNIT,R2
011006  006302                          ASL     R2
011010                                  IF #BIT15 SETIN @LPCSR(R2) THEN
011010  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
011016  001410                          BEQ     50110$
011020                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
011020  005262  003126                  INC     ERRTBL(R2)
011024                                      ERRHRD 6, CSRERR
```

```
                                        TRAP    C$ERHRD
                                        .WORD   6
                                        .WORD   CSRERR
                                        .WORD   0
1868  011034                                    LET @LPCSR(R2) := #0
 (4)  011034  005072  002370                    CLR     @LPCSR(R2)
1869  011040                                    ENDIF
 (4)  011040                          50110$:
1870  011040                              ENDINC
 (4)  011040  000753                            BR      50106$
 (3)  011042                          50107$:
1871                                  ;CHECK FOR READY IN EACH PRINTER UNDER TEST
1872  011042                              INCR LUNIT FROM #0 TO R1 BY #1
 (4)  011042  005037  002326                    CLR     LUNIT
 (5)  011046  000402                            BR      50111$
 (4)  011050                          50112$:
 (7)  011050  005237  002326                    INC     LUNIT
 (5)  011054                          50111$:
 (5)  011054  023701  002326                    CMP     LUNIT,R1
 (7)  011060  003021                            BGT     50113$
1873  011062                                    LET R2 := LUNIT SHIFT 1
 (4)  011062  013702  002326                    MOV     LUNIT,R2
 (7)  011066  006302                            ASL     R2
1874  011070                                    LET L$LUN := LUNIT
 (4)  011070  013737  002326  002074            MOV     LUNIT,L$LUN
1875  011076                                    IF #BIT07 NOTSETIN @LPCSR(R2) THEN
 (6)  011076  032772  000200  002370            BIT     #BIT07,@LPCSR(R2)
 (9)  011104  001006                            BNE     50114$
1876  011106                                    LET ERRTBL(R2) := ERRTBL(R2) + #1
 (6)  011106  005262  003126                    INC     ERRTBL(R2)
1877  011112                                        ERRHRD 7, RDYERR
 (4)  011112  104456                            TRAP    C$ERHRD
 (5)  011114  000007                            .WORD   7
 (5)  011116  003436                            .WORD   RDYERR
 (5)  011120  000000                            .WORD   0
1878  011122                                    ENDIF
 (4)  011122                          50114$:
1879  011122                              ENDINC
 (4)  011122  000752                            BR      50112$
 (3)  011124                          50113$:
1880                                  ;
1881                                  ; PRINT TEST NAME
1882                                  ;
1883  011124                              OUTPUT #INTLK,#29.
1884                                  ;VERIFY OPERATION OF PAPER LOW INTERLOCK SWITCH
1885                                  ;HARD CODED INCREMENT LOOP
1886                                  ;
1887  011166                                  LET ERRFLG := #0
 (4)  011166  005037  002352                    CLR     ERRFLG
1888  011172  005037  002326              CLR LUNIT
1889  011176  000405                      BR  1$
1890  011200                              2$:
1891  011200  005237  002326                INC LUNIT
1892  011204                                  LET R2 := LUNIT SHIFT 1
 (4)  011204  013702  002326                    MOV     LUNIT,R2
 (7)  011210  006302                            ASL     R2
```

```
 1893  011212                          1$:
 1894  011212  023701  002326              CMP LUNIT,R1
 1895  011216  003402                      BLE 3$
 1896  011220  000137  011560              JMP 4$
 1897  011224                          3$:
 1898  011224                                LET FLAG := #0
  (4)  011224  005037  002306                CLR     FLAG
 1899  011230                                PRINTF #PAPRSW,LUNIT
  (8)  011230  013746  002326                MOV     LUNIT,-(SP)
  (7)  011234  012746  012256                MOV     #PAPRSW,-(SP)
  (6)  011240  012746  000002                MOV     #2,-(SP)
  (3)  011244  010600                        MOV     SP,R0
  (4)  011246  104417                        TRAP    C$PNTF
  (4)  011250  062706  000006                ADD     #6,SP
 1900  011254                                PRINTF #PAPSW1
  (7)  011254  012746  012326                MOV     #PAPSW1,-(SP)
  (6)  011260  012746  000001                MOV     #1,-(SP)
  (3)  011264  010600                        MOV     SP,R0
  (4)  011266  104417                        TRAP    C$PNTF
  (4)  011270  062706  000004                ADD     #4,SP
 1901  011274                                GMANIL READY, FLAG, 100000, YES
  (3)  011274  104443                        TRAP    C$GMAN
  (3)  011276  000404                        BR      10000$
  (4)  011300  002306                        .WORD   FLAG
  (5)  011302  000130                        .WORD   T$CODE
  (5)  011304  007072                        .WORD   READY
  (5)  011306  100000                        .WORD   100000
  (3)  011310                           10000$:
 1902  011310                                LET LINCNT := #400.     ; ALLOW FOR ABOUT 6 PAGES OF PAPER
  (4)  011310  012737  000620  002310        MOV     #400.,LINCNT
 1903  011316                                LET ERRFLG := #0
  (4)  011316  005037  002352                CLR     ERRFLG
 1904  011322                                REPEAT
  (3)  011322                           50115$:
 1905  011322                                   OUTPUI #PAPTST,#25.,#5$,LUNIT
 1906  011364                                   LET LINCNT := LINCNT - #1
  (6)  011364  005337  002310                DEC     LINCNT
 1907  011370                                UNTIL LINCNT EQ #0 OR ERRFLG NE #0
  (4)  011370  005737  002310                TST     LINCNT
  (6)  011374  001403                        BEQ     50116$
  (4)  011376  005737  002352                TST     ERRFLG
  (7)  011402  001747                        BEQ     50115$
  (4)  011404                           50116$:
 1908  011404                                IF ERRFLG EQ #0 THEN
  (6)  011404  005737  002352                TST     ERRFLG
  (9)  011410  001011                        BNE     50117$
 1909  011412                                   ERRHRD 8,PAPSWI
  (4)  011412  104456                        TRAP    C$ERHRD
  (5)  011414  000010                        .WORD   8
  (5)  011416  003460                        .WORD   PAPSWI
  (5)  011420  000000                        .WORD   0
 1910  011422                                   LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  011422  005262  003126                INC     ERRTBL(R2)
 1911  011426                                   INLINE      <JMP    11002$>
  (2)  011426  000137  011440                JMP     11002$
 1912  011432                                ELSE
```

```
  (4)  011432  000402                        BR      50120$
  (3)  011434                          50117$:
1913   011434                                  LET ERRFLG := #0
  (4)  011434  005037  002352                CLR     ERRFLG
1914   011440                                  ENDIF
  (4)  011440                          50120$:
1915   011440                          11002$: PRINTF #PAPRDY,LUNIT
  (8)  011440  013746  002326                MOV     LUNIT,-(SP)
  (7)  011444  012746  012404                MOV     #PAPRDY,-(SP)
  (6)  011450  012746  000002                MOV     #2,-(SP)
  (3)  011454  010600                        MOV     SP,R0
  (4)  011456  104417                        TRAP    C$PNTF
  (4)  011460  062706  000006                ADD     #6,SP
1916   011464                                  LET FLAG := #0
  (4)  011464  005037  002306                CLR     FLAG
1917   011470                                  GMANIL READY,FLAG,100000,YES
  (3)  011470  104443                        TRAP    C$GMAN
  (3)  011472  000404                        BR      10001$
  (4)  011474  002306                        .WORD   FLAG
  (5)  011476  000130                        .WORD   T$CODE
  (5)  011500  007072                        .WORD   READY
  (5)  011502  100000                        .WORD   100000
  (3)  011504                          10001$:
1918   011504                                  LET R2 := LUNIT SHIFT 1
  (4)  011504  013702  002326                MOV     LUNIT,R2
  (7)  011510  006302                        ASL     R2
1919   011512                                  LET @LPCSR(R2) := #0            ; RESET THE LP CSR
  (4)  011512  005072  002370                CLR     @LPCSR(R2)
1920   011516  000137  011200                JMP 2$
1921                                   ;EXPECTED ERROR HANDLER.
1922                                   ;JUST SET EXPECTED ERROR INDICATOR.
1923                                   ;
1924   011522                          5$:     LET ERRFLG := #1
  (4)  011522  012737  000001  002352        MOV     #1,ERRFLG
1925   011530                                  LET ERRCOD := #0
  (4)  011530  005037  002350                CLR     ERRCOD
1926   011534                                  LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR!ACTIVE
  (6)  011534  042762  120000  002524        BIC     #ERROR!ACTIVE,STATUS(R2)
1927   011542                                  LET CURCNT(R2) := #0    ; CLEAN UP THE DRIVER PARAMETERS
  (4)  011542  005062  002764                CLR     CURCNT(R2)
1928   011546                                  LET CURADD(R2) := #0
  (4)  011546  005062  002564                CLR     CURADD(R2)
1929   011552                                  LET REPCNT(R2) := #0
  (4)  011552  005062  002664                CLR     REPCNT(R2)
1930   011556  000207                        RTS PC                          ;AND RETURN
1931                                   ;VERIFY OPERATION OF HAMMER BANK INTERLOCK SWITCH
1932   011560                          4$:    INCR LUNIT FROM #0 TO R1 BY #1
  (4)  011560  005037  002326                CLR     LUNIT
  (5)  011564  000402                        BR      50121$
  (4)  011566                          50122$:
  (7)  011566  005237  002326                INC     LUNIT
  (5)  011572                          50121$:
  (5)  011572  023701  002326                CMP     LUNIT,R1
  (7)  011576  003077                        BGT     50123$
1933   011600                                  LET R2 := LUNIT SHIFT 1
  (4)  011600  013702  002326                MOV     LUNIT,R2
```

```
  (7)  011604  006302                          ASL     R2
1934   011606                                  LET L$LUN := LUNIT
  (4)  011606  013737  002326  002074          MOV     LUNIT,L$LUN
1935   011614                                  LET FLAG := #0
  (4)  011614  005037  002306                  CLR     FLAG
1936   011620                                  PRINTF #HAMRSW,LUNIT
  (8)  011620  013746  002326                  MOV     LUNIT,-(SP)
  (7)  011624  012746  013147                  MOV     #HAMRSW,-(SP)
  (6)  011630  012746  000002                  MOV     #2,-(SP)
  (3)  011634  010600                          MOV     SP,R0
  (4)  011636  104417                          TRAP    C$PNTF
  (4)  011640  062706  000006                  ADD     #6,SP
1937   011644                                  PRINTF #HAMSW1
  (7)  011644  012746  013233                  MOV     #HAMSW1,-(SP)
  (6)  011650  012746  000001                  MOV     #1,-(SP)
  (3)  011654  010600                          MOV     SP,R0
  (4)  011656  104417                          TRAP    C$PNTF
  (4)  011660  062706  000004                  ADD     #4,SP
1938   011664                                  GMANIL READY, FLAG, 100000, YES
  (3)  011664  104443                          TRAP    C$GMAN
  (3)  011666  000404                          BR      10002$
  (4)  011670  002306                          .WORD   FLAG
  (5)  011672  000130                          .WORD   T$CODE
  (5)  011674  007072                          .WORD   READY
  (5)  011676  100000                          .WORD   100000
  (3)  011700                          10002$:
1939   011700                                  IF #BIT15 SETIN @LPCSR(R2) THEN
  (6)  011700  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
  (9)  011706  001421                          BEQ     50124$
1940   011710                                  PRINTF #HAMRDY,LUNIT
  (8)  011710  013746  002326                  MOV     LUNIT,-(SP)
  (7)  011714  012746  013310                  MOV     #HAMRDY,-(SP)
  (6)  011720  012746  000002                  MOV     #2,-(SP)
  (3)  011724  010600                          MOV     SP,R0
  (4)  011726  104417                          TRAP    C$PNTF
  (4)  011730  062706  000006                  ADD     #6,SP
1941   011734                                  GMANIL READY, FLAG, 100000, YES
  (3)  011734  104443                          TRAP    C$GMAN
  (3)  011736  000404                          BR      10003$
  (4)  011740  002306                          .WORD   FLAG
  (5)  011742  000130                          .WORD   T$CODE
  (5)  011744  007072                          .WORD   READY
  (5)  011746  100000                          .WORD   100000
  (3)  011750                          10003$:
1942   011750                                  ELSE
  (4)  011750  000411                          BR      50125$
  (3)  011752                          50124$:
1943   011752                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  011752  005262  003126                  INC     ERRTBL(R2)
1944   011756                                  LET L$LUN := LUNIT
  (4)  011756  013737  002326  002074          MOV     LUNIT,L$LUN
1945
1946   011764                                      ERRHRD 9, BNKSWI
  (4)  011764  104456                          TRAP    C$ERHRD
  (5)  011766  000011                          .WORD   9
  (5)  011770  003523                          .WORD   BNKSWI
```

```
  (5)  011772  000000                        .WORD   0
1947   011774                               ENDIF
  (4)  011774                          50125$:
1948   011774                              ENDINC
  (4)  011774  000674                       BR      50122$
  (3)  011776                          50123$:
1949                                   ;VERIFY OPERATION OF CHARACTER BAND INTERLOCK SWITCH
1950   011776                              INCR LUNIT FROM #0 TO R1 BY #1
  (4)  011776  005037  002326               CLR     LUNIT
  (5)  012002  000402                       BR      50126$
  (4)  012004                          50127$:
  (7)  012004  005237  002326               INC     LUNIT
  (5)  012010                          50126$:
  (5)  012010  023701  002326               CMP     LUNIT,R1
  (7)  012014  003077                       BGT     50130$
1951   012016                              LET R2 := LUNIT SHIFT 1
  (4)  012016  013702  002326               MOV     LUNIT,R2
  (7)  012022  006302                       ASL     R2
1952   012024                              LET FLAG := #0
  (4)  012024  005037  002306               CLR     FLAG
1953   012030                              PRINTF #BANDSW,LUNIT
  (8)  012030  013746  002326               MOV     LUNIT,-(SP)
  (7)  012034  012746  013425               MOV     #BANDSW,-(SP)
  (6)  012040  012746  000002               MOV     #2,-(SP)
  (3)  012044  010600                       MOV     SP,R0
  (4)  012046  104417                       TRAP    C$PNTF
  (4)  012050  062706  000006               ADD     #6,SP
1954   012054                              PRINTF #BNDSW1
  (7)  012054  012746  013513               MOV     #BNDSW1,-(SP)
  (6)  012060  012746  000001               MOV     #1,-(SP)
  (3)  012064  010600                       MOV     SP,R0
  (4)  012066  104417                       TRAP    C$PNTF
  (4)  012070  062706  000004               ADD     #4,SP
1955   012074                              GMANIL READY, FLAG, 100000, YES
  (3)  012074  104443                       TRAP    C$GMAN
  (3)  012076  000404                       BR      10004$
  (4)  012100  002306                       .WORD   FLAG
  (5)  012102  000130                       .WORD   T$CODE
  (5)  012104  007072                       .WORD   READY
  (5)  012106  100000                       .WORD   100000
  (3)  012110                          10004$:
1956   012110                              IF #BIT15 SETIN @LPCSR(R2) THEN
  (6)  012110  032772  100000  002370       BIT     #BIT15,@LPCSR(R2)
  (9)  012116  001421                       BEQ     50131$
1957   012120                                  PRINTF #BNDRDY,LUNIT
  (8)  012120  013746  002326               MOV     LUNIT,-(SP)
  (7)  012124  012746  013562               MOV     #BNDRDY,-(SP)
  (6)  012130  012746  000002               MOV     #2,-(SP)
  (3)  012134  010600                       MOV     SP,R0
  (4)  012136  104417                       TRAP    C$PNTF
  (4)  012140  062706  000006               ADD     #6,SP
1958   012144                                  GMANIL READY, FLAG, 100000, YES
  (3)  012144  104443                       TRAP    C$GMAN
  (3)  012146  000404                       BR      10005$
  (4)  012150  002306                       .WORD   FLAG
  (5)  012152  000130                       .WORD   T$CODE
```

```
                                        READY LINE INTERLOCKS   TEST 2

  (5)  012154  007072                                .WORD   READY
  (5)  012156  100000                                .WORD   100000
  (3)  012160                           10005$:
1959   012160                                       ELSE
  (4)  012160  000411                                BR      50132$
  (3)  012162                           50131$:
1960   012162                                       LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  012162  005262  003126                        INC     ERRTBL(R2)
1961   012166                                       LET L$LUN := LUNIT
  (4)  012166  013737  002326  002074                MOV     LUNIT,L$LUN
1962   012174                                           ERRHRD 10, BNDSWI
  (4)  012174  104456                                TRAP    C$ERHRD
  (5)  012176  000012                                .WORD   10
  (5)  012200  003570                                .WORD   BNDSWI
  (5)  012202  000000                                .WORD   0
1963   012204                                       ENDIF
  (4)  012204                           50132$:
1964   012204                                       LET @LPCSR(R2) := #00
  (4)  012204  012772  000000  002370                MOV     #00,@LPCSR(R2)
1965   012212                                   ENDINC
  (4)  012212  000674                                BR      50127$
  (3)  012214                           50130$:
1966   012214                                   EXIT TST
  (3)  012214  104432                                TRAP    C$EXIT
  (3)  012216  001464                                .WORD   L10011-.
1967
1968                                    .NLIST BEX
1969
1970   012220  042522  042101  020131  INTLK:  .ASCIZ  /READY LINE INTERLOCK TEST 2/<12><12>
1971   012256  047045  040445  042524  PAPRSW: .ASCIZ /%N%ATEAR OFF PAPER JUST BELOW LUNIT %D2/
1972   012326  040445  052040  020117  PAPSW1: .ASCIZ  /%A TO CHECK PAPER LOW %N%AINTERLOCK SWITCH.%N/
1973   012404  047045  040445  042522  PAPRDY: .ASCIZ  /%N%ARESTORE PAPER, ALARM CLEAR,  AND PLACE LUNIT %D2%A ON LINE.%N/
1974   012506  040520  042520  020122  PAPTST: .ASCIZ  /PAPER LOW INTERLOCK TEST/<12>
1975   012540  054130  054130  054130  PAPOUT: .ASCIZ  /XXXXXXXXXXXXXXXXXXXXXXX/<12>
1976   012572  047045  040445  051120  PRINT4: .ASCII  /%N%APRESS ON-LINE SW OF UNIT %D2%A 4 TIMES /
1977   012645     054  030440  046040          .ASCIZ  /, 1 LINE SHOULD BE PRINTED AFTER EACH PRESS%N/
1978   012723     045  022516  050101  DOTOF:  .ASCIZ  /%N%APRESS ON-LINE SW OF UNIT %D2%A FORM FEED SHOULD OCCUR%N/
1979   013017     045  022516  052101  LN3EX:  .ASCII  /%N%ATHE LAST 3 LINES PRINTED SHOULD BE X'S%N/
1980   013073     045  050101  042522          .ASCIZ  /%APRESSING ON-LINE AGAIN SHOULD NOT PRINT%N/
1981   013147     045  022516  042101  HAMRSW: .ASCIZ  /%N%ADISENGAGE HAMMER BANK LATCH SWITCH ON LUNIT %D2/
1982   013233     045  022516  052101  HAMSW1: .ASCIZ  /%N%ATO CHECK HAMMER BANK INTERLOCK SWITCH.%N/
1983   013310  047045  040445  047105  HAMRDY: .ASCIZ  /%N%AENGAGE HAMMER BANK LATCH, ALARM CLEAR,  AND PLACE LUNIT %D2%A ON LI
1984   013425     045  022516  047501  BANDSW: .ASCIZ  /%N%AOPEN CHARACTER BAND COVER ON LUNIT %D2%A TO CHECK/
1985   013513     045  022516  041501  BNDSW1: .ASCIZ  /%N%ACHARACTER BAND INTERLOCK SWITCH.%N/
1986   013562  047045  040445  046103  BNDRDY: .ASCIZ/%N%ACLOSE CHARACTER BAND COVER ON LUNIT %D2%A, ALARM CLEAR,  AND PLACE ON
1987                                    .EVEN
1988
1989                                    .LIST BEX
1990   013702                           ENDTST
  (3)  013702                           L10011:
  (3)  013702  104401                        TRAP    C$ETST
1991
1992   013704                           ENDMOD
1993
```

```
 1995                                   .SBTTL FORMS LENGTH SELECTION
 1996  013704                           BGNMOD
 1997                                   ;++
 1998                                   ;THIS TEST CHECKS ALL POSITIONS OF THE FORM LENGTH SELECT SWITCH. THE
 1999                                   ;PROGRAM INDICATES THE SPECIFIED SETTING OF THE FORM LENGTH SELECT SWITCH
 2000                                   ;AND WAITS FOR THE OPERATOR TO SET THE SWITCH ON THE PRINTER. THE PAPER
 2001                                   ;IS THEN ADVANCED UNDER PROGRAM CONTROL. THE PRINTER OUTPUT IS VISUALLY
 2002                                   ;INSPECTED AFTER ALL SWITCH SETTINGS HAVE BEEN RUN THROUGH BY THE OPERATOR
 2003                                   ;TO VERIFY THAT THE PROPER PAPER MOVEMENT HAS OCCURRED FOR EACH SWITCH
 2004                                   ;SETTING.
 2005                                   ;--
 2006  013704                           BGNTST  3
  (3)  013704                           T3::
 2007                                   ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
 2008  013704                                   MANUAL
  (3)  013704  104450                     TRAP    C$MANI
 2009  013706                                   BCOMPLETE 1$
  (2)  013706  103402                     BCS     1$
 2010  013710                                   EXIT TST
  (3)  013710  104432                     TRAP    C$EXIT
  (3)  013712  002246                     .WORD   L10012-.
 2011                                   ;EXIT TEST IF MANUAL INTERVENTION TESTS ARE NOT SPECIFIED
 2012  013714  005737  002272           1$:     TST INHINT
 2013  013720  001002                           BNE 2$
 2014  013722                                   EXIT TST
  (3)  013722  104432                     TRAP    C$EXIT
  (3)  013724  002234                     .WORD   L10012-.
 2015                                   ;PRINT TEST IDENTIFICATION
 2016  013726                           2$:     OUTPUT  #FRMLTH,#32.
 2017                                   ;
 2018                                   ;HARD CODE INCREMENT LOOP
 2019                                   ;
 2020  013770  005037  002326                   CLR     LUNIT
 2021  013774  000402                           BR      4$                      ;COMPARE LOOP
 2022  013776  005237  002326           5$:     INC     LUNIT
 2023  014002  023701  002326           4$:     CMP     LUNIT,R1
 2024  014006  003402                           BLE     6$                      ;EXIT ONLY IF GREATER THAN
 2025  014010  000137  014566                   JMP     7$                      ;EXIT
 2026  014014                           6$:
 2027  014014                                   LET R2 := LUNIT SHIFT 1
  (4)  014014  013702  002326                   MOV     LUNIT,R2
  (7)  014020  006302                           ASL     R2
 2028  014022                               PRINTF #LINSWI,LUNIT                ;PRINT LUNIT MESSAGE
  (8)  014022  013746  002326                   MOV     LUNIT,-(SP)
  (7)  014026  012746  014637                   MOV     #LINSWI,-(SP)
  (6)  014032  012746  000002                   MOV     #2,-(SP)
  (3)  014036  010600                           MOV     SP,R0
  (4)  014040  104417                           TRAP    C$PNTF
  (4)  014042  062706  000006                   ADD     #6,SP
 2029  014046                               PRINTF #LINSW1                      ;SECOND PART OF MESSAGE
  (7)  014046  012746  014723                   MOV     #LINSW1,-(SP)
  (6)  014052  012746  000001                   MOV     #1,-(SP)
  (3)  014056  010600                           MOV     SP,R0
  (4)  014060  104417                           TRAP    C$PNTF
  (4)  014062  062706  000004                   ADD     #4,SP
 2030  014066                               PRINTF #FLSSEL,LUNIT                ;SET TO "FLS" POSITION
```

```
   (8)  014066  013746  002326                MOV     LUNIT,-(SP)
   (7)  014072  012746  015003                MOV     #FLSSEL,-(SP)
   (6)  014076  012746  000002                MOV     #2,-(SP)
   (3)  014102  010600                        MOV     SP,R0
   (4)  014104  104417                        TRAP    C$PNTF
   (4)  014106  062706  000006                ADD     #6,SP
 2031   014112                             INCR R4 FROM #0 TO #20. BY #2
   (4)  014112  005004                        CLR     R4
   (5)  014114  000402                        BR      50133$
   (4)  014116                         50134$:
   (7)  014116  062704  000002                ADD     #2,R4
   (5)  014122                         50133$:
   (5)  014122  020427  000024                CMP     R4,#20.
   (7)  014126  003137                        BGT     50135$
 2032   014130                                LET R3 := R4                    ;INDEX INTO SWITCH SETTING TABLE
   (4)  014130  010403                        MOV     R4,R3
 2033   014132  006303                        ASL R3                          ;ACTUAL OFFSET FOR SWITCH SETTINGS
 2034   014134                                LET T3SET := #FFSET + R3
   (4)  014134  012737  015474  014572        MOV     #FFSET,T3SET
   (6)  014142  060337  014572                ADD     R3,T3SET
 2035   014146                                PRINTF #FLSMSG,LUNIT,T3SET                      ;SELECT MESSAGE
   (9)  014146  013746  014572                MOV     T3SET,-(SP)
   (8)  014152  013746  002326                MOV     LUNIT,-(SP)
   (7)  014156  012746  015550                MOV     #FLSMSG,-(SP)
   (6)  014162  012746  000003                MOV     #3,-(SP)
   (3)  014166  010600                        MOV     SP,R0
   (4)  014170  104417                        TRAP    C$PNTF
   (4)  014172  062706  000010                ADD     #10,SP
 2036   014176                                PRINTF #FLSMS1
   (7)  014176  012746  015642                MOV     #FLSMS1,-(SP)
   (6)  014202  012746  000001                MOV     #1,-(SP)
   (3)  014206  010600                        MOV     SP,R0
   (4)  014210  104417                        TRAP    C$PNTF
   (4)  014212  062706  000004                ADD     #4,SP
 2037   014216                                LET FLAG := #0                  ;CLEAR FLAG INDICATOR FOR MANUAL
   (4)  014216  005037  002306                CLR     FLAG
 2038   014222                                GMANIL READY,FLAG,100000,YES    ;WAIT FOR OPERATOR
   (3)  014222  104443                        TRAP    C$GMAN
   (3)  014224  000404                        BR      10000$
   (4)  014226  002306                        .WORD   FLAG
   (5)  014230  000130                        .WORD   T$CODE
   (5)  014232  007072                        .WORD   READY
   (5)  014234  100000                        .WORD   100000
   (3)  014236                         10000$:
 2039   014236                                OUTPUI #REFLIN,#133.,,LUNIT     ;OUTPUT REFERENCE LINE AND TERMINATOR
 2040   014300                                OUTPUI T3SET,#3.,,LUNIT  ;# OF LINES FOR SPACING
 2041   014342                                OUTPUI #MOVMSG,#130.,,LUNIT     ;FINAL REFERENCE LINE
 2042   014404                                LET FLAG := #0
   (4)  014404  005037  002306                CLR     FLAG
 2043   014410                                GMANIL READY,FLAG,100000,YES
   (3)  014410  104443                        TRAP    C$GMAN
   (3)  014412  000404                        BR      10001$
   (4)  014414  002306                        .WORD   FLAG
   (5)  014416  000130                        .WORD   T$CODE
   (5)  014420  007072                        .WORD   READY
   (5)  014422  100000                        .WORD   100000
```

```
                                        FORMS LENGTH SELECTION

  (3)  014424                           10001$:
2044   014424                               ENDINC
  (4)  014424  000634                          BR      50134$
  (3)  014426                           50135$:
2045                                    ;SET FORMS LENGTH SELECT SWITCH TO ITS "REGULAR" SETTING
2046   014426                                 PRINTF #NMLFLS,LUNIT
  (8)  014426  013746  002326                 MOV     LUNIT,-(SP)
  (7)  014432  012746  015402                 MOV     #NMLFLS,-(SP)
  (6)  014436  012746  000002                 MOV     #2,-(SP)
  (3)  014442  010600                         MOV     SP,R0
  (4)  014444  104417                         TRAP    C$PNTF
  (4)  014446  062706  000006                 ADD     #6,SP
2047   014452                                 LET OUTBUF := #14
  (4)  014452  012737  000014  003172         MOV     #14,OUTBUF
2048   014460                                 LET FLAG := #0                  ;CLEAR <CR> FLAG
  (4)  014460  005037  002306                 CLR     FLAG
2049   014464                                 GMANIL READY,FLAG,100000,YES    ;AND WAIT FOR RESPONSE
  (3)  014464  104443                         TRAP    C$GMAN
  (3)  014466  000404                         BR      10002$
  (4)  014470  002306                         .WORD   FLAG
  (5)  014472  000130                         .WORD   T$CODE
  (5)  014474  007072                         .WORD   READY
  (5)  014476  100000                         .WORD   100000
  (3)  014500                           10002$:
2050   014500                                 OUTPUI #OUTBUF,#1,,LUNIT
2051   014542                                 PRINTF #PAPCHK                  ;MAKE SURE MOVEMENT WAS RIGHT
  (7)  014542  012746  015102                 MOV     #PAPCHK,-(SP)
  (6)  014546  012746  000001                 MOV     #1,-(SP)
  (3)  014552  010600                         MOV     SP,R0
  (4)  014554  104417                         TRAP    C$PNTF
  (4)  014556  062706  000004                 ADD     #4,SP
2052   014562  000137  013776                 JMP     5$                      ;END OF HARDCODED INCREMENT LOOP
2053   014566                           7$:
2054   014566                           EXIT TST
  (3)  014566  104432                         TRAP    C$EXIT
  (3)  014570  001370                         .WORD   L10012-.
2055   014572  000000                 T3SET:  .WORD   0
2056   014574  000000                 T3MOV:  .WORD   0
2057                                    .NLIST BEX
2058   014576  047506  046522  020123 FRMLTH: .ASCIZ /FORMS LENGTH SELECTION TEST 3/<12><12><12>
2059   014637     045  022516  051501 LINSWI: .ASCIZ /%N%ASET LINES SWITCH ON UNIT %D2%A TO "6" TO SELECT/
2060   014723     045  022516  033101 LINSW1: .ASCIZ /%N%A6 LINES PER INCH VERTICAL PRINTING DENSITY./
2061   015003     045  031116  040445 FLSSEL: .ASCIZ /%N2%ASET VFU-FLS SWITCH ON UNIT %D2%A TO THE "FLS" POSITION.%N/
2062   015102  047045  040445  042526 PAPCHK: .ASCIZ /%N%AVERIFY PROPER PAPER MOVEMENT FOR EACH SWITCH SETTING./
2063   015174  042522  042506  042522 REFLIN: .ASCII /REFERENCE LINE FOR FORMS LENGTH SELECTION...................../
2064   015271     056  027056  027056 REFLI1: .ASCII /.............................................................../
2065   015366  027056  027056  027056 REFLI2: .ASCIZ /........../<14>
2066   015402  047045  040445  042523 NMLFLS: .ASCIZ /%N%ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO 11.%N/
2067                                    ;SWITCH SETTINGS FOR FORMS LENGTH MESSAGES
2068   015474  020063  000040         FFSET:  .ASCIZ /3  /
2069   015500  027063  000065                 .ASCIZ /3.5/
2070   015504  020064  000040                 .ASCIZ /4  /
2071   015510  027065  000065                 .ASCIZ /5.5/
2072   015514  020066  000040                 .ASCIZ /6  /
2073   015520  020067  000040                 .ASCIZ /7  /
2074   015524  020070  000040                 .ASCIZ /8  /
```

```
 2075  015530  027070  000065                    .ASCIZ /8.5/
 2076  015534  030461  000040                    .ASCIZ /11 /
 2077  015540  031061  000040                    .ASCIZ /12 /
 2078  015544  032061  000040                    .ASCIZ /14 /
 2079  015550  047045  040445  042523  FLSMSG:   .ASCIZ /%N%ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO %T%A,/
 2080  015642  047045  040445  042504  FLSMS1:   .ASCIZ /%N%ADEPRESS "ALARM CLEAR", TOP OF FORM, AND PLACE PRINTER BACK ON LINE.%
 2081  015754  044440  041516  042510  MOVMSG:   .ASCII / INCHES SHOULD OCCUR BETWEEN THIS AND THE REFERENCE LINE...../
 2082  016051     056  027056  027056  MOVMS1:   .ASCII /...................................................................../
 2083  016147     056  027056  027056  MOVMS2:   .ASCIZ /....../<12>
 2084        016160                              .EVEN
 2085                                  .LIST BEX
 2086  016160                          ENDTST
  (3)  016160                          L10012:
  (3)  016160  104401                             TRAP    C$ETST
 2087  016162                          ENDMOD
 2088
```

```
                                .SBTTL PRINTING SPEED MEASUREMENT
016162                          BGNMOD
                                ;++
                                ; THE PRINT SPEED TEST WILL REPORT TO THE OPERATOR THE TOTAL NUMBER OF
                                ; LINES PRINTED WITHIN A SPECIFIED TIME PERIOD.  THE DATA PATTERN USED
                                ; IS DESIGNED TO CAUSE PRINTING SPEED TO BE MINIMAL AND IS DEPENDENT
                                ; ON PRINTER TYPE AND THE CHARACTER SET (BAND TYPE ) ON EACH PRINTER.
                                ; THE TIME PERIOD CAN BE CONTROLLED THRU MANUAL OPERATION, OR IF THE
                                ; SYSTEM HAS A CLOCK VIA SUPPLYING A COUNT OF SECONDS. ANY TIME INTERVAL OF
                                ; 4 TO 60 SECONDS MAY BE SELECTED. THIS IS ONE OF THE "SW" QUESTIONS .
                                ;--
016162                          BGNTST  4
016162                          T4::
                                ;
016162                          LET R1 := L$UNIT - #1                   ;NUMBER OF UNITS TO TEST
016162  013701  002012                  MOV     L$UNIT,R1
016166  005301                        DEC     R1
016170                          IF MANSPD NE #0 THEN            ; DETERMIN IF MANUAL TESTING SELECTED
016170  005737  002276                  TST     MANSPD
016174  001416                        BEQ     50136$
016176                              MANUAL                              ;DETERMINE IF MANUAL INTERVENTION ALLOWED
016176  104450                            TRAP    C$MANI
016200                              BCOMPLETE 1$
016200  103402                            BCS     1$
016202                              EXIT TST
016202  104432                            TRAP    C$EXIT
016204  004122                            .WORD   L10013-.
016206                          1$:     IF INHINT NE #0 THEN    ; EXIT IF INTERVENTION INHIBITED
016206  005737  002272                  TST     INHINT
016212  001403                        BEQ     50137$
016214                                  EXIT TST
016214  104432                        TRAP    C$EXIT
016216  004110                        .WORD   L10013-.
016220                                ELSE
016220  000403                        BR      50140$
016222                          50137$:
016222                                  LET WORK := PERIOD
016222  013737  002300  003166        MOV     PERIOD,WORK
016230                                ENDIF
016230                          50140$:
016230                          ELSE
016230  000403                        BR      50141$
016232                          50136$:
016232                              LET WORK := PERIOD                  ; CLOCK TEST  TIME
016232  013737  002300  003166        MOV     PERIOD,WORK
016240                          ENDIF
016240                          50141$:
016240                          OUTPUT  #PRTSPD,#36.            ;PRINT TEST ID
016302                          SELECT CLKTYP OF 4 VERIFY        ;SET UP THE RIGHT CLOCK
016302  013746  002334                MOV     CLKTYP,-(SP)
016306  003403                      BLE     50142$
016310  021627  000004                CMP     (SP),#4
016314  003402                      BLE     50143$
016316                          50142$:
016316  012716  000005                MOV     #5,(SP)
016322                          50143$:
```

```
  (2)  016322  006316                        ASL     (SP)
  (2)  016324  060716                        ADD     PC,(SP)
  (2)  016326  063607                        ADD     @(SP)+,PC
  (3)  016330                           50144$:
  (4)  016330  000012                        .WORD   50151$-50144$
  (4)  016332  000020                        .WORD   50150$-50144$
  (4)  016334  000072                        .WORD   50147$-50144$
  (4)  016336  000160                        .WORD   50146$-50144$
  (3)  016340  000162                        .WORD   50145$-50144$
2118
2119  016342                           CASE 1
  (4)  016342                           50151$:
2120  016342  000137  017514                JMP     END4            ;JUST EXIT TEST NO CLOCK AVAILBLE
2121
2122  016346                           CASE 2                  ;KW11-L LINE CLOCK SELECTED
  (4)  016346  000461                        BR      50145$
  (4)  016350                           50150$:
2123  016350                                LET CLKENA := #100        ;INTERRUPT ENABLE/ CLR MONITOR
  (4)  016350  012737  000100  002346        MOV     #100,CLKENA
2124                                        ; SET PRI7 WHILE CHANGING VECTOR ADDRESS
2125  016356                                SETPRI #PRI07
  (3)  016356  012700  000340               MOV     #PRI07,R0
  (3)  016362  104441                       TRAP    C$SPRI
2126  016364                                SETVEC CLKVEC,#CLKTCK,#PRI06    ;SET UP INTERRUPT VECTOR
  (7)  016364  012746  000300               MOV     #PRI06,-(SP)
  (6)  016370  012746  041066               MOV     #CLKTCK,-(SP)
  (5)  016374  013746  002344               MOV     CLKVEC,-(SP)
  (4)  016400  012746  000003               MOV     #3,-(SP)
  (3)  016404  104437                       TRAP    C$SVEC
  (2)  016406  062706  000010               ADD     #10,SP
2127  016412                                SETPRI #PRI00
  (3)  016412  012700  000000               MOV     #PRI00,R0
  (3)  016416  104441                       TRAP    C$SPRI
2128
2129  016420                           CASE 3                  ;KW11-P REAL TIME CLOCK
  (4)  016420  000434                        BR      50145$
  (4)  016422                           50147$:
2130  016422                                LET CLKSET := CLKCSR + #2
  (4)  016422  013737  002340  002342        MOV     CLKCSR,CLKSET
  (6)  016430  062737  000002  002342        ADD     #2,CLKSET
2131  016436                                LET CLKENA := #111        ;SET UP ENABLE BITS
  (4)  016436  012737  000111  002346        MOV     #111,CLKENA
2132                                        ; RUN, RATE = 10KHZ, REPEAT INTR, DOWN,INT ENABLE
2133  016444                                SETPRI #PRI07
  (3)  016444  012700  000340               MOV     #PRI07,R0
  (3)  016450  104441                       TRAP    C$SPRI
2134  016452                                SETVEC CLKVEC,#CLKTCK,#PRI06    ;INTERRUPT VECTOR
  (7)  016452  012746  000300               MOV     #PRI06,-(SP)
  (6)  016456  012746  041066               MOV     #CLKTCK,-(SP)
  (5)  016462  013746  002344               MOV     CLKVEC,-(SP)
  (4)  016466  012746  000003               MOV     #3,-(SP)
  (3)  016472  104437                       TRAP    C$SVEC
  (2)  016474  062706  000010               ADD     #10,SP
2135  016500                                SETPRI #PRI00
  (3)  016500  012700  000000               MOV     #PRI00,R0
  (3)  016504  104441                       TRAP    C$SPRI
```

```
 2136
 2137  016506                                 CASE 4
  (4)  016506  000401                             BR      50145$
  (4)  016510                                 50146$:
 2138  016510  000240                             NOP                     ;THIS IS JUST A DUMMY
 2139  016512                                 ENDSELECT
  (3)  016512                                 50145$:
 2140  016512                                      LET OUTBUF :B= #LF
  (4)  016512  112737  000012  003172             MOVB    #LF,OUTBUF
 2141  016520                                 LET LUNIT := R1
  (4)  016520  010137  002326                     MOV     R1,LUNIT
 2142  016524                                 11$:
 2143  016524                                      LET ERRFLG := #0
  (4)  016524  005037  002352                     CLR     ERRFLG
 2144  016530                                      LET R2 := LUNIT SHIFT 1
  (4)  016530  013702  002326                     MOV     LUNIT,R2
  (7)  016534  006302                             ASL     R2
 2145                                        ;
 2146                                        ; DETERMIN WHICH BAND, AND SEND APPROPRIATE PATTERN
 2147                                        ;
 2148  016536                                     IF #FLAG96 NOTSETIN STATUS(R2) THEN ; 64 CHAR BAND
  (6)  016536  032762  010000  002524             BIT     #FLAG96,STATUS(R2)
  (9)  016544  001152                             BNE     50152$
 2149  016546                                      IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG27 NOTSETIN STATUS(R2) THEN
  (6)  016546  032762  001000  002524             BIT     #FLAG26,STATUS(R2)
  (9)  016554  001065                             BNE     50153$
  (6)  016556  032762  002000  002524             BIT     #FLAG27,STATUS(R2)
  (9)  016564  001061                             BNE     50153$
 2150  016566                                      LET BNDPAT := #TABA64           ;64 CHARACTER BAND LP25
  (4)  016566  012737  021712  020742             MOV     #TABA64,BNDPAT
 2151  016574                                      LET WORK := #133.
  (4)  016574  012737  000205  003166             MOV     #133.,WORK
 2152  016602                                      PRINTF #LPM64                   ; SHOULD BE 285 LPM.
  (7)  016602  012746  021406                     MOV     #LPM64,-(SP)
  (6)  016606  012746  000001                     MOV     #1,-(SP)
  (3)  016612  010600                             MOV     SP,R0
  (4)  016614  104417                             TRAP    C$PNTF
  (4)  016616  062706  000004                     ADD     #4,SP
 2153  016622                                      OUTPUI  #LPM64+4,#42.,,LUNIT    ; SEND SPEED MSG TO PRINTER TOO
 2154  016664                                      OUTPUI #OUTBUF,#1,,LUNIT
 2155  016726                                    ELSE      ; IT'S AN LP26
  (4)  016726  000460                             BR      50154$
  (3)  016730                                 50153$:
 2156  016730                                      LET BNDPAT := #TABA64           ; LP26 64 CHAR BAND
  (4)  016730  012737  021712  020742             MOV     #TABA64,BNDPAT
 2157  016736                                      LET WORK := #133.
  (4)  016736  012737  000205  003166             MOV     #133.,WORK
 2158  016744                                      PRINTF #L26M64                  ; LP26 64 CHAR SPEED MSG. 600 LPM.
  (7)  016744  012746  021550                     MOV     #L26M64,-(SP)
  (6)  016750  012746  000001                     MOV     #1,-(SP)
  (3)  016754  010600                             MOV     SP,R0
  (4)  016756  104417                             TRAP    C$PNTF
  (4)  016760  062706  000004                     ADD     #4,SP
 2159  016764                                      OUTPUI  #L26M64+4,#42.,,LUNIT   ; SEND SPEED MSG TO PRINTER TOO
 2160  017026                                      OUTPUI #OUTBUF,#1,,LUNIT
 2161  017070                                    ENDIF
```

```
  (4)  017070                           50154$:
 2162  017070                                ELSE          ; 96 CHAR BAND
  (4)  017070  000551                             BR      50155$
  (3)  017072                           50152$:
 2163  017072                                     IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG27 NOTSETIN STATUS(R2) THEN
  (6)  017072  032762  001000  002524             BIT     #FLAG26,STATUS(R2)
  (9)  017100  001065                             BNE     50156$
  (6)  017102  032762  002000  002524             BIT     #FLAG27,STATUS(R2)
  (9)  017110  001061                             BNE     50156$
 2164  017112                                     LET BNDPAT := #TABA96           ;96 CHARACTER BAND
  (4)  017112  012737  022120  020742             MOV     #TABA96,BNDPAT
 2165  017120                                     LET WORK := #133.
  (4)  017120  012737  000205  003166             MOV     #133.,WORK
 2166  017126                                     PRINTF #LPM96                   ; SHOULD BE 204 LPM.
  (7)  017126  012746  021467                     MOV     #LPM96,-(SP)
  (6)  017132  012746  000001                     MOV     #1,-(SP)
  (3)  017136  010600                             MOV     SP,R0
  (4)  017140  104417                             TRAP    C$PNTF
  (4)  017142  062706  000004                     ADD     #4,SP
 2167  017146                                     OUTPUI  #LPM96+4,#42.,,LUNIT    ; SEND SPEED MSG TO PRINTER TOO
 2168  017210                                     OUTPUI #OUTBUF,#1,,LUNIT
 2169  017252                                ELSE
  (4)  017252  000460                             BR      50157$
  (3)  017254                           50156$:
 2170  017254                                     LET BNDPAT := #TABA96           ; LP26 96 CHAR BAND
  (4)  017254  012737  022120  020742             MOV     #TABA96,BNDPAT
 2171  017262                                     LET WORK := #133.
  (4)  017262  012737  000205  003166             MOV     #133.,WORK
 2172  017270                                     PRINTF #L26M96                  ; LP26 96 CHAR SPEED MSG. 450 LPM.
  (7)  017270  012746  021631                     MOV     #L26M96,-(SP)
  (6)  017274  012746  000001                     MOV     #1,-(SP)
  (3)  017300  010600                             MOV     SP,R0
  (4)  017302  104417                             TRAP    C$PNTF
  (4)  017304  062706  000004                     ADD     #4,SP
 2173  017310                                     OUTPUI  #L26M96+4,#42.,,LUNIT   ; SEND SPEED MSG TO PRINTER TOO
 2174  017352                                     OUTPUI #OUTBUF,#1,,LUNIT
 2175  017414                                ENDIF
  (4)  017414                           50157$:
 2176  017414                                ENDIF
  (4)  017414                           50155$:
 2177  017414                                LET LINCNT := #0                     ;CLEAR LINE COUNTER
  (4)  017414  005037  002310                     CLR     LINCNT
 2178  017420                                LET TICK := #60.                     ;SET UP INITIAL CLOCK VALUE
  (4)  017420  012737  000074  041142             MOV     #60.,TICK
 2179  017426  004737  017632           JSR PC,REPLUP                             ;DO THE OUTPUT
 2180  017432                           LET LUNIT := LUNIT - #1
  (6)  017432  005337  002326                     DEC     LUNIT
 2181  017436                           IFCOND GE THEN
  (6)  017436  002402                             BLT     50160$
 2182  017440  000137  016524                     JMP 11$
 2183  017444                           ENDIF
  (4)  017444                           50160$:
 2184
 2185  017444                           LET OUTBUF := #14
  (4)  017444  012737  000014  003172             MOV     #14,OUTBUF
 2186  017452                           OUTPUT #OUTBUF,#1
```

```
 2187
 2188  017514                                 END4:   IF CLKTYP EQ #3 THEN
  (6)  017514  023727  002334  000003                CMP     CLKTYP,#3
  (9)  017522  001011                                BNE     50161$
 2189  017524                                             SETPRI #PRI07
  (3)  017524  012700  000340                        MOV     #PRI07,R0
  (3)  017530  104441                                TRAP    C$SPRI
 2190  017532                                             CLRVEC CLKVEC
  (3)  017532  013700  002344                        MOV     CLKVEC,R0
  (3)  017536  104436                                TRAP    C$CVEC
 2191  017540                                             LET @CLKCSR := #00
  (4)  017540  012777  000000  162572                MOV     #00,@CLKCSR
 2192  017546                                             ENDIF
  (4)  017546                                50161$:
 2193  017546                                             IF CLKTYP EQ #2 THEN
  (6)  017546  023727  002334  000002                CMP     CLKTYP,#2
  (9)  017554  001021                                BNE     50162$
 2194  017556                                             SETPRI #PRI07
  (3)  017556  012700  000340                        MOV     #PRI07,R0
  (3)  017562  104441                                TRAP    C$SPRI
 2195  017564                                             SETVEC CLKVEC,#IGNORE,#PRI06
  (7)  017564  012746  000300                        MOV     #PRI06,-(SP)
  (6)  017570  012746  007266                        MOV     #IGNORE,-(SP)
  (5)  017574  013746  002344                        MOV     CLKVEC,-(SP)
  (4)  017600  012746  000003                        MOV     #3,-(SP)
  (3)  017604  104437                                TRAP    C$SVEC
  (2)  017606  062706  000010                        ADD     #10,SP
 2196  017612                                             LET @CLKCSR := #00
  (4)  017612  012777  000000  162520                MOV     #00,@CLKCSR
 2197  017620                                             ENDIF
  (4)  017620                                50162$:
 2198  017620                                             SETPRI #PRI00
  (3)  017620  012700  000000                        MOV     #PRI00,R0
  (3)  017624  104441                                TRAP    C$SPRI
 2199                                        ;
 2200  017626                                EXIT TST
  (3)  017626  104432                                TRAP    C$EXIT
  (3)  017630  002476                                .WORD   L10013-.
 2201                                        ;
 2202                                        ;
 2203                                        ;THIS IS SUBROUTINED TO DECREASE THE SIZE OF THE INITIAL INCREMENT LOOP.
 2204                                        ;
 2205                                        ;
 2206  017632                                REPLUP:
 2207  017632                                             IF CLKTYP EQ #4 THEN
  (6)  017632  023727  002334  000004                CMP     CLKTYP,#4
  (9)  017640  001124                                BNE     50163$
 2208  017642                                             PRINTF #OFFLIN              ;TELL OPERATOR TO PLACE PRINTERS OFFLINE
  (7)  017642  012746  021051                        MOV     #OFFLIN,-(SP)
  (6)  017646  012746  000001                        MOV     #1,-(SP)
  (3)  017652  010600                                MOV     SP,R0
  (4)  017654  104417                                TRAP    C$PNTF
  (4)  017656  062706  000004                        ADD     #4,SP
 2209  017662                                             LET FLAG := #0
  (4)  017662  005037  002306                        CLR     FLAG
 2210  017666                                             GMANIL READY,FLAG,100000,YES
```

```
  (3)  017666  104443                          TRAP    C$GMAN
  (3)  017670  000404                          BR      10000$
  (4)  017672  002306                          .WORD   FLAG
  (5)  017674  000130                          .WORD   T$CODE
  (5)  017676  007072                          .WORD   READY
  (5)  017700  100000                          .WORD   100000
  (3)  017702                          10000$:
2211   017702                              PRINTF #ONLIN1,LUNIT        ;PUT LUNIT TO TEST ON LINE
  (8)  017702  013746  002326                  MOV     LUNIT,-(SP)
  (7)  017706  012746  021111                  MOV     #ONLIN1,-(SP)
  (6)  017712  012746  000002                  MOV     #2,-(SP)
  (3)  017716  010600                          MOV     SP,R0
  (4)  017720  104417                          TRAP    C$PNTF
  (4)  017722  062706  000006                  ADD     #6,SP
2212   017726                              PRINTF #ONLIN2,LUNIT        ;END OF TEST.
  (8)  017726  013746  002326                  MOV     LUNIT,-(SP)
  (7)  017732  012746  021212                  MOV     #ONLIN2,-(SP)
  (6)  017736  012746  000002                  MOV     #2,-(SP)
  (3)  017742  010600                          MOV     SP,R0
  (4)  017744  104417                          TRAP    C$PNTF
  (4)  017746  062706  000006                  ADD     #6,SP
2213   017752                              PRINTF #ONLIN3,LUNIT
  (8)  017752  013746  002326                  MOV     LUNIT,-(SP)
  (7)  017756  012746  021310                  MOV     #ONLIN3,-(SP)
  (6)  017762  012746  000002                  MOV     #2,-(SP)
  (3)  017766  010600                          MOV     SP,R0
  (4)  017770  104417                          TRAP    C$PNTF
  (4)  017772  062706  000006                  ADD     #6,SP
2214   017776                              WHILE #BIT15 SETIN @LPCSR(R2) DO    ; WAIT FOR LP SET ON-LINE
  (4)  017776                          50164$:
  (6)  017776  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
  (9)  020004  001402                          BEQ     50165$
2215   020006  000240                              NOP
2216   020010                              ENDDO
  (4)  020010  000772                          BR      50164$
  (3)  020012                          50165$:
2217   020012                              LET LINCNT := #0
  (4)  020012  005037  002310                  CLR     LINCNT
2218   020016                              WHILE #BIT15 NOTSETIN @LPCSR(R2) DO    ; REPEAT UNTIL LP GOES OFF-LINE
  (4)  020016                          50166$:
  (6)  020016  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
  (9)  020024  001031                          BNE     50167$
2219   020026                                  LET R5 := BNDPAT
  (4)  020026  013705  020742                  MOV     BNDPAT,R5
2220   020032                                  LET R3 := WORK
  (4)  020032  013703  003166                  MOV     WORK,R3
2221   020036                                  WHILE R3 GT #0 DO       ; PRINT R3 CHARACTERS
  (4)  020036                          50170$:
  (6)  020036  005703                          TST     R3
  (9)  020040  003417                          BLE     50171$
2222   020042                                      WHILE #BIT7 NOTSETIN @LPCSR(R2) DO  ; WAIT FOR READY
  (4)  020042                          50172$:
  (6)  020042  032772  000200  002370          BIT     #BIT7,@LPCSR(R2)
  (9)  020050  001007                          BNE     50173$
2223   020052                                          IF #BIT15 SETIN @LPCSR(R2) THEN
  (6)  020052  032772  100000  002370          BIT     #BIT15,@LPCSR(R2)
```

```
  (9)  020060  001402                          BEQ     50174$
2224   020062  000137  020224                          JMP 99$   ; EXIT LOOP IF OFF-LINE AGAIN
2225   020066                                       ENDIF
  (4)  020066                            50174$:
2226   020066                                   ENDDO
  (4)  020066  000765                      BR      50172$
  (3)  020070                            50173$:
2227   020070                                   LET @LPBUF(R2) :B= (R5)+   ; PUT CHAR INTO LP BUFFER
  (4)  020070  112572  002464              MOVB    (R5)+,@LPBUF(R2)
2228   020074                                   LET R3 := R3 - #1      ; DECRIMENT CHAR COUNTER
  (6)  020074  005303                      DEC     R3
2229   020076                                   ENDDO
  (4)  020076  000757                      BR      50170$
  (3)  020100                            50171$:
2230   020100                                   BREAK   ; ALLOW CTL-C ABORT
  (3)  020100  104422                      TRAP    C$BRK
2231   020102                                   LET LINCNT := LINCNT + #1
  (6)  020102  005237  002310              INC     LINCNT
2232   020106                               ENDDO
  (4)  020106  000743                      BR      50166$
  (3)  020110                            50167$:
2233   020110                               ELSE
  (4)  020110  000445                      BR      50175$
  (3)  020112                            50163$:
2234   020112                               IF CLKTYP EQ #3 THEN
  (6)  020112  023727  002334  000003      CMP     CLKTYP,#3
  (9)  020120  001003                      BNE     50176$
2235   020122                                   LET @CLKSET := #1666.   ; 1/60 SEC.
  (4)  020122  012777  003202  162212      MOV     #1666.,@CLKSET
2236   020130                               ENDIF
  (4)  020130                            50176$:
2237   020130                               LET @CLKCSR := CLKENA       ;ENABLE THE CLOCK TO DO ITS THING
  (4)  020130  013777  002346  162202      MOV     CLKENA,@CLKCSR
2238   020136                               LET TIME := #0
  (4)  020136  005037  041140              CLR     TIME
2239   020142                               LET LINCNT := #0
  (4)  020142  005037  002310              CLR     LINCNT
2240   020146                               WHILE TIME LT PERIOD DO             ; REPEAT UNTIL TIME EXHAUSTED
  (4)  020146                            50177$:
  (6)  020146  023737  041140  002300      CMP     TIME,PERIOD
  (9)  020154  002023                      BGE     50200$
2241   020156                                   LET R5 := BNDPAT
  (4)  020156  013705  020742              MOV     BNDPAT,R5
2242   020162                                   LET R3 := WORK
  (4)  020162  013703  003166              MOV     WORK,R3
2243   020166                                   WHILE R3 GT #0 DO       ; SEND R3 CHARACTERS
  (4)  020166                            50201$:
  (6)  020166  005703                      TST     R3
  (9)  020170  003412                      BLE     50202$
2244   020172                                   WHILE #BIT7 NOTSETIN @LPCSR(R2) DO  ; WAIT FOR READY
  (4)  020172                            50203$:
  (6)  020172  032772  000200  002370      BIT     #BIT7,@LPCSR(R2)
  (9)  020200  001002                      BNE     50204$
2245   020202  000240                           NOP
2246   020204                                   ENDDO
  (4)  020204  000772                      BR      50203$
```

```
  (3)  020206                          50204$:
 2247  020206                                  LET @LPBUF(R2) :B= (R5)+          ; PUT DATA INTO BUFFER
  (4)  020206  112572  002464                 MOVB    (R5)+,@LPBUF(R2)
 2248  020212                                  LET R3 := R3 - #1                 ; DECRIMENT CHAR COUNTER
  (6)  020212  005303                         DEC     R3
 2249  020214                                  ENDDO
  (4)  020214  000764                         BR      50201$
  (3)  020216                          50202$:
 2250  020216                                  LET LINCNT := LINCNT + #1
  (6)  020216  005237  002310                 INC     LINCNT
 2251  020222                              ENDDO
  (4)  020222  000751                         BR      50177$
  (3)  020224                          50200$:
 2252  020224                                  ENDIF
  (4)  020224                          50175$:
 2253  020224                          99$:
 2254                                 ;
 2255                                 ;IF MANUAL PRINT SPEED TESTS HAVE BEEN PERFORMED INSURE PRINTERS ARE
 2256                                 ;BACK ON LINE WHEN DONE
 2257                                 ;
 2258  020224                          IF CLKTYP EQ #4 THEN
  (6)  020224  023727  002334  000004         CMP     CLKTYP,#4
  (9)  020232  001020                         BNE     50205$
 2259  020234                              LET FLAG := #0                        ;CLEAR <CR> FLAG
  (4)  020234  005037  002306                 CLR     FLAG
 2260  020240                              PRINTF #MRESET
  (7)  020240  012746  006773                 MOV     #MRESET,-(SP)
  (6)  020244  012746  000001                 MOV     #1,-(SP)
  (3)  020250  010600                         MOV     SP,R0
  (4)  020252  104417                         TRAP    C$PNTF
  (4)  020254  062706  000004                 ADD     #4,SP
 2261  020260                              GMANIL READY,FLAG,100000,YES          ;WAIT FOR OPERATOR
  (3)  020260  104443                         TRAP    C$GMAN
  (3)  020262  000404                         BR      10001$
  (4)  020264  002306                         .WORD   FLAG
  (5)  020266  000130                         .WORD   T$CODE
  (5)  020270  007072                         .WORD   READY
  (5)  020272  100000                         .WORD   100000
  (3)  020274                          10001$:
 2262  020274                          ENDIF
  (4)  020274                          50205$:
 2263  020274  012777  000000  162036         MOV     #00,@CLKCSR
 2264                                 ;
 2265                                 ; REPORT TOTAL NUMBER OF LINES PRINTED
 2266                                 ;
 2267  020302                              PRINTB  #LINPER,LINCNT,LUNIT
  (9)  020302  013746  002326                 MOV     LUNIT,-(SP)
  (8)  020306  013746  002310                 MOV     LINCNT,-(SP)
  (7)  020312  012746  021004                 MOV     #LINPER,-(SP)
  (6)  020316  012746  000003                 MOV     #3,-(SP)
  (3)  020322  010600                         MOV     SP,R0
  (4)  020324  104414                         TRAP    C$PNTB
  (4)  020326  062706  000010                 ADD     #10,SP
 2268  020332                                  PUSH    LINCNT,#OUTBUF+1          ; CONVERT LINE COUNT TO ASCII
  (2)  020332  013746  002310                 MOV     LINCNT,-(SP)
  (3)  020336  012746  003173                 MOV     #OUTBUF+1,-(SP)
```

```
2269  020342  004737  004472                        JSR     PC,BIN2DA
2270  020346                                        OUTPUI  #OUTBUF,#6,,LUNIT       ; DISPLAY LINE COUNT ON LP
2271  020410  004737  005466                        JSR     PC,QUIET
2272  020414                                        OUTPUI  #SPED1,#19.,,LUNIT      ; "LINES PRINTED"
2273                                                ; IF A CLOCK WAS USED DISPLAY THE TIME USED ALSO
2274  020456                                        IF CLKTYP EQ #2 OR CLKTYP EQ #3 THEN
 (6)  020456  023727  002334  000002                CMP     CLKTYP,#2
 (8)  020464  001404                                BEQ     50206$
 (6)  020466  023727  002334  000003                CMP     CLKTYP,#3
 (9)  020474  001074                                BNE     50207$
 (6)  020476                                50206$:
2275  020476                                            OUTPUI #SPED2,#4,,LUNIT     ; " IN "
2276  020540                                            PUSH  PERIOD,#OUTBUF+1      ; CONVERT TIME TO ASCII
 (2)  020540  013746  002300                        MOV     PERIOD,-(SP)
 (3)  020544  012746  003173                        MOV     #OUTBUF+1,-(SP)
2277  020550  004737  004472                            JSR PC,BIN2DA
2278  020554                                            OUTPUI      #OUTBUF+3,#3,,LUNIT  ; DISPLAY THE TIME IN SECONDS
2279  020616  004737  005466                        JSR     PC,QUIET
2280  020622                                            OUTPUI      #SPED3,#9.,,LUNIT    ; "SECONDS" <FF>
2281  020664                                        ELSE
 (4)  020664  000421                                BR      50210$
 (3)  020666                                50207$:
2282  020666                                            OUTPUI      #SPED4,#1,,LUNIT        ; <FF>
2283  020730                                        ENDIF
 (4)  020730                                50210$:
2284  020730  000207                                RTS PC                                  ;GO BACK AND DO IT AGAIN
2285                                                ;
2286                                                ;EXPECTED ERROR HANDLER
2287                                                ;
2288  020732                                    LPERR2: LET ERRFLG := #1                    ;SET ERROR FOUND
 (4)  020732  012737  000001  002352                    MOV     #1,ERRFLG
2289  020740  000207                                    RTS     PC                          ;AND EXIT
2290                                                ;
2291                                                ;
2292  020742  000000                            BNDPAT: .WORD   0               ; CONTAINS ADDRESS OF PRINT PATTERN
2293                                                .NLIST BEX
2294                                                ;
2295                                                ;ASSOCIATED MESSAGES
2296                                                ;
2297  020744  046040  047111  051505            SPED1:  .ASCII  / LINES WERE PRINTED/
2298  020767     040  047111     040            SPED2:  .ASCII  / IN /
2299  020773     040  042523  047503            SPED3:  .ASCII  / SECONDS/
2300  021003     014                            SPED4:  .BYTE   14
2301  021004  047045  042045  022463            LINPER: .ASCIZ  /%N%D3%A LINES PRINTED ON LUNIT %D2%N/
2302  021051     045  022516  044501            OFFLIN: .ASCIZ /%N%AINSURE PRINTER(S) OFF LINE./
2303  021111     045  022516  050101            ONLIN1: .ASCIZ/%N%APLACE LUNIT %D2%A ON LINE TO INITIATE TIME PERIOD FOR MANUAL/
2304  021212  047045  040445  051120            ONLIN2: .ASCIZ  /%N%APRINTING SPEED MEASUREMENT AND BACK OFF LINE TO TERMINATE/
2305  021310  047045  040445  044124            ONLIN3: .ASCIZ /%N%ATHE TIME INTERVAL.%N/
2306  021341     120  044522  052116            PRTSPD: .ASCIZ  /PRINTING SPEED MEASUREMENT TEST 4/<12><12><12>
2307  021406  047045  040445  032066            LPM64:  .ASCIZ  /%N%A64 CHARACTER BAND SHOULD PRINT AT 285 LPM.%N/
2308  021467     045  022516  034501            LPM96:  .ASCIZ  /%N%A96 CHARACTER BAND SHOULD PRINT AT 204 LPM.%N/
2309  021550  047045  040445  032066            L26M64: .ASCIZ  /%N%A64 CHARACTER BAND SHOULD PRINT AT 600 LPM.%N/
2310  021631     045  022516  034501            L26M96: .ASCIZ  /%N%A96 CHARACTER BAND SHOULD PRINT AT 450 LPM.%N/
2311                                                .LIST BEX
2312                                                .EVEN
2313                                                ;64 CHARACTER BAND  PATTERN  285 LPM / 600 LPM.
```

```
2314                                    ;
2315                                    .SBTTL  PRINT SPEED TEST PATTERNS
2316
2317  021712  105  061  104  TABA64: .BYTE   105,061,104,075,064,041,103,136,102,060,163
      021715  075  064  041
      021720  103  136  102
      021723  060  163
2318  021725  042  062  134          .BYTE   042,062,134,054,124,101,133,101,133,043,135
      021730  054  124  101
      021733  133  101  133
      021736  043  135
2319  021740  041  105  061          .BYTE   041,105,061,100,075,077,041,056,136,074,060
      021743  100  075  077
      021746  041  056  136
      021751  074  060
2320  021753  076  042  073          .BYTE   076,042,073,042,073,134,055,124,044,133,057
      021756  042  073  134
      021761  055  124  044
      021764  133  057
2321  021766  135  054  105          .BYTE   135,054,105,072,100,050,077,052,056,051,056
      021771  072  100  050
      021774  077  052  056
      021777  051  056
2322  022001  051  074  046          .BYTE   051,074,046,076,071,073,045,055,053,044,137
      022004  076  071  073
      022007  045  055  053
      022012  044  137
2323  022014  057  070  054          .BYTE   057,070,054,132,072,131,072,131,050,067,052
      022017  132  072  131
      022022  072  131  050
      022025  067  052
2324  022027  130  051  127          .BYTE   130,051,127,046,066,071,126,045,125,053,065
      022032  046  066  071
      022035  126  045  125
      022040  053  065
2325  022042  137  123  137          .BYTE   137,123,137,123,070,122,132,121,131,064,067
      022045  123  070  122
      022050  132  121  131
      022053  064  067
2326  022055  120  130  117          .BYTE   120,130,117,124,063,066,116,126,115,126,115
      022060  124  063  066
      022063  116  126  115
      022066  126  115
2327  022070  125  062  065          .BYTE   125,062,065,114,123,113,122,061,121,112,064
      022073  114  123  113
      022076  122  061  121
      022101  112  064
2328  022103  111  120  110          .BYTE   111,120,110,117,060,117,060,063,107,116,106,012,015
      022106  117  060  117
      022111  060  063  107
      022114  116  106  012
      022117  015
2329
2330                                    ;
2331                                    ;96 CHARACTER BAND TABLE  204 LPM. / 450 LPM.
2332                                    ;       MINIMUM PRINT SPEED PATTERN 96 CHARACTER BAND
```

```
 2333  022120  061  055  144  TABA96: .BYTE   061,055,144,047,143,043,142,041,060,052,100
       022123  047  143  043
       022126  142  041  060
       022131  052  100
 2334  022133  075  140  174          .BYTE   075,140,174,176,041,056,054,056,054,136,042
       022136  176  041  056
       022141  054  056  054
       022144  136  042
 2335  022146  176  134  173          .BYTE   176,134,173,133,175,135,055,164,047,100,043
       022151  133  175  135
       022154  055  164  047
       022157  100  043
 2336  022161  077  041  074          .BYTE   077,041,074,041,074,052,062,075,076,174,073
       022164  041  074  052
       022167  062  075  076
       022172  174  073
 2337  022174  041  053  054          .BYTE   041,053,054,071,042,057,134,072,133,050,133
       022177  071  042  057
       022202  134  072  133
       022205  050  133
 2338  022207  050  135  051          .BYTE   050,135,051,164,070,100,046,124,045,123,044
       022212  164  070  100
       022215  046  124  045
       022220  123  044
 2339  022222  122  067  064          .BYTE   122,067,064,137,073,132,073,132,053,131,071
       022225  137  073  132
       022230  073  132  053
       022233  131  071
 2340  022235  066  057  130          .BYTE   066,057,130,072,127,050,120,151,125,070,065
       022240  072  127  050
       022243  120  151  125
       022246  070  065
 2341  022250  046  124  046          .BYTE   046,124,046,124,045,123,044,122,067,064,137
       022253  124  045  123
       022256  044  122  067
       022261  064  137
 2342  022263  121  132  120          .BYTE   121,132,120,131,117,066,063,130,116,130,116
       022266  131  117  066
       022271  063  130  116
       022274  130  116
 2343  022276  127  115  126          .BYTE   127,115,126,114,125,113,065,062,124,112,123
       022301  114  125  113
       022304  065  062  124
       022307  112  123
 2344  022311  111  122  110          .BYTE   111,122,110,064,061,064,061,121,107,102,106,012,015
       022314  064  061  064
       022317  061  121  107
       022322  102  106  012
       022325  015
 2345                          .EVEN
 2346  022326                  ENDTST
   (3) 022326                  L10013:
   (3) 022326  104401                  TRAP    C$ETST
 2347  022330                  ENDMOD
```

```
                                .SBTTL DAVFU ERROR DETECTION

                                BGNMOD
                                ;++
                                ;THIS TEST CONSISTS OF TWO PARTS TO VERIFY
                                ;THAT THE DAVFU CAN DETECT ERROR CONDITIONS
                                ;OF TWO TYPES:
                                ;1. DAVFU WILL NOT ACCEPT INCOMPLETE DATA.
                                ;2. DAVFU WILL NOT ACCEPT DATA THAT DOES
                                ;   NOT INCLUDE A STOP BIT (ONE) CHARACTER.
                                ;--
022330                          BGNTST  5
(3)  022330                     T5::
                                ;EXIT TEST IF DAVFU OPTION IS NOT SPECIFIED
022330                          IF VFUOPT EQ #0 THEN
(6)  022330  005737  002274             TST     VFUOPT
(9)  022334  001002             BNE     50211$
022336                          EXIT TST
(3)  022336  104432             TRAP    C$EXIT
(3)  022340  001354             .WORD   L10014-.
022342                          ENDIF
(4)  022342                     50211$:
                                ;PRINT TEST IDENTIFICATION
022342                          OUTPUT #VFUERR, #32., LPERR
                                ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
022404                          MANUAL
(3)  022404  104450             TRAP    C$MANI
022406                          BCOMPLETE 1$
(2)  022406  103402             BCS     1$
                                ;EXIT TEST IF NEGATIVE DETERMINATION FOR MANUAL
                                ;INTERVENTION IS MADE
022410                          EXIT TST
(3)  022410  104432             TRAP    C$EXIT
(3)  022412  001302             .WORD   L10014-.
                                ;DETERMINE IF INTERVENTION IS INHIBITED
022414  005737  002272     1$:   TST INHINT
022420  001002                  BNE 2$
022422                          EXIT TST
(3)  022422  104432             TRAP    C$EXIT
(3)  022424  001270             .WORD   L10014-.
022426                  2$:     BGNSUB
(3)  022426                     T5.1:
(3)  022426  104402             TRAP    C$BSUB
                                ;SEND MESSAGE TO OPERATOR TO SET VFU-FLS SWITCH(ES)
022430                          PRINTF  #VFUSEL
(7)  022430  012746  004045     MOV     #VFUSEL,-(SP)
(6)  022434  012746  000001     MOV     #1,-(SP)
(3)  022440  010600             MOV     SP,R0
(4)  022442  104417             TRAP    C$PNTF
(4)  022444  062706  000004     ADD     #4,SP
                                ;WAIT FOR OPERATOR RESPONSE
022450                          LET FLAG := #0
(4)  022450  005037  002306     CLR     FLAG
022454                          GMANIL READY,FLAG,100000,YES
(3)  022454  104443             TRAP    C$GMAN
(3)  022456  000404             BR      10000$
```

```
   (4)  022460  002306                        .WORD   FLAG
   (5)  022462  000130                        .WORD   T$CODE
   (5)  022464  007072                        .WORD   READY
   (5)  022466  100000                        .WORD   100000
   (3)  022470                          10000$:
  2383                                  ;PERFORM INCOMPLETE DATA ERROR DETECTION TEST
  2384  022470                                LET R1 := L$UNIT - #1
   (4)  022470  013701  002012                MOV     L$UNIT,R1
   (6)  022474  005301                        DEC     R1
  2385  022476                              INCR LUNIT FROM #0 TO R1 BY #1
   (4)  022476  005037  002326                CLR     LUNIT
   (5)  022502  000402                        BR      50212$
   (4)  022504                          50213$:
   (7)  022504  005237  002326                INC     LUNIT
   (5)  022510                          50212$:
   (5)  022510  023701  002326                CMP     LUNIT,R1
   (7)  022514  003064                        BGT     50214$
  2386  022516                                LET R2 := LUNIT SHIFT 1
   (4)  022516  013702  002326                MOV     LUNIT,R2
   (7)  022522  006302                        ASL     R2
  2387  022524                                LET FLAG := #0
   (4)  022524  005037  002306                CLR     FLAG
  2388  022530                                OUTPUI #INCDAT, #41.,, LUNIT
  2389  022572                                OUTPUI #INCTBL, #5, #GETFLG, LUNIT
  2390  022634                                IF FLAG EQ #0 THEN
   (6)  022634  005737  002306                TST     FLAG
   (9)  022640  001011                        BNE     50215$
  2391  022642                                LET ERRTBL(R2) := ERRTBL(R2) + #1
   (6)  022642  005262  003126                INC     ERRTBL(R2)
  2392  022646                                LET L$LUN := LUNIT
   (4)  022646  013737  002326  002074        MOV     LUNIT,L$LUN
  2393  022654                                    ERRHRD 11, ERR06
   (4)  022654  104456                        TRAP    C$ERHRD
   (5)  022656  000013                        .WORD   11
   (5)  022660  023563                        .WORD   ERR06
   (5)  022662  000000                        .WORD   0
  2394  022664                                ENDIF
   (4)  022664                          50215$:
  2395  022664                            ENDINC
   (4)  022664  000707                        BR      50213$
   (3)  022666                          50214$:
  2396  022666                            ENDSUB
   (3)  022666                          L10015:
   (3)  022666  104403                        TRAP    C$ESUB
  2397
  2398  022670                            BGNSUB
   (3)  022670                          T5.2:
   (3)  022670  104402                        TRAP    C$BSUB
  2399                                  ;PERFORM STOP CODE ERROR DETECTION TEST
  2400  022672                              INCR LUNIT FROM #0 TO R1 BY #1
   (4)  022672  005037  002326                CLR     LUNIT
   (5)  022676  000402                        BR      50216$
   (4)  022700                          50217$:
   (7)  022700  005237  002326                INC     LUNIT
   (5)  022704                          50216$:
   (5)  022704  023701  002326                CMP     LUNIT,R1
```

```
  (7)  022710  003163                          BGT     50220$
2401   022712                                  LET R2 := LUNIT SHIFT 1
  (4)  022712  013702  002326                  MOV     LUNIT,R2
  (7)  022716  006302                          ASL     R2
2402   022720                                  LET FLAG := #0
  (4)  022720  005037  002306                  CLR     FLAG
2403   022724                                  OUTPUI #NOSTOP, #35.,, LUNIT
2404   022766                                  OUTPUI #NSTTBL, #6, #GETFLG, LUNIT
2405   023030                                  INCR VFUCMD FROM #200 TO #213 BY #1
  (4)  023030  012737  000200  002360          MOV     #200,VFUCMD
  (5)  023036  000402                          BR      50221$
  (4)  023040                          50222$:
  (7)  023040  005237  002360                  INC     VFUCMD
  (5)  023044                          50221$:
  (5)  023044  023727  002360  000213          CMP     VFUCMD,#213
  (7)  023052  003055                          BGT     50223$
2406   023054                                      LET OUTBUF := #15           ;''CR'' TO OUTPUT BUFFER
  (4)  023054  012737  000015  003172          MOV     #15,OUTBUF
2407   023062                                      OUTPUI #OUTBUF, #1, GETFLG, LUNIT
2408   023124                                      DELAY 2
  (2)  023124  012727  000002                  MOV     #2,(PC)+
  (2)  023130  000000                          .WORD   0
  (2)  023132  013727  002116                  MOV     L$DLY,(PC)+
  (2)  023136  000000                          .WORD   0
  (2)  023140  005367  177772                  DEC     -6(PC)
  (2)  023144  001375                          BNE     .-4
  (2)  023146  005367  177756                  DEC     -22(PC)
  (2)  023152  001367                          BNE     .-20
2409   023154                                      IF FLAG EQ #0 THEN
  (6)  023154  005737  002306                  TST     FLAG
  (9)  023160  001011                          BNE     50224$
2410   023162                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  023162  005262  003126                  INC     ERRTBL(R2)
2411   023166                                  LET L$LUN := LUNIT
  (4)  023166  013737  002326  002074          MOV     LUNIT,L$LUN
2412   023174                                      ERRHRD 12, ERR07
  (4)  023174  104456                          TRAP    C$ERHRD
  (5)  023176  000014                          .WORD   12
  (5)  023200  023634                          .WORD   ERR07
  (5)  023202  000000                          .WORD   0
2413   023204                                      ENDIF
  (4)  023204                          50224$:
2414   023204                                  ENDINC
  (4)  023204  000715                          BR      50222$
  (3)  023206                          50223$:
2415   023206                                  LET OUTBUF := #14               ;''FF'' TO OUTPUT BUFFER
  (4)  023206  012737  000014  003172          MOV     #14,OUTBUF
2416   023214                                  OUTPUI #OUTBUF, #1,,LUNIT
2417   023256                              ENDINC
  (4)  023256  000610                          BR      50217$
  (3)  023260                          50220$:
2418   023260                              EXIT TST
  (3)  023260  104432                          TRAP    C$EXIT
  (3)  023262  000432                          .WORD   L10014-.
2419   023264                              ENDSUB
  (3)  023264                          L10016:
```

```
  (3)  023264  104403                        TRAP    C$ESUB
2420                                 ;
2421                                 ;EXPECTED ERROR ROUTINE FOR THIS TEST
2422                                 ;
2423  023266                         GETFLG: GMANIL RESET, FLAG, 100000, YES
  (3)  023266  104443                        TRAP    C$GMAN
  (3)  023270  000404                        BR      10000$
  (4)  023272  002306                        .WORD   FLAG
  (5)  023274  000130                        .WORD   T$CODE
  (5)  023276  023460                        .WORD   RESET
  (5)  023300  100000                        .WORD   100000
  (3)  023302                        10000$:
2424  023302  000207                 RTS     PC                      ;RETURN
2425                                 ;
2426                                 ;
2427
2428                                 .NLIST BEX
2429  023304  040504  043126  020125 VFUERR: .ASCIZ /DAVFU ERROR DETECTION TEST 5/<12><12><12>
2430  023344  040504  043126  020125 INCDAT: .ASCIZ /DAVFU INCOMPLETE DATA ERROR DETECTION/<12><12><12>
2431  023415     104  053101  052506 NOSTOP: .ASCIZ /DAVFU STOP CODE ERROR DETECTION/<12><12><12>
2432  023460  042524  052123  047440 RESET:  .ASCII /TEST O.K. - PLACE PRINTER ON LINE AND DEPRESS/
2433  023535     042  042522  052524         .ASCIZ  /"RETURN" WHEN READY./<12>
2434
2435  023563     104  053101  052506 ERR06:  .ASCIZ /DAVFU INCOMPLETE DATA ERROR NOT DETECTED/
2436  023634  040504  043126  020125 FRR07:  .ASCIZ /DAVFU STOP CODE  ERROR NOT DETECTED/
2437  023700     356     001     002 INCTBL: .BYTE 356, 1, 2, 3, 357
2438  023705     356     000     000 NSTTBL: .BYTE 356, 0, 0, 0, 0, 357
2439          023714                         .EVEN
2440
2441
2442
2443
2444
2445                                 .LIST BEX
2446  023714                         ENDTST
  (3)  023714                        L10014:
  (3)  023714  104401                        TRAP    C$ETST
2447
2448  023716                         ENDMOD
```

```
2450                                    .SBTTL DAVFU LINE COUNT PAPER CONTROL
2451  023716                            BGNMOD
2452                                    ;++
2453                                    ;THIS TEST CHECKS THE LINE COUNT METHOD
2454                                    ;OF PAPER ADVANCE USING THE DAVFU. THE
2455                                    ;DAVFU MEMORY IS LOADED WITH DUMMY
2456                                    ;DATA, AND THEN EACH OF THE LINE COUNT
2457                                    ;SLEWING COMMANDS IS TESTED IN SEQUENCE
2458                                    ;IN THE RANGE FROM ZERO TO 15 LINES.
2459                                    ;--
2460
2461  023716                            BGNTST  6
 (3)  023716                            T6::
2462                                    ;EXIT TEST IF DAVFU OPTION IS NOT SPECIFIED
2463  023716                                    IF VFUOPT EQ #0 THEN
 (6)  023716  005737  002274                      TST     VFUOPT
 (9)  023722  001002                              BNE     50225$
                                                      EXIT TST
2464  023724
 (3)  023724  104432                              TRAP    C$EXIT
 (3)  023726  001472                              .WORD   L10017-.
2465  023730                                    ENDIF
 (4)  023730                            50225$:
2466                                    ;PRINT TEST IDENTIFICATION
2467  023730                            OUTPUT #VFULCT, #40., LPERR
2468                                    ;SEND MESSAGE TO OPERATOR TO SET VFU-FLS SWITCH(ES)
2469  023772                            PRINTF  #VFUSEL
 (7)  023772  012746  004045                      MOV     #VFUSEL,-(SP)
 (6)  023776  012746  000001                      MOV     #1,-(SP)
 (3)  024002  010600                              MOV     SP,R0
 (4)  024004  104417                              TRAP    C$PNTF
 (4)  024006  062706  000004                      ADD     #4,SP
2470                                    ;WAIT FOR OPERATOR RESPONSE
2471  024012                            LET FLAG := #0
 (4)  024012  005037  002306                      CLR     FLAG
2472  024016                            GMANIL READY,FLAG,100000,YES
 (3)  024016  104443                              TRAP    C$GMAN
 (3)  024020  000404                              BR      10000$
 (4)  024022  002306                              .WORD   FLAG
 (5)  024024  000130                              .WORD   T$CODE
 (5)  024026  007072                              .WORD   READY
 (5)  024030  100000                              .WORD   100000
 (3)  024032                            10000$:
2473                                    ;INITIALIZE PARAMETERS
2474  024032                            LET VFUCMD := #200
 (4)  024032  012737  000200  002360              MOV     #200,VFUCMD
2475  024040                            LET R4 := #0
 (4)  024040  005004                              CLR     R4
2476                                    ;LOAD DAVFU MEMORY
2477  024042                            OUTPUT #VFUTBL, #18.
2478                                    ;PRINT FIRST PART OF ZERO LINE SLEW MESSAGE
2479  024104                            OUTPUT #FSTMSG, #29.
2480                                    ;SEND ZERO LINE SLEW COMMAND
2481  024146                            OUTPUT #VFUCMD, #1.
2482                                    ;PRINT SECOND PART OF ZERO LINE SLEW MESSAGE
2483  024210                            OUTPUT #SECMSG, #103.
2484  024252                            LET OUTBUF := #15                     ;"CR" TO OUTPUT BUFFER
```

```
  (4)  024252  012737  000015  003172          MOV     #15,OUTBUF
2485   024260                                OUTPUT #OUTBUF, #1.
2486                                         ;SEND OTHER DAVFU PAPER ADVANCE COMMANDS
2487   024322                                INCR VFUCMD FROM #221 TO #237 BY #1
  (4)  024322  012737  000221  002360          MOV     #221,VFUCMD
  (5)  024330  000402                          BR      50226$
  (4)  024332                                50227$:
  (7)  024332  005237  002360                  INC     VFUCMD
  (5)  024336                                50226$:
  (5)  024336  023727  002360  000237          CMP     VFUCMD,#237
  (7)  024344  003115                          BGT     50230$
2488                                         ;PERFORM PAPER MOVEMENT
2489   024346                                    OUTPUT #VFUCMD, #1.
2490   024410                                    LET OUTBUF := #15
  (4)  024410  012737  000015  003172          MOV     #15,OUTBUF
2491   024416                                    OUTPUT #OUTBUF, #1.
2492   024460                                    LET OUTBUF := LCTTBL(R4)
  (4)  024460  016437  024726  003172          MOV     LCTTBL(R4),OUTBUF
2493   024466                                   OUTPUT #OUTBUF,#2
2494                                         ;APPEND MESSAGE AND PRINT MOVEMENT MESSAGE
2495   024530                                    OUTPUT #LCTMSG, #131.
2496   024572                                    LET R4 := R4 + #2
  (6)  024572  062704  000002                  ADD     #2,R4
2497   024576                                ENDINC
  (4)  024576  000655                          BR      50227$
  (3)  024600                                50230$:
2498   024600                                LET OUTBUF := #14
  (4)  024600  012737  000014  003172          MOV     #14,OUTBUF
2499   024606                                OUTPUT #OUTBUF, #1.
2500   024650                                EXIT TST
  (3)  024650  104432                          TRAP    C$EXIT
  (3)  024652  000546                          .WORD   L10017-.
2501                                         .NLIST BEX
2502   024654  040504  043126  020125  VFULCT: .ASCIZ /DAVFU LINE COUNT PAPER CONTROL TEST 6/ <12><12><12>
2503           024726                        .EVEN
2504   024726  030460  031060  031460  LCTTBL: .ASCIZ/0102030405060708091011121314 15/
2505
2506   024765     356     001     002  VFUTBL: .BYTE 356, 1, 2, 3, 4, 5, 6
2507   024774     007     010     011          .BYTE 7, 10, 11, 12, 13, 14
2508   025002     015     016     017          .BYTE 15, 16, 17, 20, 357
2509
2510   025007     124  044510  020123  FSTMSG: .ASCIZ /THIS LINE SHOULD BE PRINTED /
2511
2512   025044  046101  020114  047117  SECMSG: .ASCII /ALL ON ONE LINE IF SLEWED ZERO LINES/
2513   025110  027056  027056  027056          .ASCII  /......................................................./
2514   025206  027056  000056                  .ASCIZ  /.../
2515   025212  041040  040514  045516  LCTMSG: .ASCII / BLANK LINES SHOULD OCCUR BETWEEN THIS LINE AND THE/
2516   025275     040  051120  053105          .ASCII  / PREVIOUS LINE ...................../
2517   025336  027056  027056  027056          .ASCII  /......................................./
2518   025402  027056  027056  027056          .ASCIZ  /.......... / <15>
2519                                         .EVEN
2520                                         .LIST BEX
2521   025420                                ENDTST
  (3)  025420                                L10017:
  (3)  025420  104401                          TRAP    C$ETST
2522   025422                                ENDMOD
```

```
                                .SBTTL DAVFU CHANNEL SELECTION PAPER CONTROL

                                BGNMOD
                                ;++
                                ;THIS TEST CHECKS DAVFU PAPER ADVANCE USING
                                ;STOP BITS LOADED IN DAVFU MEMORY. THE
                                ;DATA FORMAT IS SELECTED TO PROVIDE AN
                                ;OUTPUT IN A TRIANGULAR PATTERN.
                                ;--

                                BGNTST  7
  (3)  025422                   T7::
                                ;EXIT TEST IF DAVFU OPTION IS NOT SPECIFIED
                                        IF VFUOPT EQ #0 THEN
  (6)  025422  005737  002274           TST     VFUOPT
  (9)  025426  001002                   BNE     50231$
                                            EXIT TST
  (3)  025430  104432                   TRAP    C$EXIT
  (3)  025432  001236                   .WORD   L10020-.
                                        ENDIF
  (4)  025434                   50231$:
                                ;PRINT TEST IDENTIFICATION
  025434                        OUTPUT #VFUCHL, #47.
                                ;SEND MESSAGE TO OPERATOR TO SET VFU-FLS SWITCH(ES)
  025476                        PRINTF  #VFUSEL
  (7)  025476  012746  004045           MOV     #VFUSEL,-(SP)
  (6)  025502  012746  000001           MOV     #1,-(SP)
  (3)  025506  010600                   MOV     SP,R0
  (4)  025510  104417                   TRAP    C$PNTF
  (4)  025512  062706  000004           ADD     #4,SP
                                ;WAIT FOR RESPONSE FROM OPERATOR
  025516                        LET FLAG := #0
  (4)  025516  005037  002306           CLR     FLAG
  025522                        GMANIL READY,FLAG,100000,YES
  (3)  025522  104443                   TRAP    C$GMAN
  (3)  025524  000404                   BR      10000$
  (4)  025526  002306                   .WORD   FLAG
  (5)  025530  000130                   .WORD   T$CODE
  (5)  025532  007072                   .WORD   READY
  (5)  025534  100000                   .WORD   100000
  (3)  025536                   10000$:
                                ;LOAD DAVFU MEMORY
  025536                        OUTPUT #VFUDAT, #50.
  025600  012704  000002        MOV     #2,R4           ;SET UP ITERATION COUNTER
  025604  000401                BR      1$
  025606  005304                2$:     DEC     R4              ;BACK UP COUNTER
  025610  020427  000001        1$:     CMP     R4,#1           ;TEST FOR LAST TIME THROUGH
  025614  002002                BGE     3$              ;BRANCH IF LAST TIME THROUGH FALSE
  025616  000137  026446        JMP     4$              ;EXIT TEST
                                ;SEND PAPER INSTRUCTIONS TO ALL 12 CHANNELS
  025622                        3$: INCR INSTR FROM #200 TO #213 BY #1
  (4)  025622  012737  000200  026522    MOV     #200,INSTR
  (5)  025630  000402                   BR      50232$
  (4)  025632                   50233$:
  (7)  025632  005237  026522           INC     INSTR
  (5)  025636                   50232$:
```

```
  (5)  025636  023727  026522  000213          CMP     INSTR,#213
  (7)  025644  003144                          BGT     50234$
2556                                        ;PERFORM PAPER MOVEMENT
2557  025646                                    OUTPUT #INSTR, #1
2558  025710                                    LET OUTBUF := #15
  (4)  025710  012737  000015  003172          MOV     #15,OUTBUF
2559  025716                                    OUTPUT #OUTBUF, #1
2560                                        ;GET SPACE COUNT
2561  025760                                    LET R3 := INSTR
  (4)  025760  013703  026522                  MOV     INSTR,R3
2562  025764  142703  000360                   BICB #360, R3
2563  025770                                    WHILE R3 NE #0 DO
  (4)  025770                               50235$:
  (6)  025770  005703                          TST     R3
  (9)  025772  001423                          BEQ     50236$
2564  025774                                    OUTPUT #CHARSP, #1
2565  026036                                    LET R3 := R3 - #1
  (6)  026036  005303                          DEC     R3
2566  026040                                    ENDDO
  (4)  026040  000753                          BR      50235$
  (3)  026042                               50236$:
2567  026042                                    OUTPUT #CHARX, #1
2568                                        ;SEND PRINT COMMAND TO PRINT SPACES AND X'S
2569  026104                                    LET OUTBUF := #15
  (4)  026104  012737  000015  003172          MOV     #15,OUTBUF
2570  026112                                    OUTPUT #OUTBUF, #1.
2571  026154                                ENDINC
  (4)  026154  000626                          BR      50233$
  (3)  026156                               50234$:
2572                                        ;DUPLICATE TEST TO REVERSE OUTPUT PATTERN
2573  026156                                DECR INSTR FROM #213 TO #200 BY #1
  (4)  026156  012737  000213  026522          MOV     #213,INSTR
  (5)  026164  000402                          BR      50237$
  (4)  026166                               50240$:
  (7)  026166  005337  026522                  DEC     INSTR
  (5)  026172                               50237$:
  (5)  026172  023727  026522  000200          CMP     INSTR,#200
  (7)  026200  002520                          BLT     50241$
2574  026202                                OUTPUT #INSTR, #1.
2575  026244                                LET R3 := INSTR
  (4)  026244  013703  026522                  MOV     INSTR,R3
2576  026250  142703  000360                BICB #360, R3
2577  026254                                    WHILE R3 NE #0 DO
  (4)  026254                               50242$:
  (6)  026254  005703                          TST     R3
  (9)  026256  001423                          BEQ     50243$
2578  026260                                        OUTPUT #CHARSP, #1
2579  026322                                        LET R3 := R3 - #1
  (6)  026322  005303                          DEC     R3
2580  026324                                    ENDDO
  (4)  026324  000753                          BR      50242$
  (3)  026326                               50243$:
2581  026326                                    OUTPUT #CHARX, #1
2582  026370                                    LET OUTBUF := #15
  (4)  026370  012737  000015  003172          MOV     #15,OUTBUF
2583  026376                                    OUTPUT #OUTBUF, #1.
```

```
2584  026440                          ENDDEC
 (4)  026440  000652                        BR      50240$
 (3)  026442                        50241$:
2585  026442  000137  025606                JMP     2$                      ;GO BACK AND TRY IT AGAIN
2586  026446                        4$:     LET OUTBUF := #14
 (4)  026446  012737  000014  003172        MOV     #14,OUTBUF
2587  026454                                OUTPUT #OUTBUF, #1
2588  026516                        EXIT TST
 (3)  026516  104432                        TRAP    C$EXIT
 (3)  026520  000150                        .WORD   L10020-.
2589
2590  026522  000000                  INSTR:  .WORD 0
2591  026524  040504  043126  020125  VFUCHL: .ASCIZ /DAVFU CHANNEL SELECTION PAPER CONTROL TEST 7/ <12><12><12>
      026532  044103  047101  042516
      026540  020114  042523  042514
      026546  052103  047511  020116
      026554  040520  042520  020122
      026562  047503  052116  047522
      026570  020114  042524  052123
      026576  033440  005012  000012
2592                                    .EVEN
2593  026604     040                  CHARSP: .BYTE 40
2594  026605     130                  CHARX:  .BYTE 130
2595
2596  026606     356     001     000  VFUDAT: .BYTE 356, 1, 0, 2, 0, 4, 0
      026611     002     000     004
      026614     000
2597  026615     010     000     020          .BYTE 10, 0, 20, 0, 40, 0, 0, 1
      026620     000     040     000
      026623     000     001
2598  026625     000     002     000          .BYTE 0, 2, 0, 4, 0, 10, 0, 20
      026630     004     000     010
      026633     000     020
2599  026635     000     040     000          .BYTE 0, 40, 0, 40, 0, 20, 0, 10
      026640     040     000     020
      026643     000     010
2600  026645     000     004     000          .BYTE 0, 4, 0, 2, 0, 1, 40, 0
      026650     002     000     001
      026653     040     000
2601  026655     020     000     010          .BYTE 20, 0, 10, 0, 4, 0, 2, 0
      026660     000     004     000
      026663     002     000
2602  026665     001     000     357          .BYTE 1, 0, 357
2603                                  .EVEN
2604  026670                          ENDTST
 (3)  026670                          L10020:
 (3)  026670  104401                        TRAP    C$ETST
2605
2606  026672                          ENDMOD
```

```
                                        DATA TRANSFER PATHS

                                        .SBTTL DATA TRANSFER PATHS

                                        BGNMOD
                                        ;++
                                        ;THIS TEST CHECKS THE DATA TRANSFER
                                        ;PATHS FROM THE PROCESSOR INTERFACE
                                        ;TO THE PRINTER OUTPUT. AN ALTERNATING
                                        ;PATTERN OF ONES AND ZEROES CORRESPONDING
                                        ;TO AN ALTERNATING STRING OF ''*'' AND
                                        ;''U'' CHARACTERS ARE TRANSMITTED ON THE
                                        ;FULL 132 COLUMNS.  AFTER 16 LINES OF
                                        ;THIS PATTERN, THE OUTPUT PATTERN IS
                                        ;SWITCHED TO AN ALTERNATING PATTERN
                                        ;OF ''?'' AND ''@'' CHARACTERS FOR ANOTHER
                                        ;16 LINES.
                                        ;--

                                        BGNTST  8.
  (3)  026672                           T8::
                                        ;PRINT TEST IDENTIFICATION
  026672                                OUTPUT #DATPTH,#30.
                                        ;PRINT ALTERNATING STRINGS OF CHARACTERS
  026734                                INCR PATTERN FROM #1 TO #2 BY #1
  (4)  026734  012737  000001  027300           MOV     #1,PATTERN
  (5)  026742  000402                           BR      50244$
  (4)  026744                           50245$:
  (7)  026744  005237  027300                   INC     PATTERN
  (5)  026750                           50244$:
  (5)  026750  023727  027300  000002           CMP     PATTERN,#2
  (7)  026756  003101                           BGT     50246$
  026760                                    IF PATTERN EQ #1 THEN
  (6)  026760  023727  027300  000001           CMP     PATTERN,#1
  (9)  026766  001004                           BNE     50247$
  026770                                            LET CHAR :B= #'U
  (4)  026770  112737  000125  027236           MOVB    #'U,CHAR
  026776                                        ELSE
  (4)  026776  000403                           BR      50250$
  (3)  027000                           50247$:
  027000                                            LET CHAR :B= #'?
  (4)  027000  112737  000077  027236           MOVB    #'?,CHAR
  027006                                    ENDIF
  (4)  027006                           50250$:
  027006                                    LET R4 := #OUTBUF
  (4)  027006  012704  003172                   MOV     #OUTBUF,R4
  027012                                    INCR CCNT FROM #1 TO #66. BY #1
  (4)  027012  012737  000001  002316           MOV     #1,CCNT
  (5)  027020  000402                           BR      50251$
  (4)  027022                           50252$:
  (7)  027022  005237  002316                   INC     CCNT
  (5)  027026                           50251$:
  (5)  027026  023727  002316  000102           CMP     CCNT,#66.
  (7)  027034  003011                           BGT     50253$
  027036                                            LET (R4)+ :B= CHAR
  (4)  027036  113724  027236                   MOVB    CHAR,(R4)+
  027042  105137  027236                           COMB        CHAR
  027046                                            LET (R4)+ :B= CHAR
```

```
  (4)  027046  113724  027236                    MOVB    CHAR,(R4)+
2640   027052  105137  027236                        COMB        CHAR
2641   027056                                    ENDINC
  (4)  027056  000761                            BR      50252$
  (3)  027060                              50253$:
2642   027060                                    LET (R4)+ :B= #15
  (4)  027060  112724  000015                    MOVB    #15,(R4)+
2643   027064                                    LET (R4) :B= #12
  (4)  027064  112714  000012                    MOVB    #12,(R4)
2644   027070                                    INCR LINCNT FROM #1 TO #16. BY #1
  (4)  027070  012737  000001  002310            MOV     #1,LINCNT
  (5)  027076  000402                            BR      50254$
  (4)  027100                              50255$:
  (7)  027100  005237  002310                    INC     LINCNT
  (5)  027104                              50254$:
  (5)  027104  023727  002310  000020            CMP     LINCNT,#16.
  (7)  027112  003022                            BGT     50256$
2645   027114                                        OUTPUT #OUTBUF, #134.
2646   027156                                    ENDINC
  (4)  027156  000750                            BR      50255$
  (3)  027160                              50256$:
2647   027160                              ENDINC
  (4)  027160  000671                            BR      50245$
  (3)  027162                              502[illegible]$:
2648   027162                              LET OUTBUF :B= #14
  (4)  027162  112737  000014  003172            MOVB    #14,OUTBUF
2649   027170                                    OUTPUT  #OUTBUF, #1
2650   027232                              EXIT TST
  (3)  027232  104432                            TRAP    C$EXIT
  (3)  027234  000046                            .WORD   L10021-.
2651
2652   027236  000000                      CHAR:   .WORD 0
2653   027240  042014  052101  020101      DATPTH: .ASCIZ <14>/DATA TRANSFER PATHS TEST 8/ <12><12><12>
       027246  051124  047101  043123
       027254  051105  050040  052101
       027262  051510  052040  051505
       027270  020124  005070  005012
       027276     000
2654
2655           027300                     .EVEN
2656   027300  000000                     PATTERN:        .WORD 0
2657                                                     .EVEN
2658                                      .EVEN
2659
2660   027302                             ENDTST
  (3)  027302                             L10021:
  (3)  027302  104401                            TRAP    C$ETST
2661
2662   027304                             ENDMOD
```

```
                                        .SBTTL PRINTABLE CHARACTERS
 2665  027304                           BGNMOD
                                        ;++
                                        ; THIS TEST WILL PRINT A FULL LINE OF EACH PRINTABLE CHARACTER.
                                        ; BAND TYPE IS CHECKED ON A UNIT BY UNIT BASIS.
                                        ; UNITS WITH 96 CHAR BAND WILL BE SENT THE CHARACTER CODES :
                                        ; 140(8) THRU 176(8).
                                        ;--

 2673  027304                           BGNTST  9.
  (3)  027304                           T9::
 2674  027304                           OUTPUT #PRTCHR, #30.            ; PRINT TEST ID
                                        ;
                                        ; PRINT ALL UPPER CASE CHARACTERS ON ALL UNITS
                                        ;
 2678  027346                                   INCR WORK FROM #40 TO #137 BY #1
  (4)  027346  012737  000040  003166           MOV     #40,WORK
  (5)  027354  000402                           BR      50257$
  (4)  027356                           50260$:
  (7)  027356  005237  003166                   INC     WORK
  (5)  027362                           50257$:
  (5)  027362  023727  003166  000137           CMP     WORK,#137
  (7)  027370  003045                           BGT     50261$
 2679  027372                                      LET R4 := #OUTBUF
  (4)  027372  012704  003172                   MOV     #OUTBUF,R4
 2680  027376                                    INCR COUNT FROM #1 TO #132. BY #1
  (4)  027376  012737  000001  002314           MOV     #1,COUNT
  (5)  027404  000402                           BR      50262$
  (4)  027406                           50263$:
  (7)  027406  005237  002314                   INC     COUNT
  (5)  027412                           50262$:
  (5)  027412  023727  002314  000204           CMP     COUNT,#132.
  (7)  027420  003003                           BGT     50264$
 2681  027422                                      LET (R4)+ :B= WORK
  (4)  027422  113724  003166                   MOVB    WORK,(R4)+
 2682  027426                                    ENDINC
  (4)  027426  000767                           BR      50263$
  (3)  027430                           50264$:
 2683  027430                                      LET (R4)+ :B= #LF
  (4)  027430  112724  000012                   MOVB    #LF,(R4)+
 2684  027434                                    OUTPUT #OUTBUF,#133.
 2685  027476  004737  005466                   JSR PC,QUIET
 2686  027502                                    ENDINC
  (4)  027502  000725                           BR      50260$
  (3)  027504                           50261$:
                                        ;
                                        ; NOW DO ALL THE LOWER CASE CHARACTERS ON THOSE UNITS
                                        ; EQUIPPED WITH 96 CHARACTER BANDS.
                                        ;

                                        ; FIRST DETERMIN IF ANY UNITS HAVE 96 CHAR BANDS
                                        ;
 2694  027504                           LET WORK := #0                  ; COUNTER FOR 96 CHAR UNITS
  (4)  027504  005037  003166                   CLR     WORK
 2695  027510                           LET WORK1 := L$UNIT - #1                ; GET UNIT COUNT
  (4)  027510  013737  002012  003170           MOV     L$UNIT,WORK1
```

```
  (6)  027516  005337  003170                     DEC     WORK1
 2696  027522                           INCR LUNIT FROM #0 TO WORK1 BY #1
  (4)  027522  005037  002326                     CLR     LUNIT
  (5)  027526  000402                             BR      50265$
  (4)  027530                           50266$:
  (7)  027530  005237  002326                     INC     LUNIT
  (5)  027534                           50265$:
  (5)  027534  023737  002326  003170             CMP     LUNIT,WORK1
  (7)  027542  003012                             BGT     50267$
 2697  027544                                     LET R2 := LUNIT SHIFT 1
  (4)  027544  013702  002326                     MOV     LUNIT,R2
  (7)  027550  006302                             ASL     R2
 2698  027552                                     IF #FLAG96 SETIN STATUS(R2) THEN    ; IS THIS UNIT 96 CHAR ?
  (6)  027552  032762  010000  002524             BIT     #FLAG96,STATUS(R2)
  (9)  027560  001402                             BEQ     50270$
 2699  027562                                             LET WORK := WORK + #1          ; YES ADD 1 TO COUNT
  (6)  027562  005237  003166                     INC     WORK
 2700  027566                                     ENDIF
  (4)  027566                           50270$:
 2701  027566                           ENDINC
  (4)  027566  000760                             BR      50266$
  (3)  027570                           50267$:
 2702  027570                           IF WORK EQ #0 THEN          ; ANY 96 CHAR UNITS ?
  (6)  027570  005737  003166                     TST     WORK
  (9)  027574  001002                             BNE     50271$
 2703  027576                                     EXIT TST              ; ALL UNITS 64 CHAR...EXIT
  (3)  027576  104432                             TRAP    C$EXIT
  (3)  027600  000330                             .WORD   L10022-.
 2704  027602                           ENDIF
  (4)  027602                           50271$:
 2705                                   ; SETUP FOR LOWER CASE CHARACTERS DISPLAY
 2706                                   ;
 2707  027602                                     INCR WORK FROM #140 TO #176 BY #1
  (4)  027602  012737  000140  003166             MOV     #140,WORK
  (5)  027610  000402                             BR      50272$
  (4)  027612                           50273$:
  (7)  027612  005237  003166                     INC     WORK
  (5)  027616                           50272$:
  (5)  027616  023727  003166  000176             CMP     WORK,#176
  (7)  027624  003073                             BGT     50274$
 2708  027626                                         LET R4 := #OUTBUF
  (4)  027626  012704  003172                     MOV     #OUTBUF,R4
 2709  027632                                         INCR COUNT FROM #1 TO #132. BY #1
  (4)  027632  012737  000001  002314             MOV     #1,COUNT
  (5)  027640  000402                             BR      50275$
  (4)  027642                           50276$:
  (7)  027642  005237  002314                     INC     COUNT
  (5)  027646                           50275$:
  (5)  027646  023727  002314  000204             CMP     COUNT,#132.
  (7)  027654  003003                             BGT     50277$
 2710  027656                                             LET (R4)+ :B= WORK
  (4)  027656  113724  003166                     MOVB    WORK,(R4)+
 2711  027662                                         ENDINC
  (4)  027662  000767                             BR      50276$
  (3)  027664                           50277$:
 2712  027664                                         LET (R4)+ :B= #LF
```

```
  (4)  027664  112724  000012                    MOVB    #LF,(R4)+
2713   027670                                        LET WORK1 := L$UNIT - #1
  (4)  027670  013737  002012  003170              MOV     L$UNIT,WORK1
  (6)  027676  005337  003170                      DEC     WORK1
2714   027702                                        INCR LUNIT FROM #0 TO WORK1 BY #1
  (4)  027702  005037  002326                      CLR     LUNIT
  (5)  027706  000402                              BR      50300$
  (4)  027710                              50301$:
  (7)  027710  005237  002326                      INC     LUNIT
  (5)  027714                              50300$:
  (5)  027714  023737  002326  003170              CMP     LUNIT,WORK1
  (7)  027722  003031                              BGT     50302$
2715   027724                                        LET R2 := LUNIT SHIFT 1
  (4)  027724  013702  002326                      MOV     LUNIT,R2
  (7)  027730  006302                              ASL     R2
2716   027732                                        IF #FLAG96 SETIN STATUS(R2) THEN
  (6)  027732  032762  010000  002524              BIT     #FLAG96,STATUS(R2)
  (9)  027740  001421                              BEQ     50303$
2717   027742                                            OUTPUI #OUTBUF,#133.,,LUNIT
2718   030004                                        ENDIF
  (4)  030004                              50303$:
2719   030004                                    ENDINC
  (4)  030004  000741                              BR      50301$
  (3)  030006                              50302$:
2720   030006  004737  005466                          JSR PC,QUIET        ; WAIT FOR ALL DONE
2721   030012                                    ENDINC
  (4)  030012  000677                              BR      50273$
  (3)  030014                              50274$:
2722   030014                                LET OUTBUF :B= #FF
  (4)  030014  112737  000014  003172              MOVB    #FF,OUTBUF
2723   030022                                OUTPUT #OUTBUF,#1   ; EXECUTE TOF
2724   030064                              EXIT TST
  (3)  030064  104432                              TRAP    C$EXIT
  (3)  030066  000042                              .WORD   L10022-.
2725
2726   030070  051120  047111  040524  PRTCHR: .ASCIZ /PRINTABLE CHARACTERS TEST 9/ <12><12><12>
       030076  046102  020105  044103
       030104  051101  041501  042524
       030112  051522  052040  051505
       030120  020124  005071  005012
       030126     000
2727           030130                             .EVEN
2728
2729   030130                              ENDTST
  (3)  030130                              L10022:
  (3)  030130  104401                              TRAP    C$ETST
2730
2731   030132                              ENDMOD
2732
```

```
                                        .SBTTL  NON-PRINTABLE CHARACTERS
                                        
030132                                  BGNMOD
                                        ;++
                                        ;THIS TEST CHECKS FOR DETECTION OF ALL NON-PRINTABLE CHARACTERS.
                                        ;EACH CHARACTER WILL APPEAR ON THE PRINTER OUTPUT IN THE FORM OF ITS OCTAL
                                        ;CODE ACCOMPANIED WITH ITS MNEMONIC.
                                        ; 123 OF THE TESTED CODE ARE THEN SENT.
                                        ;--
                                        
030132                                  BGNTST  10.
030132                                  T10::
                                        ;INDICATE TEST CURRENTLY BEING DONE
                                        
030132                                          OUTPUT #NONCHR,#71.
030174                                          LET R4 := #NONBUF
030174  012704  031065                          MOV     #NONBUF,R4
030200                                          LET WORK1 := #27.
030200  012737  000033  003170                  MOV     #27.,WORK1
                                        ;
                                        ; DO ONE LINE FOR EACH TABLE ENTRY
                                        ;
030206                                          INCR LINCNT FROM #0 TO WORK1 BY #1
030206  005037  002310                          CLR     LINCNT
030212  000402                                  BR      50304$
030214                                  50305$:
030214  005237  002310                          INC     LINCNT
030220                                  50304$:
030220  023737  002310  003170                  CMP     LINCNT,WORK1
030226  003061                                  BGT     50306$
030230                                              LET R3 := #OUTBUF
030230  012703  003172                          MOV     #OUTBUF,R3
                                        ;
                                        ; MOVE CODE AND MNEMONIC TO PRINT BUFFER
                                        ;
030234                                              INCR WORK FROM #1 TO #8. BY #1
030234  012737  000001  003166                  MOV     #1,WORK
030242  000402                                  BR      50307$
030244                                  50310$:
030244  005237  003166                          INC     WORK
030250                                  50307$:
030250  023727  003166  000010                  CMP     WORK,#8.
030256  003002                                  BGT     50311$
030260                                              LET (R3)+ :B= (R4)+
030260  112423                                  MOVB    (R4)+,(R3)+
030262                                          ENDINC
030262  000770                                  BR      50310$
030264                                  50311$:
                                        
                                        ;
                                        ;  PUT 120 BYTES OF CODE INTO PRINT BUFFER
                                        ;
030264                                              INCR WORK FROM #1 TO #123. BY #1
030264  012737  000001  003166                  MOV     #1,WORK
030272  000402                                  BR      50312$
030274                                  50313$:
```

```
   (7)  030274  005237  003166                  INC     WORK
   (5)  030300                          50312$:
   (5)  030300  023727  003166  000173          CMP     WORK,#123.
   (7)  030306  003002                          BGT     50314$
  2766  030310                                      LET (R3)+ :B= (R4)
   (4)  030310  111423                          MOVB    (R4),(R3)+
  2767  030312                                  ENDINC
   (4)  030312  000770                          BR      50313$
   (3)  030314                          50314$:
  2768
  2769                                 ;
  2770                                 ; FOLLOWED BY CRLF
  2771                                 ;
  2772  030314                                      LET (R3)+ :B= #15
   (4)  030314  112723  000015                  MOVB    #15,(R3)+
  2773  030320                                      LET (R3)+ :B= #12
   (4)  030320  112723  000012                  MOVB    #12,(R3)+
  2774                                 ;
  2775                                 ; PRINT LINE OF OCTAL CODE, MNEMONIC, AND 120 BYTES(NONPRINTABLE CODE)
  2776                                 ;
  2777  030324                                      OUTPUT  #OUTBUF,#133.
  2778  030366                                      LET R4 := R4 + #1
   (6)  030366  005204                          INC     R4
  2779  030370                                  ENDINC
   (4)  030370  000711                          BR      50305$
   (3)  030372                          50306$:
  2780                                 ;
  2781                                 ; UNITS WITH 64 CHAR BAND SHOULD STRIP BIT 6 OF DATA
  2782                                 ; AND PRINT THE DATA FOR CODES 140(8) THRU 177(8)
  2783                                 ; AS IF CODES 100(8) THRU 137(8) WERE RECIEVED.
  2784                                 ; **NOTE**     DELETE IS PRINTED AS UNDERSCORE '_'
  2785                                 ;
  2786  030372                                  LET R3 := #OUTBUF
   (4)  030372  012703  003172                  MOV     #OUTBUF,R3
  2787  030376                                  INCR R4 FROM #140 TO #177 BY #1
   (4)  030376  012704  000140                  MOV     #140,R4
   (5)  030402  000401                          BR      50315$
   (4)  030404                          50316$:
   (7)  030404  005204                          INC     R4
   (5)  030406                          50315$:
   (5)  030406  020427  000177                  CMP     R4,#177
   (7)  030412  003002                          BGT     50317$
  2788  030414                                      LET (R3)+ :B= R4    ; FILL BUFFER WITH CODES & LF
   (4)  030414  110423                          MOVB    R4,(R3)+
  2789  030416                                  ENDINCR
   (4)  030416  000772                          BR      50316$
   (3)  030420                          50317$:
  2790  030420                                  LET (R3)+ :B= #LF
   (4)  030420  112723  000012                  MOVB    #LF,(R3)+
  2791  030424                                  LET WORK := L$UNIT - #1         ; SEND MSG AND BUFFER TO ALL
   (4)  030424  013737  002012  003166          MOV     L$UNIT,WORK
   (6)  030432  005337  003166                  DEC     WORK
  2792  030436                                  INCR LUNIT FROM #0 TO WORK BY #1   ; UNITS WITH 64 CHAR BAND
   (4)  030436  005037  002326                  CLR     LUNIT
   (5)  030442  000402                          BR      50320$
   (4)  030444                          50321$:
```

```
  (7)  030444  005237  002326                    INC     LUNIT
  (5)  030450                              50320$:
  (5)  030450  023737  002326  003166            CMP     LUNIT,WORK
  (7)  030456  003052                            BGT     50322$
2793   030460                                        LET R2 := LUNIT SHIFT #1
  (4)  030460  013702  002326                    MOV     LUNIT,R2
  (7)  030464  006302                            ASL     R2
2794   030466                                        IF #FLAG96 NOTSETIN STATUS(R2) THEN
  (6)  030466  032762  010000  002524            BIT     #FLAG96,STATUS(R2)
  (9)  030474  001042                            BNE     50323$
2795   030476                                            OUTPUI #AUTCON,#61.,,LUNIT
2796   030540                                            OUTPUI #OUTBUF,#33.,,LUNIT
2797   030602                                        ENDIF
  (4)  030602                              50323$:
2798   030602                                    ENDINCR
  (4)  030602  000720                            BR      50321$
  (3)  030604                              50322$:
2799   030604                                    OUTPUT  #SKIP3,#4
2800   030646                                    EXIT    TST                     ;AND EXIT TEST
  (3)  030646  104432                            TRAP    C$EXIT
  (3)  030650  000612                            .WORD   L10023-.
2801                                       ;
2802                                       ;CHARACTER BUFFER AND TEST HEADER MESSAGE
2803                                       ;
2804                                       .NLIST BEX
2805   030652  047516  026516  051120     NONCHR: .ASCII  /NON-PRINTABLE CHARACTERS TEST 10/<12>
2806   030713     101  043040  046125             .ASCIZ  /A FULL LINE OF EACH CODE WILL BE SENT/<12>
2807   030762  047503  042504  020123     AUTCON: .ASCIZ  /CODES 140(8) ..177(8) SHOULD BE CONVERTED TO 100(8)..137(8)/<12><12>
2808   031060  005015  005012     000     SKIP3:  .ASCIZ  <15><12><12><12>
2809
2810   031065     040  030060  020060     NONBUF: .ASCII  / 000 NUL/<0>
2811   031076  030040  030460  051440             .ASCII  / 001 SOH/<1>
2812   031107     040  030060  020062             .ASCII  / 002 STX/<2>
2813   031120  030040  031460  042440             .ASCII  / 003 ETX/<3>
2814   031131     040  030060  020064             .ASCII  / 004 EOT/<4>
2815   031142  030040  032460  042440             .ASCII  / 005 ENQ/<5>
2816   031153     040  030060  020066             .ASCII  / 006 ACK/<6>
2817   031164  030040  033460  041040             .ASCII  / 007 BEL/<7>
2818   031175     040  030460  020060             .ASCII  / 010 BS /<10>
2819   031206  030040  030461  044040             .ASCII  / 011 HT /<11>
2820   031217     040  030460  020066             .ASCII  / 016 SO /<16>
2821   031230  030040  033461  051440             .ASCII  / 017 SI /<17>
2822   031241     040  031060  020060             .ASCII  / 020 DLE/<20>
2823   031252  030040  030462  054040             .ASCII  / 021 XON/<21>
2824   031263     040  031060  020062             .ASCII  / 022 DC2/<22>
2825   031274  030040  031462  054040             .ASCII  / 023 XOF/<23>
2826   031305     040  031060  020064             .ASCII  / 024 DC4/<24>
2827   031316  030040  032462  047040             .ASCII  / 025 NAK/<25>
2828   031327     040  031060  020066             .ASCII  / 026 SYN/<26>
2829   031340  030040  033462  042440             .ASCII  / 027 ETB/<27>
2830   031351     040  031460  020060             .ASCII  / 030 CAN/<30>
2831   031362  030040  030463  042440             .ASCII  / 031 EM /<31>
2832   031373     040  031460  020062             .ASCII  / 032 SUB/<32>
2833   031404  030040  031463  042440             .ASCII  / 033 ESC/<33>
2834   031415     040  031460  020064             .ASCII  / 034 FS /<34>
2835   031426  030040  032463  043440             .ASCII  / 035 GS /<35>
```

```
 2836  031437     040  031460  020066          .ASCII  / 036 RS /<36>
 2837  031450  030040  033463  052440          .ASCII  / 037 US /<37>
 2838          031462                          .EVEN
 2839
 2840                                  .LIST BEX
 2841  031462                          ENDTST
  (3)  031462                          L10023:
  (3)  031462  104401                             TRAP    C$ETST
 2842
 2843  031464                          ENDMOD
 2844
```

```
                                    .SBTTL  BAND PATTERN
031464                              BGNMOD
                                    ;++
                                    ;BAND PATTERN TEST
                                    ;
                                    ;THIS TEST PRODUCES AN IMAGE OF THE ENTIRE BAND PATTERN. THE PRINT-OUT
                                    ;IS ORGANIZED TO LOCATE THE FOUR  QUADRANTS OF THE BAND IN THE FOLLOWING
                                    ;FORMAT:
                                    ;
                                    ;       QUADRANT NO.1           QUADRANT NO. 2
                                    ;       QUADRANT NO.3           QUADRANT NO.4
                                    ;
                                    ;       QUADRANT NO.1           ETC.
                                    ;
                                    ;THE REASON FOR THIS ARRANGEMENT IS TO FACILITATE VISUAL INSPECTION
                                    ;OF THE PRINTOUT AS WELL AS TO ACCOMODATE THE 208 CHARACTERS OF THE BAND
                                    ;IN 132 COLUMNS.
                                    ;
                                    ;       STATUS(R2) FLAG96       =1 FOR 96 CHAR BAND
                                    ;                               =0 FOR 64 CHAR BAND
                                    ;       SOFTWARE SWITCH USA    =1 FOR AMERICAN PRINT SET
                                    ;                              =0 FOR BRITISH  PRINT SET
                                    ;
                                    ;--
```

```
 2871  031464                          BGNTST  11.
  (3)  031464                          T11::
 2872          000014                  TOF = 014
 2873                                  ;
 2874                                  ; PRINT TEST IDENTIFICATION ON ALL UNITS
 2875                                  ;
 2876  031464                                OUTPUT  #BNDTST,#23.
 2877                                  ;
 2878                                  ; SETUP PATTERNS FOR EUROPEAN OR AMERICAN PRINTERS
 2879                                  ;
 2880  031526                                IF USA NE #0 THEN       ; AMERICAN, PRINT SHARP SIGN '#'
  (6)  031526  005737  002302                TST     USA
  (9)  031532  001432                        BEQ     50324$
 2881  031534                                   LET BP64Q2+18. :B= #0
  (4)  031534  105037  033143                CLRB    BP64Q2+18.
 2882  031540                                   LET BP64Q2+22. :B= #43
  (4)  031540  112737  000043  033147        MOVB    #43,BP64Q2+22.
 2883  031546                                   LET BP64Q3+35. :B= #0
  (4)  031546  105037  033261                CLRB    BP64Q3+35.
 2884  031552                                   LET BP64Q3+40. :B= 43
  (4)  031552  113737  000043  033266        MOVB    43,BP64Q3+40.
 2885  031560                                   LET BP64Q4+53. :B= #0
  (4)  031560  105037  033377                CLRB    BP64Q4+53.
 2886  031564                                   LET BP64Q4+57. :B= #43
  (4)  031564  112737  000043  033403        MOVB    #43,BP64Q4+57.
 2887  031572                                   LET BP96Q2+13. :B= #0
  (4)  031572  105037  033521                CLRB    BP96Q2+13.
 2888  031576                                   LET BP96Q2+18. :B= #43
  (4)  031576  112737  000043  033526        MOVB    #43,BP96Q2+18.
 2889  031604                                   LET BP96Q4+13. :B= #0
  (4)  031604  105037  033712                CLRB    BP96Q4+13.
 2890  031610                                   LET BP96Q4+18. :B= #43
  (4)  031610  112737  000043  033717        MOVB    #43,BP96Q4+18.
 2891  031616                                ELSE                    ; EUROPEAN, PRINT POUND STERLING SIGN
  (4)  031616  000431                        BR      50325$
  (3)  031620                          50324$:
 2892  031620                                   LET BP64Q2+18. :B= #43
  (4)  031620  112737  000043  033143        MOVB    #43,BP64Q2+18.
 2893  031626                                   LET BP64Q2+22. :B= #0
  (4)  031626  105037  033147                CLRB    BP64Q2+22.
 2894  031632                                   LET BP64Q3+35. :B= #43
  (4)  031632  112737  000043  033261        MOVB    #43,BP64Q3+35.
 2895  031640                                   LET BP64Q3+40. :B= #0
  (4)  031640  105037  033266                CLRB    BP64Q3+40.
 2896  031644                                   LET BP64Q4+53. :B= #43
  (4)  031644  112737  000043  033377        MOVB    #43,BP64Q4+53.
 2897  031652                                   LET BP64Q4+57. :B= #0
  (4)  031652  105037  033403                CLRB    BP64Q4+57.
 2898  031656                                   LET BP96Q2+13. :B= #43
  (4)  031656  112737  000043  033521        MOVB    #43,BP96Q2+13.
 2899  031664                                   LET BP96Q2+18. :B= #0
  (4)  031664  105037  033526                CLRB    BP96Q2+18.
 2900  031670                                   LET BP96Q4+13. :B= #43
  (4)  031670  112737  000043  033712        MOVB    #43,BP96Q4+13.
 2901  031676                                   LET BP96Q4+18. :B= #0
  (4)  031676  105037  033717                CLRB    BP96Q4+18.
```

```
2902  031702                                          ENDIF
 (4)  031702                                   50325$:
2903                                           ;
2904                                           ; PRINT PROPER BAND IDENTIFICATION MSG. ON EACH PRINTER
2905                                           ;
2906  031702                                          LET R1 := L$UNIT - #1
 (4)  031702  013701  002012                       MOV     L$UNIT,R1
 (6)  031706  005301                               DEC     R1
2907  031710                                          INCR LUNIT FROM #0 TO R1 BY #1
 (4)  031710  005037  002326                       CLR     LUNIT
 (5)  031714  000402                               BR      50326$
 (4)  031716                                   50327$:
 (7)  031716  005237  002326                       INC     LUNIT
 (5)  031722                                   50326$:
 (5)  031722  023701  002326                       CMP     LUNIT,R1
 (7)  031726  003135                               BGT     50330$
2908  031730                                          LET R2 := LUNIT SHIFT 1
 (4)  031730  013702  002326                       MOV     LUNIT,R2
 (7)  031734  006302                               ASL     R2
2909  031736                                          IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG27 NOTSETIN STATUS(R2) THEN
 (6)  031736  032762  001000  002524               BIT     #FLAG26,STATUS(R2)
 (9)  031744  001025                               BNE     50331$
 (6)  031746  032762  002000  002524               BIT     #FLAG27,STATUS(R2)
 (9)  031754  001021                               BNE     50331$
2910  031756                                              OUTPUI  #BPID25,#6,,LUNIT       ; PRINTER IS LP25
2911  032020                                          ENDIF
 (4)  032020                                   50331$:
2912  032020                                          IF #FLAG26 SETIN STATUS(R2) AND #FLAG27 NOTSETIN STATUS(R2) THEN
 (6)  032020  032762  001000  002524               BIT     #FLAG26,STATUS(R2)
 (9)  032026  001425                               BEQ     50332$
 (6)  032030  032762  002000  002524               BIT     #FLAG27,STATUS(R2)
 (9)  032036  001021                               BNE     50332$
2913  032040                                              OUTPUI  #BPID26,#6,,LUNIT       ; PRINTER IS LP26
2914  032102                                          ENDIF
 (4)  032102                                   50332$:
2915  032102                                          IF #FLAG96 SETIN STATUS(R2) THEN
 (6)  032102  032762  010000  002524               BIT     #FLAG96,STATUS(R2)
 (9)  032110  001422                               BEQ     50333$
2916  032112                                              OUTPUI  #BP96ID,#22.,,LUNIT     ; 96 CHAR BAND
2917  032154                                          ELSE
 (4)  032154  000421                               BR      50334$
 (3)  032156                                   50333$:
2918  032156                                              OUTPUI  #BP64ID,#22.,,LUNIT     ; 64 CHAR BAND
2919  032220                                          ENDIF
 (4)  032220                                   50334$:
2920  032220                                          ENDINC
 (4)  032220  000636                               BR      50327$
 (3)  032222                                   50330$:
2921                                           ;
2922                                           ; NOW PRINT 2_LINE PATTERN 15. TIMES, WITH BLANK LINE BETWEEN PATTERNS
2923                                           ;
2924  032222                                          LET LINCNT := #14.
 (4)  032222  012737  000016  002310               MOV     #14.,LINCNT
2925  032230                                   2$:
2926  032230                                          INCR LUNIT FROM #0 TO R1 BY #1      ; PRINT QUADRANTS 1 & 2
 (4)  032230  005037  002326                       CLR     LUNIT
```

```
   (5)  032234  000402                          BR      50335$
   (4)  032236                            50336$:
   (7)  032236  005237  002326                INC     LUNIT
   (5)  032242                            50335$:
   (5)  032242  023701  002326                CMP     LUNIT,R1
   (7)  032246  003101                        BGT     50337$
  2927  032250                                LET R2 := LUNIT SHIFT 1          ; INDEX INTO STATUS TABLES
   (4)  032250  013702  002326                MOV     LUNIT,R2
   (7)  032254  006302                        ASL     R2
  2928  032256                                IF #FLAG96 NOTSETIN STATUS(R2) THEN
   (6)  032256  032762  010000  002524        BIT     #FLAG96,STATUS(R2)
   (9)  032264  001022                        BNE     50340$
  2929  032266                                    OUTPUI  #BP64Q1,#121.,,LUNIT         ; 64 CHAR PATTERN
  2930  032330                                ELSE
   (4)  032330  000447                        BR      50341$
   (3)  032332                            50340$:
  2931  032332                                    IF #FLAG26!FLAG27 NOTSETIN STATUS(R2) THEN
   (6)  032332  032762  003000  002524        BIT     #FLAG26!FLAG27,STATUS(R2)
   (9)  032340  001022                        BNE     50342$
  2932  032342                                    OUTPUI  #BP96Q3,#121.,,LUNIT     ; LP25 96 CHAR PATTERN
  2933  032404                                ELSE
   (4)  032404  000421                        BR      50343$
   (3)  032406                            50342$:
  2934  032406                                    OUTPUI  #BP96Q1,#121.,,LUNIT     ; LP26 96 CHAR PATTERN
  2935  032450                                ENDIF
   (4)  032450                            50343$:
  2936  032450                                ENDIF
   (4)  032450                            50341$:
  2937  032450                            ENDINC
   (4)  032450  000672                        BR      50336$
   (3)  032452                            50337$:
  2938                                      ; NOW DO QUADRANTS 3 & 4
  2939  032452                            INCR LUNIT FROM #0 TO R1 BY #1           ; REPEAT FOR ALL UNITS
   (4)  032452  005037  002326                CLR     LUNIT
   (5)  032456  000402                        BR      50344$
   (4)  032460                            50345$:
   (7)  032460  005237  002326                INC     LUNIT
   (5)  032464                            50344$:
   (5)  032464  023701  002326                CMP     LUNIT,R1
   (7)  032470  003053                        BGT     50346$
  2940  032472                                LET R2 := LUNIT SHIFT 1          ; INDEX INTO STATUS TABLES
   (4)  032472  013702  002326                MOV     LUNIT,R2
   (7)  032476  006302                        ASL     R2
  2941  032500                                IF #FLAG96 NOTSETIN STATUS(R2) THEN
   (6)  032500  032762  010000  002524        BIT     #FLAG96,STATUS(R2)
   (9)  032506  001022                        BNE     50347$
  2942  032510                                    OUTPUI  #BP64Q3,#122.,,LUNIT         ; 64 CHAR PATTERN
  2943  032552                                ELSE
   (4)  032552  000421                        BR      50350$
   (3)  032554                            50347$:
  2944  032554                                    OUTPUI  #BP96Q3,#122.,,LUNIT         ; 96 CHAR PATTERN
  2945  032616                                ENDIF
   (4)  032616                            50350$:
  2946  032616                            ENDINC
   (4)  032616  000720                        BR      50345$
   (3)  032620                            50346$:
```

```
2947   032620                                        LET LINCNT := LINCNT - #1
 (6)   032620  005337  002310                       DEC     LINCNT
2948   032624  001402                               BEQ     3$
2949   032626  000137  032230                       JMP     2$
2950   032632                                3$:
2951                                         ;
2952                                         ; DO TOF THEN EXIT
2953                                         ;
2954   032632                                        LET OUTBUF :B= #TOF
 (4)   032632  112737  000014  003172               MOVB    #TOF,OUTBUF
2955   032640                                        OUTPUT #OUTBUF,#1
2956   032702                                EXIT TST
 (3)   032702  104432                               TRAP    C$EXIT
 (3)   032704  001070                               .WORD   L10024-.
2957   032706  040502  042116  050040   BNDTST: .ASCII  /BAND PATTERN TEST_11   /
       032714  052101  042524  047122
       032722  052040  051505  057524
       032730  030461  020040     040
2958   032735     114  031120  020065   BPID25: .ASCII  /LP25  /
       032742     040
2959   032743     114  031120  020066   BPID26: .ASCII  /LP26  /
       032750     040
2960   032751     066  020064  044103   BP64ID: .ASCII  /64 CHAR BAND PATTERN/<12><12>
       032756  051101  041040  047101
       032764  020104  040520  052124
       032772  051105  005116     012
2961   032777     071  020066  044103   BP96ID: .ASCII  /96 CHAR BAND PATTERN/<12><12>
       033004  051101  041040  047101
       033012  020104  040520  052124
       033020  051105  005116     012
2962
2963                                         ; 64 CHAR BAND PATTERN  LP25 & LP26
2964                                         ;
2965   033025     040  020040  020040   BP64Q1: .ASCII  /        /
       033032  020040     040
2966   033035     101  042502  042103            .ASCII  /ABECDTFGOHIJ1KL2MN3OP4QRS5/
       033042  043124  030107  044510
       033050  030512  046113  046462
       033056  031516  050117  050464
       033064  051522     065
2967   033067     125  033126  054127            .ASCII  /UV6WX7YZ8_+%9&)*(:,/<57>/$-;><./
       033074  054467  034132  025537
       033102  034445  024446  024052
       033110  026072  022057  035455
       033116  036076     056
2968
2969   033121     040  020040  020040   BP64Q2: .ASCII  /        /
       033126  020040     040
2970   033131     077  042500  055535            .ASCII  /?@E][T\''C^#=1!#2'A3BC4DFG5/
       033136  056124  030042  021536
       033144  030475  021441  023462
       033152  031501  041502  042064
       033160  043506     065
2971   033163     110  033111  045512            .ASCII  /HI6JK7LM8NOP9QR*SU,VW-XYZ./<12>
       033170  046067  034115  047516
       033176  034520  051121  051452
```

```
        033204  026125  053526  054055
        033212  055131  005056
 2972
 2973   033216  020040  020040  020040  BP64Q3: .ASCII  /        /
        033224  020040
 2974   033226  025537  022505  052046          .ASCII  /_+E%&T)(0:/<57>/$1;>2<?3@]4[\''5/
        033234  024051  035060  022057
        033242  035461  031076  037474
        033250  040063  032135  056133
        033256  032442
 2975   033260  021536  036466  033441          .ASCII  /^#6=!7#'8ABC9DF*GH,IJ-KLM./
        033266  023443  040470  041502
        033274  042071  025106  044107
        033302  044454  026512  046113
        033310  027115
 2976
 2977   033312  020040  020040  020040  BP64Q4: .ASCII  /        /
        033320  020040
 2978   033322  047516  050105  052121          .ASCII  /NOEPQTRSOUVW1XY2Z_3+%4&)(5/
        033330  051522  052460  053526
        033336  054061  031131  057532
        033344  025463  032045  024446
        033352  032450
 2979   033354  027472  022066  033473          .ASCII  _:/6%;7><8?@]9[\*''^,#=-!#'._<12><12>
        033362  036076  037470  056500
        033370  055471  025134  057042
        033376  021454  026475  021441
        033404  027047  005012
 2980                                   ;
 2981                                   ; 96 CHAR BAND     LP25 = Q3..Q4     LP26 = Q1..Q4
 2982                                   ;
 2983   033410  020040  020040  020040  BP96Q1: .ASCII  /        /
        033416  020040
 2984   033420  041101  042103  042460          .ASCII  /ABCDOEFG1HIJ2KLMN3OPQ4RST5/
        033426  043506  044061  045111
        033434  045462  046514  031516
        033442  050117  032121  051522
        033450  032524
 2985   033452  053125  054127  054466          .ASCII  /UVWX6YZ_7$%&8)(:/<57>/9+;>/
        033460  057532  022067  023045
        033466  024470  035050  034457
        033474  035453     076
 2986   033477     145                          .BYTE   145
 2987   033500  037474     100                  .ASCII  /<?@/
 2988   033503     164                          .BYTE   164
 2989
 2990   033504  020040  020040  020040  BP96Q2: .ASCII  /        /
        033512  020040
 2991   033514  055535  021134  021454          .ASCII  /][\'',#/
 2992   033522     174                          .BYTE   174
 2993   033523     075  020452  023443          .ASCII  /=*:#'-/
        033530     055
 2994   033531     175     173     176          .BYTE   175,173,176,136,56,177,140,141,60,142,143,144,61
        033534     136     056     177
        033537     140     141     060
        033542     142     143     144
```

```
                033545      061
 2995           033546      146       147       150   .BYTE   146,147,150,151,62,152,153,154,63,155,156
                033551      151       062       152
                033554      153       154       063
                033557      155       156
 2996           033561      157       064       160           .BYTE   157,64,160,161,162,163,65,165,166,167,66
                033564      161       162       163
                033567      065       165       166
                033572      167       066
 2997           033574      170       171       172           .BYTE   170,171,172,67,12
                033577      067       012
 2998
 2999           033601       040   020040    020040   BP96Q3: .ASCII  /       /
                033606    020040       040
 3000           033611       101   041502    034104           .ASCII  /ABCD8EFG9HIJ/
                033616    043105   034507    044510
                033624       112
 3001           033625       145                              .BYTE   145
 3002           033626    046113   047115                     .ASCII  /KLMN/
 3003           033632       164                              .BYTE   164
 3004           033633       117   050520    051054           .ASCII  /OPQ,RST*/
                033640    052123       052
 3005           033643       125   053526    026530           .ASCII  /UVWX-YZ_.$%&0)(:/<57>/1+;>2<?@3/
                033650    055131   027137    022444
                033656    030046   024051    027472
                033664    025461   037073    036062
                033672    040077       063
 3006
 3007           033675       040   020040    020040   BP96Q4: .ASCII  /       /
                033702    020040       040
 3008           033705       135   056133    032042           .ASCII  /][\''4#/
                033712       043
 3009           033713       174                              .BYTE   174
 3010           033714    032475   021441    033047           .ASCII  /=5!#'6/
 3011           033722       175       173       176           .BYTE   175,173,176,136,67,177,140,141,70,142,143,144,71
                033725       136       067       177
                033730       140       141       070
                033733       142       143       144
                033736       071
 3012           033737       146       147       150           .BYTE   146,147,150,151,145,152,153,154,164,155,156,157
                033742       151       145       152
                033745       153       154       164
                033750       155       156       157
 3013           033753       054       160       161           .BYTE   54,160,161,162,163,52,165,166,167,55,170
                033756       162       163       052
                033761       165       166       167
                033764       055       170
 3014           033766       171       172       056           .BYTE   171,172,56,12,12
                033771       012       012
 3015
 3016                    033774                       .EVEN
 3017           033774                                ENDTST
  (3)           033774                                L10024:
  (3)           033774   104401                                TRAP    C$ETST
 3018           033776                                ENDMOD
```

```
                                        .SBTTL SPURIOUS HAMMER FIRING

 033776                                 BGNMOD

                                        ;++
                                        ;THE PURPOSE OF THIS TEST IS TO DETECT SPURIOUS HAMMER FIRINGS AND DEFECTIVE
                                        ;HAMMER DRIVERS DURING THE OPERATION OF THE LINE PRINTER. THE PROGRAM
                                        ;PRODUCES A LEFT WEDGE PATTERN CONSISTING OF 132 LINES OF PRINT WITH EACH
                                        ;LINE BEGINNING WITH A "?" CHARACTER. ANY POINT OUTSIDE THE WEDGE
                                        ;BOUNDARIES IS CAUSED BY HAMMER MISFIRES OR BY HAMMER BOUNCE.
                                        ;--

 033776                                 BGNTST  12.
 033776                                 T12::

                                        ;PRINT THE TEST HEADER

 033776                                 OUTPUT #HAMFIR,#33.

                                        ;OUTPUT THE ACTUAL WEDGE AT THIS POINT

 034040                                 INCR WORK FROM #1 TO #132. BY #1                ;NUMBER OF LINES TO OUTPUT
 034040  012737  000001  003166                 MOV     #1,WORK
 034046  000402                               BR      50351$
 034050                                 50352$:
 034050  005237  003166                       INC     WORK
 034054                                 50351$:
 034054  023727  003166  000204               CMP     WORK,#132.
 034062  003123                               BGT     50353$
                                                                ;ALSO NUMBER OF PRINTING CHARACTERS

 034064                                     LET R4 := #OUTBUF                          ;OUTPUT BUFFER POINTER
 034064  012704  003172                         MOV     #OUTBUF,R4
 034070                                     LET SPCCNT := #132. - WORK                 ;NUMBER OF SPACES TO FILL IN
 034070  012737  000204  034400                 MOV     #132.,SPCCNT
 034076  163737  003166  034400                 SUB     WORK,SPCCNT

                                        ;FILL THE OUTPUT BUFFER WITH THE REQUIRED NUMBER OF SPACES

                                        50354$:
                                            WHILE SPCCNT NE #0 DO
 034104                                 
 034104                                 
 034104  005737  034400                       TST     SPCCNT
 034110  001405                               BEQ     50355$
 034112                                     LET (R4)+ :B= #40                          ;SPACE FILL
 034112  112724  000040                       MOVB    #40,(R4)+
 034116                                     LET SPCCNT := SPCCNT - #1                  ;UPDATE FILLER COUNTER
 034116  005337  034400                       DEC     SPCCNT
 034122                                     ENDDO
 034122  000770                               BR      50354$
 034124                                 50355$:
 034124                                     LET CCNT := #0
 034124  005037  002316                       CLR     CCNT
 034130                                     LET CHRGEN := #77                          ;FIRST CHARACTER A "?"
 034130  012737  000077  002322                 MOV     #77,CHRGEN
 034136                                     LET STRCNT := #33.                         ;# OF CHARACTERS IN GROUP
 034136  012737  000041  002320                 MOV     #33.,STRCNT
```

```
3055  034144                                  WHILE CCNT LT WORK DO                ;NOW FILL IN REST OF BUFFER
 (4)  034144                                  50356$:
 (6)  034144  023737  002316  003166                CMP     CCNT,WORK
 (9)  034152  002022                                BGE     50357$
3056  034154                                        IF STRCNT EQ #0 THEN
 (6)  034154  005737  002320                        TST     STRCNT
 (9)  034160  001006                                BNE     50360$
3057
3058                                         ;RESET  GROUP POINTERS AND COUNTERS
3059
3060  034162                                            LET STRCNT := #33.
 (4)  034162  012737  000041  002320                    MOV     #33.,STRCNT
3061  034170                                            LET CHRGEN := #77
 (4)  034170  012737  000077  002322                    MOV     #77,CHRGEN
3062  034176                                            ENDIF
 (4)  034176                                  50360$:
3063  034176                                            LET (R4)+ :B= CHRGEN
 (4)  034176  113724  002322                            MOVB    CHRGEN,(R4)+
3064  034202                                            LET CHRGEN := CHRGEN + #1
 (6)  034202  005237  002322                            INC     CHRGEN
3065  034206                                            LET CCNT := CCNT + #1
 (6)  034206  005237  002316                            INC     CCNT
3066  034212                                            LET STRCNT := STRCNT - #1            ;UPDATE POINTERS AND COUNTERS
 (6)  034212  005337  002320                            DEC     STRCNT
3067  034216                                        ENDDO
 (4)  034216  000752                                BR      50356$
 (3)  034220                                  50357$:
3068
3069                                         ;NOW SET UP LINE TERMINATOR AND OUTPUT THE LINE.
3070
3071  034220                                        LET (R4)+ :B= #12
 (4)  034220  112724  000012                            MOVB    #12,(R4)+
3072                                         ;OUTPUT THE  LINE
3073
3074  034224                                        OUTPUT #OUTBUF,#132.          ; SEND THE DATA, NO LF YET
3075  034266                                        OUTPUT #OUTBUF+132.,#1        ; THIS MAKES SURE OUTPUT IS SENT
3076                                                                              ; BEFORE CHANGING OUTBUF DATA !
3077
3078  034330                                  ENDINC
 (4)  034330  000647                                BR      50352$
 (3)  034332                                  50353$:
3079
3080  034332                                  OUTPUT #SKIP3,#4
3081  034374                                  EXIT TST
 (3)  034374  104432                                TRAP    C$EXIT
 (3)  034376  000046                                .WORD   L10025-.
3082
3083                                         ;COUNTERS, POINTERS, TEXT BUFFER,  AND HEADER FOR TEST PRINTOUT
3084
3085  034400  000000                         SPCCNT: .WORD 0
3086
3087                                         ;TEST HEADER  MESSAGE
3088
3089  034402  050123  051125  047511         HAMFIR: .ASCIZ  /SPURIOUS HAMMER FIRING TEST 12/<12><12><12>
      034410  051525  044040  046501
      034416  042515  020122  044506
```

```
        034424  044522  043516  052040
        034432  051505  020124  031061
        034440  005012  000012
  3090
  3091                                  ;
  3092
  3093                                  .EVEN
  3094
  3095  034444                          ENDTST
   (3)  034444                          L10025:
   (3)  034444  104401                          TRAP    C$ETST
  3096
  3097  034446                          ENDMOD
```

```
                                .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.
                                ; 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
                                ;12345678901234567890123456789 0.....................          012
                                ;
                                ; NOTICE THAT THE PRINTOUT SHOULD IDENTIFY 132 COLUMNS ACROSS THE PAGE.
                                ;
                                ; THIS OUTPUT IS REPEATED 16 TIMES.
                                ;--
034446                          BGNTST  13.
034446                          T13::
                                ;PRINT TEST IDENTIFICATION
034446                          OUTPUT #PRTCTL, #56.
034510                          LET COUNT := #16.
034510  012737  000020  002314          MOV     #16.,COUNT
034516                          1$:
034516                          LET R5 := #TABLE1
034516  012705  035512                  MOV     #TABLE1,R5
034522                          WHILE (R5) NE #0 DO
034522                          50361$:
034522  005715                          TST     (R5)
034524  001421                          BEQ     50362$
034526                              OUTPUT (R5)+,#10.
034566                          ENDDO
034566  000755                          BR      50361$
034570                          50362$:
034570                          LET OUTBUF :B= #12
034570  112737  000012  003172          MOVB    #12,OUTBUF
034576                          OUTPUT #OUTBUF,#1

034640                          LET R5 := #TABLE2
034640  012705  035546                  MOV     #TABLE2,R5
034644                          WHILE (R5) NE #0 DO
034644                          50363$:
034644  005715                          TST     (R5)
034646  001421                          BEQ     50364$
034650                              OUTPUT (R5)+,#10.
034710                          ENDDO
034710  000755                          BR      50363$
034712                          50364$:
034712                          OUTPUT #OUTBUF,#1

034754                          DECR LINCNT FROM #14. TO #1 BY #1
034754  012737  000016  002310          MOV     #14.,LINCNT
034762  000402                          BR      50365$
034764                          50366$:
034764  005337  002310                  DEC     LINCNT
034770                          50365$:
```

```
   (5)  034770  023727  002310  000001          CMP     LINCNT,#1
   (7)  034776  002422                          BLT     50367$
 3135   035000                              OUTPUT #X11,#10.
 3136   035042                              ENDDECR
   (4)  035042  000750                          BR      50366$
   (3)  035044                              50367$:
 3137   035044                              OUTPUT #OUTBUF,#1
 3138   035106                              OUTPUT #OUTBUF,#1
 3139   035150                              LET COUNT := COUNT - #1
   (6)  035150  005337  002314                     DEC     COUNT
 3140   035154                              IF COUNT GT #0 THEN
   (6)  035154  005737  002314                     TST     COUNT
   (9)  035160  003402                          BLE     50370$
 3141   035162  000137  034516                     JMP 1$
 3142   035166                              ENDIF
   (4)  035166                              50370$:
 3143   035166                              LET OUTBUF := #14
   (4)  035166  012737  000014  003172              MOV     #14,OUTBUF
 3144   035174                              OUTPUT #OUTBUF,#1
 3145   035236                              EXIT TST
   (3)  035236  104432                          TRAP    C$EXIT
   (3)  035240  000342                          .WORD   L10026-.
 3146
 3147   035242  051120  047111  020124  PRTCTL: .ASCII /PRINT CONTROL TEST 13/ <12>
        035250  047503  052116  047522
        035256  020114  042524  052123
        035264  030440  005063
 3148   035270  044123  052517  042114          .ASCIZ  /SHOULD SHOW 132 COLUMNS PRINTED/<12><12><15>
        035276  051440  047510  020127
        035304  031461  020062  047503
        035312  052514  047115  020123
        035320  051120  047111  042524
        035326  005104  006412     000
 3149
 3150   035333     040  020040  020040  X0:     .ASCII  /         0/
        035340  020040  020040     060
 3151   035345     040  020040  020040  X1:     .ASCII  /         1/
        035352  020040  020040     061
 3152   035357     040  020040  020040  X2:     .ASCII  /         2/
        035364  020040  020040     062
 3153   035371     040  020040  020040  X3:     .ASCII  /         3/
        035376  020040  020040     063
 3154   035403     040  020040  020040  X4:     .ASCII  /         4/
        035410  020040  020040     064
 3155   035415     040  020040  020040  X5:     .ASCII  /         5/
        035422  020040  020040     065
 3156   035427     040  020040  020040  X6:     .ASCII  /         6/
        035434  020040  020040     066
 3157   035441     040  020040  020040  X7:     .ASCII  /         7/
        035446  020040  020040     067
 3158   035453     040  020040  020040  X8:     .ASCII  /         8/
        035460  020040  020040     070
 3159   035465     040  020040  020040  X9:     .ASCII  /         9/
        035472  020040  020040     071
 3160
 3161   035477     061  031462  032464  X11:    .ASCII  /1234567890/
```

```
                                        PRINT CONTROL

        035504  033466  034470     060
 3162
 3163           035512                  .EVEN
 3164   035512  035333  035333  035333  TABLE1: .WORD   X0,X0,X0,X0,X0,X0,X0,X0,X0,X1,X1,X1,X1,0
        035520  035333  035333  035333
        035526  035333  035333  035333
        035534  035345  035345  035345
        035542  035345  000000
 3165   035546  035345  035357  035371  TABLE2: .WORD   X1,X2,X3,X4,X5,X6,X7,X8,X9,X0,X1,X2,X3,0
        035554  035403  035415  035427
        035562  035441  035453  035465
        035570  035333  035345  035357
        035576  035371  000000
 3166                                           .EVEN
 3167
 3168   035602                  ENDTST
  (3)   035602                  L10026:
  (3)   035602  104401                   TRAP    C$ETST
 3169   035604                  ENDMOD
```

```
                                        .SBTTL  CRITICAL PATH
                                        BGNMOD
                                        STARS
 (2)                                    ;;**********************************************************************
                                        ;++
                                        ;THIS TEST ATTAINS THE HIGHEST POSSIBLE PRINTING SPEED BY SELECTING
                                        ;A DATA PATTERN WHICH EXERCISES THE PRINTER AT THE MAXIMUM DUTY CYCLE OF
                                        ;THE HAMMERS,THE TIMING LOGIC, AND THE POWER SUPPLY. A TOTAL OF 32 LINES
                                        ;LINES OF A WORST CASE PATTERN ARE PRINTED.
                                        ;
                                        STARS
 (2)                                    ;;**********************************************************************
                                        ;
                                        ;--
                                        BGNTST  14.
 (3)  035604                            T14::
                                        ;
 3186 035604                                    OUTPUT #CRTPTH,#24.
 3187 035646                                LET WORK := L$UNIT - #1
 (4)  035646  013737  002012  003166            MOV     L$UNIT,WORK
 (6)  035654  005337  003166                    DEC     WORK
 3188 035660                                INCR LUNIT FROM #0 TO WORK BY #1
 (4)  035660  005037  002326                    CLR     LUNIT
 (5)  035664  000402                            BR      50371$
 (4)  035666                            50372$:
 (7)  035666  005237  002326                    INC     LUNIT
 (5)  035672                            50371$:
 (5)  035672  023737  002326  003166            CMP     LUNIT,WORK
 (7)  035700  003143                            BGT     50373$
 3189 035702                                    LET R2 := LUNIT SHIFT #1
 (4)  035702  013702  002326                    MOV     LUNIT,R2
 (7)  035706  006302                            ASL     R2
 3190 035710                                    IF #FLAG26 NOTSETIN STATUS(R2) AND #FLAG27 NOTSETIN STATUS(R2) THEN
 (6)  035710  032762  001000  002524            BIT     #FLAG26,STATUS(R2)
 (9)  035716  001054                            BNE     50374$
 (6)  035720  032762  002000  002524            BIT     #FLAG27,STATUS(R2)
 (9)  035726  001050                            BNE     50374$
 3191                                   ; UNIT IS AN LP25
 3192 035730                                        IF #FLAG96 NOTSETIN STATUS(R2) THEN
 (6)  035730  032762  010000  002524            BIT     #FLAG96,STATUS(R2)
 (9)  035736  001022                            BNE     50375$
 3193 035740                                        OUTPUI #TAB64,#110.,,LUNIT,#32.
 3194 036002                                    ELSE
 (4)  036002  000421                            BR      50376$
 (3)  036004                            50375$:
 3195 036004                                        OUTPUI #TAB96,#132.,,LUNIT,#32.
 3196 036046                                    ENDIF
 (4)  036046                            50376$:
 3197 036046                                    ELSE
 (4)  036046  000457                            BR      50377$
 (3)  036050                            50374$:
 3198 036050                                    IF #FLAG26 SETIN STATUS(R2) AND #FLAG27 NOTSETIN STATUS(R2) THEN
 (6)  036050  032762  001000  002524            BIT     #FLAG26,STATUS(R2)
 (9)  036056  001453                            BEQ     50400$
 (6)  036060  032762  002000  002524            BIT     #FLAG27,STATUS(R2)
```

```
  (9)  036066  001047                          BNE     50400$
3199                                  ; UNIT IS AN LP26
3200  036070                                    IF #FLAG96 NOTSETIN STATUS(R2) THEN
  (6)  036070  032762  010000  002524          BIT     #FLAG96,STATUS(R2)
  (9)  036076  001022                          BNE     50401$
3201  036100                                       OUTPUI #T2664,#133.,,LUNIT,#32.
3202  036142                                    ELSE
  (4)  036142  000421                          BR      50402$
  (3)  036144                         50401$:
3203  036144                                       OUTPUI #T2696,#133.,,LUNIT,#32.
3204  036206                                    ENDIF
  (4)  036206                         50402$:
3205  036206                                 ENDIF
  (4)  036206                         50400$:
3206  036206                              ENDIF
  (4)  036206                         50377$:
3207  036206                             ENDINC
  (4)  036206  000627                          BR      50372$
  (3)  036210                         50373$:
3208  036210                                    LET OUTBUF :B= #FF       ; DO A FORM FEED
  (4)  036210  112737  000014  003172          MOVB    #FF,OUTBUF
3209  036216                                    OUTPUT #OUTBUF,#1
3210                                  ;
3211  036260                          EXIT TST
  (3)  036260  104432                          TRAP    C$EXIT
  (3)  036262  001022                          .WORD   L10027-.
3212                                  ;
3213  036264  051103  052111  041511  CRTPTH: .ASCIZ  /CRITICAL PATH TEST 14/<12><12><12>
       036272  046101  050040  052101
       036300  020110  042524  052123
       036306  030440  005064  005012
       036314     000
3214            036316                .EVEN
3215                                  ;64 CHARACTER BAND PATTERN
3216                                  ;
3217                                  ;       CRITICAL HAMMER FIRE PATTERN 64 CHARACTER BAND
3218  036316     101     064     105  TAB64:  .BYTE   101,064,105,122,104,065,106,126,060,127,111
       036321     122     104     065
       036324     106     126     060
       036327     127     111
3219  036331     067     061     132          .BYTE   067,061,132,114,137,115,045,060,127,111,067
       036334     114     137     115
       036337     045     060     127
       036342     111     067
3220  036344     061     132     114          .BYTE   061,132,114,137,115,045,063,046,120,052,121
       036347     137     115     045
       036352     063     046     120
       036355     052     121
3221  036357     072     123     057          .BYTE   072,123,057,115,045,063,046,120,052,121,072
       036362     115     045     063
       036365     046     120     052
       036370     121     072
3222  036372     123     057     125          .BYTE   123,057,125,055,066,076,130,056,131,100,123
       036375     055     066     076
       036400     130     056     131
       036403     100     123
```

```
 3223  036405  057  125  055          .BYTE   057,125,055,066,076,130,056,131,100,070,042
       036410  066  076  130
       036413  056  131  100
       036416  070  042
 3224  036420  053  124  071          .BYTE   053,124,071,042,051,136,131,100,070,135,053
       036423  042  051  136
       036426  131  100  070
       036431  135  053
 3225  036433  124  071  042          .BYTE   124,071,042,051,136,050,075,054,041,044,062
       036436  051  136  050
       036441  075  054  041
       036444  044  062
 3226  036446  073  101  051          .BYTE   073,101,051,136,050,075,054,041,044,062,073
       036451  136  050  075
       036454  054  041  044
       036457  062  073
 3227  036461  101  074  102          .BYTE   101,074,102,077,064,105,106,133,065,012,012
       036464  077  064  105
       036467  106  133  065
       036472  012  012
 3228
 3229                               .EVEN
 3230
 3231                               ;
 3232                               ;96 CHARACTER TABLE
 3233                               ;
 3234                               ;       CRITICAL HAMMER FIRE PATTERN 96 CHARACTER BAND
 3235  036474  101  064  103  TAB96:  .BYTE   101,064,103,123,060,065,106,126,061,130,111
       036477  123  060  065
       036502  106  126  061
       036505  130  111
 3236  036507  131  062  137          .BYTE   131,062,137,114,044,116,046,061,130,111,131
       036512  114  044  116
       036515  046  061  130
       036520  111  131
 3237  036522  062  137  114          .BYTE   062,137,114,044,116,046,117,051,121,072,122
       036525  044  116  046
       036530  117  051  121
       036533  072  122
 3238  036535  071  124  073          .BYTE   071,124,073,116,046,117,051,121,072,122,071
       036540  116  046  117
       036543  051  121  072
       036546  122  071
 3239  036550  124  073  125          .BYTE   124,073,125,145,127,077,066,136,132,133,124
       036553  145  127  077
       036556  066  136  132
       036561  133  124
 3240  036563  073  125  145          .BYTE   073,125,145,127,077,066,136,132,133,067,042
       036566  127  077  066
       036571  136  132  133
       036574  067  042
 3241  036576  045  040  070          .BYTE   045,040,070,075,050,041,132,133,067,042,045
       036601  075  050  041
       036604  132  133  067
       036607  042  045
 3242  036611  040  070  075          .BYTE   040,070,075,050,041,057,047,053,175,076,176
```

```
                036614    050     041     057
                036617    047     053     175
                036622    076     176
3243            036624    074     056     050             .BYTE   074,056,050,041,057,047,053,175,076,176,074
                036627    041     057     047
                036632    053     175     076
                036635    176     074
3244            036637    056     100     140             .BYTE   056,100,140,135,060,134,143,054,061,040,040
                036642    135     060     134
                036645    143     054     061
                036650    040     040
3245            036652    040     040     040             .BYTE   040,040,040,060,040,040,040,040,040,040,040
                036655    060     040     040
                036660    040     040     040
                036663    040     040
3246            036665    150     012     012             .BYTE   150,012,012
3247
3248                                                    ; LP26 96 CHAR PATTERN
3249
3250            036670  041501  030505  046112  T2696:  .ASCII  /ACE1JL3QR1J2NPR5W6NP2L36_%)TV6_$8:+/<145>
                036676  050463  030522  031112
                036704  050116  032522  033127
                036712  050116  046062  033063
                036720  022537  052051  033126
                036726  022137  035070  062453
3251            036734  022132  035070  037053          .ASCII  /Z$8:+>?]''(9>?] *'</<164>/\ *'/<173>
                036742  056477  024042  037071
                036750  056477  025040  036047
                036756  056164  025040  075447
3252            036764  023536  036454  076443          .ASCIZ  /^',=#/<175>/^@/<142><144><147><175><176><177>
                036772  040136  062142  076547
                037000  077576    000
3253            037003    060     144     147           .BYTE   60,144,147,62,154,155,60,144,146,151,113,155,64
                037006    062     154     155
                037011    060     144     146
                037014    151     113     155
                037017    064
3254            037020    162     065     151           .BYTE   162,65,151,153,155,157,161,65,167,171,101,157,161
                037023    153     155     157
                037026    161     065     167
                037031    171     101     157
                037034    161
3255            037035    065     167     170           .BYTE   65,167,170,67,103,105,71,166,170,67,103,105,107
                037040    067     103     105
                037043    071     166     170
                037046    067     103     105
                037051    107
3256            037052  045511  041116  043470          .ASCII  /IKNB8GIKNO,TH/<145>/MO,T/<12><12>
                037060  045511  047516  052054
                037066  062510  047515  052054
                037074  005012
3257
3258
3259                                                    ; LP26 64 CHAR PATTERN
3260                                                    ;
3261            037076  042501  030124  046112  T2664:  .ASCII  /AETOJLNPQOJKMOQ56XMOQSVXZ+&SVXZ+9/
```

```
        037104  050116  030121  045512
        037112  047515  032521  054066
        037120  047515  051521  054126
        037126  025532  051446  054126
        037134  025532     071
 3262   037137     043  026454  057531          .ASCII  /#,-Y_9#,$>?]):$>?]TO=;.ETO=#'B["'!'BDGI!2/
        037144  021471  022054  037476
        037152  024535  022072  037476
        037160  052135  036460  027073
        037166  052105  036460  023443
        037174  055502  020442  041047
        037202  043504  020511     062
 3263   037207     063  043464  045511          .ASCII  /34GIKMN4GHJLN9-UJLNPRUWY_PRUWX.ETOVX.ET(/
        037214  047115  043464  045110
        037222  047114  026471  045125
        037230  047114  051120  053525
        037236  057531  051120  053525
        037244  027130  052105  053060
        037252  027130  052105     050
 3264   037257     057  036073  023053          .ASCIZ  <57>/;<+&(/<57>/;<34":1234"/<12><12>
        037264  027450  036073  032063
        037272  035042  031061  032063
        037300  005042  000012
 3265                                   .EVEN
 3266
 3267   037304                         ENDTST
  (3)   037304                         L10027:
  (3)   037304  104401                          TRAP    C$ETST
 3268   037306                         ENDMOD
 3269
```

```
                                       .SBTTL  MULTIPLE LINE ADVANCE

                                       BGNMOD
                                       ;++
                                       ;THIS TEST CHECKS THE MULTIPLE LINE ADVANCE OF THE LINE PRINTER. A LINE OF
                                       ;NUMBERS IS PRINTED AND THEN THE PAPER IS ADVANCED THAT NUMBER OF LINES. THUS THE
                                       ;NUMBER PRINTED WILL INDICATE THE NUMBER OF BLANK LINES FOLLOWING THAT
                                       ;LINE. THE NUMBER OF LINES IS VARIED BETWEEN 2 AND 7 AND A LINE OF
                                       ;ALL 0'S WILL INDICATE THE END OF THE TEST  SEQUENCE.
                                       ;--


3283  037306                           BGNTST  15.
 (3)  037306                           T15::

                                       ;PRINT TEST IDENTIFICATION

3287  037306                           OUTPUT #MULINE,#86.

3289  037350                           LET STACHR := #TABSTR              ;OUTPUT CHARACTERS
 (4)  037350  012737  037616  037614          MOV     #TABSTR,STACHR

3291  037356                           REPEAT
 (3)  037356                           50403$:
3292  037356                               LET LINCNT :B= @STACHR                ;GET A CHARACTER TO OUTPUT
 (4)  037356  117737  000232  002310          MOVB    @STACHR,LINCNT
3293  037364                               LET LINCNT := LINCNT AND #7 ;MAKE THE ASCII TO OCTAL
 (6)  037364  013746  002310               MOV     LINCNT,-(SP)
 (6)  037370  042716  000007               BIC     #7,(SP)
 (6)  037374  042637  002310               BIC     (SP)+,LINCNT
3294  037400                           LET R3 := #OUTBUF                 ;SET UP OUTPUT BUFFER
 (4)  037400  012703  003172               MOV     #OUTBUF,R3
3295  037404                           INCR CCNT FROM #1 TO #132. BY #1
 (4)  037404  012737  000001  002316       MOV     #1,CCNT
 (5)  037412  000402                       BR      50404$
 (4)  037414                           50405$:
 (7)  037414  005237  002316               INC     CCNT
 (5)  037420                           50404$:
 (5)  037420  023727  002316  000204       CMP     CCNT,#132.
 (7)  037426  003003                       BGT     50406$
3296  037430                               LET (R3)+ :B= @STACHR             ;PUT CHARACTER  IN OUTPUT BUFFER
 (4)  037430  117723  000160               MOVB    @STACHR,(R3)+
3297  037434                           ENDINC
 (4)  037434  000767                       BR      50405$
 (3)  037436                           50406$:
3298  037436                           LET R4 := #0
 (4)  037436  005004                       CLR     R4
3299  037440                           WHILE R4 NE LINCNT DO
 (4)  037440                           50407$:
 (6)  037440  020437  002310               CMP     R4,LINCNT
 (9)  037444  001404                       BEQ     50410$
3300  037446                               LET (R3)+ :B= #12                 ;FILL WITH LINE FEEDS
 (4)  037446  112723  000012               MOVB    #12,(R3)+
3301  037452                               LET R4 := R4 + #1
 (6)  037452  005204                       INC     R4
3302  037454                           ENDDO
```

```
 (4)  037454  000771                        BR      50407$
 (3)  037456                          50410$:
3303
3304                                  ;NOW OUTPUT THE ACTUAL LINE
3305
3306  037456                              LET R4 := LINCNT + #132.          ;NUMBER OF CHARACTERS TO OUTPUT
 (4)  037456  013704  002310                 MOV     LINCNT,R4
 (6)  037462  062704  000204                 ADD     #132.,R4
3307  037466                              LET STACHR := STACHR + #1          ; UPDATE CHARACTER COUNT
 (6)  037466  005237  037614                 INC     STACHR
3308  037472                              OUTPUT #OUTBUF,R4                 ;OUTPUT THE LINE
3309
3310  037532                          UNTIL LINCNT EQ #0
 (3)  037532  005737  002310                 TST     LINCNT
 (6)  037536  001307                         BNE     50403$
3311  037540                          LET OUTBUF := #14
 (4)  037540  012737  000014  003172         MOV     #14,OUTBUF
3312  037546                          OUTPUT #OUTBUF,#1
3313
3314  037610                          EXIT TST
 (3)  037610  104432                         TRAP    C$EXIT
 (3)  037612  000150                         .WORD   L10030-.
3315
3316
3317  037614  000000                  STACHR: .WORD 0
3318
3319  037616  033462  033062  033463  TABSTR: .ASCIZ  /272637463540/
       037624  033064  032463  030064
       037632     000
3320  037633     115  046125  044524  MULINE: .ASCII  /MULTIPLE LINE ADVANCE TEST 15/<12>
       037640  046120  020105  044514
       037646  042516  040440  053104
       037654  047101  042503  052040
       037662  051505  020124  032461
       037670     012
3321  037671     116  046525  042502          .ASCIZ  /NUMBERS PRINTED REPRESENT # LINES TO NEXT LINE PRINTED/<12><12>
       037676  051522  050040  044522
       037704  052116  042105  051040
       037712  050105  042522  042523
       037720  052116  021440  046040
       037726  047111  051505  052040
       037734  020117  042516  052130
       037742  046040  047111  020105
       037750  051120  047111  042524
       037756  005104  000012
3322
3323
3324
3325                                  .EVEN
3326
3327  037762                          ENDTST
 (3)  037762                          L10030:
 (3)  037762  104401                         TRAP    C$ETST
3328  037764                          ENDMOD
3329
```

```
3331                                    .SBTTL  CHARACTER ALIGNMENT
3332  037764                            BGNMOD
3333                                    ;++
3334                                    ;THIS TEST CHECKS CHARACTER ALIGNMENT BY OVERPRINTING LINES OF ALTERNATING
3335                                    ;H'S AND SPACES WITH SPACES AND H'S.
3336                                    ;--
3337  037764                            BGNTST  16.
 (3)  037764                            T16::
3338                                    ;PRINT TEST IDENTIFICATION
3339  037764                                    OUTPUT #CHRALN,#30.     ; PRINT TEST NAME ON LP
3340                                    ;PRINT 24 LINES OF ALTERNATING ''H'''S AND ''SPACE'''S
3341  040026                            1$:     LET LINCNT := #24.
 (4)  040026  012737  000030  002310          MOV     #24.,LINCNT
3342  040034                            2$:     IF LINCNT LE #0 THEN
 (6)  040034  005737  002310                  TST     LINCNT
 (9)  040040  003002                          BGT     50411$
3343  040042                                      INLINE <JMP 3$>
 (2)  040042  000137  040350                  JMP     3$
3344  040046                                    ENDIF
 (4)  040046                            50411$:
3345                                    ;LOAD BUFFER WITH ALTERNATING STRING OF ''H'''S AND ''SPACE'''S
3346  040046                                    LET R4 := #OUTBUF
 (4)  040046  012704  003172                  MOV     #OUTBUF,R4
3347  040052                                    INCR WORK FROM #1 TO #66. BY #1     ; 132 CHARACTERS
 (4)  040052  012737  000001  003166          MOV     #1,WORK
 (5)  040060  000402                          BR      50412$
 (4)  040062                            50413$:
 (7)  040062  005237  003166                  INC     WORK
 (5)  040066                            50412$:
 (5)  040066  023727  003166  000102          CMP     WORK,#66.
 (7)  040074  003005                          BGT     50414$
3348  040076                                        LET (R4)+ :B= #110          ; PUT PATTERN INTO BUFFER
 (4)  040076  112724  000110                  MOVB    #110,(R4)+
3349  040102                                        LET (R4)+ :B= #40
 (4)  040102  112724  000040                  MOVB    #40,(R4)+
3350  040106                                    ENDINCR
 (4)  040106  000765                          BR      50413$
 (3)  040110                            50414$:
3351  040110                                    LET (R4)+ :B= #CR               ; FOLLOWED BY CR
 (4)  040110  112724  000015                  MOVB    #CR,(R4)+
3352                                    ;
3353                                    ; SEND BASIC PATTERN
3354                                    ;
3355  040114                                    OUTPUT  #OUTBUF,#132.
3356  040156                                    OUTPUT #OUTBUF+132.,#1
3357                                    ;
3358                                    ; OVERPRINT WITH LINE OF ALTERNATING SPACE AND 'H'
3359                                    ;
3360  040220                                    LET R4 := #OUTBUF+132.          ; FILL BUFFER WITH REVERSE PATTERN
 (4)  040220  012704  003376                  MOV     #OUTBUF+132.,R4
3361  040224                                        LET (R4)+ :B= #110  ; H
 (4)  040224  112724  000110                  MOVB    #110,(R4)+
3362  040230                                    LET (R4)+ :B= #LF       ; FOLLOWED BY A LINEFEED
 (4)  040230  112724  000012                  MOVB    #LF,(R4)+
3363
3364  040234                                    OUTPUT  #OUTBUF+1,#132.         ; OVERPRINT
```

```
 3365  040276                                    OUTPUT #OUTBUF+133.,#1
 3366  040340                                    LET LINCNT := LINCNT - #1
  (6)  040340  005337  002310                    DEC     LINCNT
 3367  040344                                    INLINE <JMP     2$>
  (2)  040344  000137  040034                    JMP     2$
 3368  040350                              3$:
 3369  040350                              LET OUTBUF := #14
  (4)  040350  012737  000014  003172            MOV     #14,OUTBUF
 3370  040356                              OUTPUT #OUTBUF,#1
 3371  040420                              EXIT TST
  (3)  040420  104432                            TRAP    C$EXIT
  (3)  040422  000042                            .WORD   L10031-.
 3372  040424  044103  051101  041501      CHRALN: .ASCIZ /CHARACTER ALIGNMENT TEST 16/<12><12><12>
       040432  042524  020122  046101
       040440  043511  046516  047105
       040446  020124  042524  052123
       040454  030440  005066  005012
       040462     000
 3373        040464                        .EVEN
 3374  040464                              ENDTST
  (3)  040464                              L10031:
  (3)  040464  104401                            TRAP    C$ETST
 3375  040466                              ENDMOD
```

```
3377                                    .SBTTL  INTERRUPT SERVICE ROUTINES
3378  040466                            BGNSRV
3379                                    ;
3380                                    ;++
3381                                    ;INTERRUPT VECTORS ARE ESTABLISHED DURING INITIALIZATION
3382                                    ;POINTING TO THE BASIC ROUTINES WHICH
3383                                    ;SET UP THE UNIT NUMBER CAUSING THE INTERRUPTS.
3384                                    ;LINE NUMBER IS RETURNED IN R2
3385                                    ;
3386                                    ;--
3387            000000                  X=0
3388  040466    000020                  INT00:  .REPT 16.
3389                                            SETPRI #PRI04
3390                                            PUSH    R2
3391                                            LET R2 := #X
3392                                            INLINE <JMP     IODRV>
3393                                    X=X+2
3394                                            .ENDR
  (4) 040466  012700  000200                   MOV     #PRI04,R0
  (4) 040472  104441                            TRAP    C$SPRI
  (3) 040474  010246                            MOV     R2,-(SP)
  (5) 040476  012702  000000                    MOV     #X,R2
  (3) 040502  000137  004704                    JMP     IODRV
  (4) 040506  012700  000200                    MOV     #PRI04,R0
  (4) 040512  104441                            TRAP    C$SPRI
  (3) 040514  010246                            MOV     R2,-(SP)
  (5) 040516  012702  000002                    MOV     #X,R2
  (3) 040522  000137  004704                    JMP     IODRV
  (4) 040526  012700  000200                    MOV     #PRI04,R0
  (4) 040532  104441                            TRAP    C$SPRI
  (3) 040534  010246                            MOV     R2,-(SP)
  (5) 040536  012702  000004                    MOV     #X,R2
  (3) 040542  000137  004704                    JMP     IODRV
  (4) 040546  012700  000200                    MOV     #PRI04,R0
  (4) 040552  104441                            TRAP    C$SPRI
  (3) 040554  010246                            MOV     R2,-(SP)
  (5) 040556  012702  000006                    MOV     #X,R2
  (3) 040562  000137  004704                    JMP     IODRV
  (4) 040566  012700  000200                    MOV     #PRI04,R0
  (4) 040572  104441                            TRAP    C$SPRI
  (3) 040574  010246                            MOV     R2,-(SP)
  (5) 040576  012702  000010                    MOV     #X,R2
  (3) 040602  000137  004704                    JMP     IODRV
  (4) 040606  012700  000200                    MOV     #PRI04,R0
  (4) 040612  104441                            TRAP    C$SPRI
  (3) 040614  010246                            MOV     R2,-(SP)
  (5) 040616  012702  000012                    MOV     #X,R2
  (3) 040622  000137  004704                    JMP     IODRV
  (4) 040626  012700  000200                    MOV     #PRI04,R0
  (4) 040632  104441                            TRAP    C$SPRI
  (3) 040634  010246                            MOV     R2,-(SP)
  (5) 040636  012702  000014                    MOV     #X,R2
  (3) 040642  000137  004704                    JMP     IODRV
  (4) 040646  012700  000200                    MOV     #PRI04,R0
  (4) 040652  104441                            TRAP    C$SPRI
  (3) 040654  010246                            MOV     R2,-(SP)
```

```
   (5)  040656  012702  000016                MOV     #X,R2
   (3)  040662  000137  004704                JMP     IODRV
   (4)  040666  012700  000200                MOV     #PRI04,R0
   (4)  040672  104441                        TRAP    C$SPRI
   (3)  040674  010246                        MOV     R2,-(SP)
   (5)  040676  012702  000020                MOV     #X,R2
   (3)  040702  000137  004704                JMP     IODRV
   (4)  040706  012700  000200                MOV     #PRI04,R0
   (4)  040712  104441                        TRAP    C$SPRI
   (3)  040714  010246                        MOV     R2,-(SP)
   (5)  040716  012702  000022                MOV     #X,R2
   (3)  040722  000137  004704                JMP     IODRV
   (4)  040726  012700  000200                MOV     #PRI04,R0
   (4)  040732  104441                        TRAP    C$SPRI
   (3)  040734  010246                        MOV     R2,-(SP)
   (5)  040736  012702  000024                MOV     #X,R2
   (3)  040742  000137  004704                JMP     IODRV
   (4)  040746  012700  000200                MOV     #PRI04,R0
   (4)  040752  104441                        TRAP    C$SPRI
   (3)  040754  010246                        MOV     R2,-(SP)
   (5)  040756  012702  000026                MOV     #X,R2
   (3)  040762  000137  004704                JMP     IODRV
   (4)  040766  012700  000200                MOV     #PRI04,R0
   (4)  040772  104441                        TRAP    C$SPRI
   (3)  040774  010246                        MOV     R2,-(SP)
   (5)  040776  012702  000030                MOV     #X,R2
   (3)  041002  000137  004704                JMP     IODRV
   (4)  041006  012700  000200                MOV     #PRI04,R0
   (4)  041012  104441                        TRAP    C$SPRI
   (3)  041014  010246                        MOV     R2,-(SP)
   (5)  041016  012702  000032                MOV     #X,R2
   (3)  041022  000137  004704                JMP     IODRV
   (4)  041026  012700  000200                MOV     #PRI04,R0
   (4)  041032  104441                        TRAP    C$SPRI
   (3)  041034  010246                        MOV     R2,-(SP)
   (5)  041036  012702  000034                MOV     #X,R2
   (3)  041042  000137  004704                JMP     IODRV
   (4)  041046  012700  000200                MOV     #PRI04,R0
   (4)  041052  104441                        TRAP    C$SPRI
   (3)  041054  010246                        MOV     R2,-(SP)
   (5)  041056  012702  000036                MOV     #X,R2
   (3)  041062  000137  004704                JMP     IODRV
3395
```

```
                                .SBTTL CLOCK SERVICE ROUTINE
                                ;++
                                ;UPDATES THE COUNTER AT A RATE OF 16.67 MILLISECONDS PER TICK
                                ;AND UPDATES A SECOND COUNTER WHEN THE FIRST OVERFLOWS.
                                ;--

                                BGNSRV
041066                          CLKTCK: SETPRI #PRIO6
041066  012700  000300                  MOV     #PRIO6,R0
041072  104441                          TRAP    C$SPRI
041074                                  IF TICK EQ #0 THEN
041074  005737  041142                  TST     TICK
041100  001005                          BNE     50415$
041102                                      LET TICK := #60.            ;60 TICKS PER SECOND
041102  012737  000074  041142          MOV     #60.,TICK
041110                                      LET TIME := TIME + #1
041110  005237  041140                  INC     TIME
041114                                  ENDIF
041114                          50415$:
041114                                  LET TICK := TICK - #1           ;BACK UP SECOND TIMER
041114  005337  041142                  DEC     TICK
041120                                  IF CLKTYP EQ #2 THEN
041120  023727  002334  000002          CMP     CLKTYP,#2
041126  001003                          BNE     50416$
041130                                      LET @CLKCSR := #100
041130  012777  000100  141202          MOV     #100,@CLKCSR
041136                                  ENDIF
041136                          50416$:

041136                          ENDSRV                              ;AND EXIT
041136                          L10033:
041136  000002                          RTI
                                ;
041140  000000                  TIME:   .WORD   0
041142  000000                  TICK:   .WORD   0
```

```
 3419                                   .SBTTL  HARDWARE PARAMETER SECTION
 3420  041144                           BGNMOD
 3421
 3422                                   ;+
 3423                                   ;THIS SECTION INCLUDES THE QUESTIONS WHICH REQUEST THE OPERATOR TO
 3424                                   ;FURNISH THE HARDWARE INFORMATION NECESSARY TO BUILD THE HARDWARE
 3425                                   ;P-TABLES.
 3426                                   ;
 3427                                   ;--
 3428  041144                           BGNHRD
  (3)  041144  000020     ;                     .WORD L10034-L$HARD/2
  (3)  041146                           L$HARD::
 3429
 3430  041146                                   GPRMA   GETADR,0,0,160000,177516,YES
  (4)  041146  000031                             .WORD   T$CODE
  (4)  041150  041206                             .WORD   GETADR
  (4)  041152  160000                             .WORD   T$LOLIM
  (4)  041154  177516                             .WORD   T$HILIM
 3431  041156                                   GPRMA   GETVEC,2,0,110,770,YES
  (4)  041156  001031                             .WORD   T$CODE
  (4)  041160  041223                             .WORD   GETVEC
  (4)  041162  000110                             .WORD   T$LOLIM
  (4)  041164  000770                             .WORD   T$HILIM
 3432  041166                                   GPRMD   GETTYP,4,0,3,0,3,YES
  (4)  041166  002032                             .WORD   T$CODE
  (4)  041170  041244                             .WORD   GETTYP
  (4)  041172  000003                             .WORD   3
  (4)  041174  000000                             .WORD   T$LOLIM
  (4)  041176  000003                             .WORD   T$HILIM
 3433  041200                                   GPRML   GETBND,6,1,YES
  (4)  041200  003130                             .WORD   T$CODE
  (4)  041202  041300                             .WORD   GETBND
  (4)  041204  000001                             .WORD   1
 3434  041206                           ENDHRD
  (2)                                             .EVEN
  (3)  041206                           L10034:
 3435
 3436  041206  050114  030461  040440   GETADR: .ASCIZ  /LP11 ADDRESS/
       041214  042104  042522  051523
       041222     000
 3437  041223     111  052116  051105   GETVEC: .ASCIZ  /INTERRUPT VECTOR/
       041230  052522  052120  053040
       041236  041505  047524  000122
 3438  041244  047105  042524  020122   GETTYP: .ASCIZ  /ENTER 0 IF LP25, 1 IF LP26 /
       041252  020060  043111  046040
       041260  031120  026065  030440
       041266  044440  020106  050114
       041274  033062  000040
 3439  041300  033071  041440  040510   GETBND: .ASCIZ  /96 CHARACTER BAND/
       041306  040522  052103  051105
       041314  041040  047101  000104
 3443                                   .EVEN
```

```
                                        .SBTTL  SOFTWARE PARAMETER SECTION
                                        ;
                                        ;++
                                        ;THIS SECTION INCLUDES THE QUESTIONS WHICH REQUEST THE OPERATOR TO FURNISH
                                        ;THE SOFTWARE INFORMATION NECESSARY TO BUILD THE SOFTWARE P-TABLES.
                                        ;--
                                        ;
041322                                  BGNSFT
(3)  041322  000026                             .WORD L10035-L$SOFT/2
(3)  041324                             L$SOFT.:
041324                                          GPRML   MGTINT,0,1,YES
(4)  041324  000130                             .WORD   T$CODE
(4)  041326  041400                             .WORD   MGTINT
(4)  041330  000001                             .WORD   1
041332                                          GPRML   GETDAV,2,1,YES
(4)  041332  001130                             .WORD   T$CODE
(4)  041334  041436                             .WORD   GETDAV
(4)  041336  000001                             .WORD   1
041340                                          GPRML   GETMAN,4,1,YES
(4)  041340  002130                             .WORD   T$CODE
(4)  041342  041465                             .WORD   GETMAN
(4)  041344  000001                             .WORD   1
041346                                          GPRMD   GETTIM,6,D,377,4,60.,YES
(4)  041346  003052                             .WORD   T$CODE
(4)  041350  041537                             .WORD   GETTIM
(4)  041352  000377                             .WORD   377
(4)  041354  000004                             .WORD   T$LOLIM
(4)  041356  000074                             .WORD   T$HILIM
041360                                          GPRML   GETPLA,10,1,YES
(4)  041360  004130                             .WORD   T$CODE
(4)  041362  041624                             .WORD   GETPLA
(4)  041364  000001                             .WORD   1
041366                                          GPRMD   GETMAX,12,D,377,1,255.,YES
(4)  041366  005052                             .WORD   T$CODE
(4)  041370  041645                             .WORD   GETMAX
(4)  041372  000377                             .WORD   377
(4)  041374  000001                             .WORD   T$LOLIM
(4)  041376  000377                             .WORD   T$HILIM

                                        ENDSFT
(2)                                             .EVEN
(3)  041400                             L10035:

                                        .NLIST BEX
041400  052522  020116  040515          MGTINT: .ASCIZ   /RUN MANUAL INTERVENTION TESTS/
041436  040504  043126  020125          GETDAV: .ASCIZ  /DAVFU OPTION INSTALLED/
041465     120  051105  047506          GETMAN: .ASCIZ  /PERFORM MANUAL PRINTING SPEED MEASUREMENT/
041537     104  051505  051111          GETTIM: .ASCIZ  /DESIRED TIME INTERVAL FOR PRINTING SPEED CALCULATION/
041624  042524  052123  047111          GETPLA: .ASCIZ  /TESTING IN U.S.A/
041645     101  052125  042117          GETMAX: .ASCIZ  /AUTODROP ERROR COUNT/
                                        .LIST BEX
                                        .EVEN
                                        ;

041672  000020                          PATCH:  .BLKW   20
```

```
 3475  041732                          LASTAD
  (2)                                          .EVEN
  (4)  041732  000000                          .WORD   0
  (4)  041734  000000                          .WORD   0
  (3)  041736                          L$LAST::
 3476  041736                          ENDMOD
 3477          000001                  .END
```

```
ACTIVE= 020000          BUFREP  002366          C$GMAN= 000043          E$LOAD= 000035          G$OFSI= 000376
ADR   = 000020 G        CCNT    002316          C$GPHR= 000042          FAKE    005642          G$PRMA= 000001
ASSEMB= 000010          CHAR    027236          C$GPLO= 000030          FF    = 000014 G        G$PRMD= 000002
AUTCON  030762          CHARSP  026604          C$GPRI= 000040          FFSET   015474          G$PRML= 000000
BANDSW  013425          CHARX   026605          C$INIT= 000011          FLAG    002306          G$RADA= 000140
BIN2DA  004472          CHRALN  040424          C$INLP= 000020          FLAGDA  004676          G$RADB= 000000
BIT0  = 000001 G        CHRGEN  002322          C$MANI= 000050          FLAG26= 001000          G$RADD= 000040
BIT00 = 000001 G        CK1     006524          C$MEM = 000031          FLAG27= 002000          G$RADL= 000120
BIT01 = 000002 G        CLEAN   007362          C$MSG = 000023          FLAG96= 010000          G$RADO= 000020
BIT02 = 000004 G        CLKCSR  002340          C$OPEN= 000034          FLSMSG  015550          G$XFER= 000004
BIT03 = 000010 G        CLKENA  002346          C$PNTB= 000014          FLSMS1  015642          G$YES = 000010
BIT04 = 000020 G        CLKSET  002342          C$PNTF= 000017          FLSSEL  015003          HAMFIR  034402
BIT05 = 000040 G        CLKTCK  041066          C$PNTS= 000016          FRMLTH  014576          HAMRDY  013310
BIT06 = 000100 G        CLKTYP  002334          C$PNTX= 000015          FSTMSG  025007          HAMRSW  013147
BIT07 = 000200 G        CLKVEC  002344          C$QIO = 000377          F$AU  = 000015          HAMSW1  013233
BIT08 = 000400 G        CLOCKP  002336          C$RDBU= 000007          F$AUTO= 000020          HOE   = 100000 G
BIT09 = 001000 G        COUNT   002314          C$REFG= 000047          F$BGN = 000040          IBE   = 010000 G
BIT1  = 000002 G        COUNTD  004700          C$RESE= 000033          F$CLEA= 000007          IDU   = 000040 G
BIT10 = 002000 G        CR    = 000015 G        C$REVI= 000003          F$DU  = 000016          IER   = 020000 G
BIT11 = 004000 G        CRTPTH  036264          C$RFLA= 000021          F$END = 000041          IGNORE  007266
BIT12 = 010000 G        CSRERR  003420          C$RPT = 000025          F$HARD= 000004          INCDAT  023344
BIT13 = 020000 G        CTLEND  005460          C$SEFG= 000046          F$HW  = 000013          INCTBL  023700
BIT14 = 040000 G        CTLLOP  005142          C$SPRI= 000041          F$INIT= 000006          INDEX   002356
BIT15 = 100000 G        CURADD  002564          C$SVEC= 000037          F$JMP = 000050          INHINT  002272
BIT2  = 000004 G        CURCNT  002764          C$TPRI= 000013          F$MOD = 000000          INSTR   026522
BIT3  = 000010 G        C$AU  = 000052          DATPTH  027240          F$MSG = 000011          INTERR  010422
BIT4  = 000020 G        C$AUTO= 000061          DEL   = 000177 G        F$PROT= 000021          INTER1  003640
BIT5  = 000040 G        C$BRK = 000022          DELCNT  003064          F$PWR = 000017          INTFAC  010460
BIT6  = 000100 G        C$BSEG= 000004          DFPTBL  002254 G        F$RPT = 000012          INTHDL  010452
BIT7  = 000200 G        C$BSUB= 000002          DIAGMC= 000000          F$SEG = 000003          INTLK   012220
BIT8  = 000400 G        C$CEFG= 000045          DIGITS  004702          F$SOFT= 000005          INT00   040466
BIT9  = 001000 G        C$CLCK= 000062          DOTOF   012723          F$SRV = 000010          IOCTRL  005102
BNDPAT  020742          C$CLEA= 000012          DROPED= 040000          F$SUB = 000002          IODRV   004704
BNDRDY  013562          C$CLOS= 000035          DROPIT  005534          F$SW  = 000014          ISR   = 000100 G
BNDSWI  003570          C$CLP1= 000006          EF.CON= 000036 G        F$TEST= 000001          IXE   = 004000 G
BNDSW1  013513          C$CVEC= 000036          EF.NEW= 000035 G        GETADR  041206          I$AU  = 000041
BNDTST  032706          C$DCLN= 000044          EF.PWR= 000034 G        GETBND  041300          I$AUTO= 000041
BNKSWI  003523          C$DODU= 000051          EF.RES= 000037 G        GETDAV  041436          I$CLN = 000041
BOE   = 000400 G        C$DRPT= 000024          EF.STA= 000040 G        GETFLG  023266          I$DU  = 000041
BPID25  032735          C$DU  = 000053          END2    010240          GETMAN  041465          I$HRD = 000041
BPID26  032743          C$EDIT= 000003          END4    017514          GETMAX  041645          I$INIT= 000041
BP64ID  032751          C$ERDF= 000055          ERRCOD  002350          GETPLA  041624          I$MOD = 000041
BP64Q1  033025          C$ERHR= 000056          ERRFLG  002352          GETTIM  041537          I$MSG = 000041
BP64Q2  033121          C$ERRO= 000060          ERROR = 100000          GETTYP  041244          I$PROT= 000040
BP64Q3  033216          C$ERSF= 000054          ERRSVC  003124          GETVEC  041223          I$PTAB= 000041
BP64Q4  033312          C$ERSO= 000057          ERRTBL  003126 G        G$CNTO= 000200          I$PWR = 000041
BP96ID  032777          C$ESCA= 000010          ERR06   023563          G$DELM= 000372          I$RPT = 000041
BP96Q1  033410          C$ESEG= 000005          ERR07   023634          G$DISP= 000003          I$SEG = 000041
BP96Q2  033504          C$ESUB= 000003          ERR11   010511          G$EXCP= 000400          I$SETU= 000041
BP96Q3  033601          C$ETST= 000001          ERR12   010565          G$HILI= 000002          I$SFT = 000041
BP96Q4  033675          C$EXIT= 000032          ERR13   010650          G$LOLI= 000001          I$SRV = 000041
BUFADD  002362          C$GETB= 000026          EVL   = 000004 G        G$NO  = 000000          I$SUB = 000041
BUFCNT  002364          C$GETW= 000027          E$END = 002100          G$OFFS= 000400          I$TST = 000041
```

```
J$JMP = 000167
LCTMSG  025212
LCTTBL  024726
LF    = 000012 G
LINCNT  002310
LINPER  021004
LINSWI  014637
LINSW1  014723
LN3EX   013017
LOBYTE= 000377
LOE   = 040000 G
LOT   = 000010 G
LPBUF   002464
LPCSR   002370
LPDROP  004230
LPERR   004276
LPERR2  020732
LPINTR  003024
LPM64   021406
LPM96   021467
LPVEC   002430
LSTCNT  002312
LUNIT   002326
L$ACP   002110 G
L$APT   002036 G
L$AUT   002070 G
L$AUTO  002264 G
L$CCP   002106 G
L$CLEA  007352 G
L$CO    002032 G
L$DEPO  002011 G
L$DESC  002172 G
L$DESP  002076 G
L$DEVP  002060 G
L$DISP  002132 G
L$DLY   002116 G
L$DTP   002040 G
L$DTYP  002034 G
L$DUT   002072 G
L$DVTY  002240 G
L$EF    002052 G
L$ENVI  002044 G
L$ETP   002102 G
L$EXP1  002046 G
L$EXP4  002064 G
L$EXP5  002066 G
L$HARD  041146 G
L$HIME  002120 G
L$HPCP  002016 G
L$HPTP  002022 G
L$HW    002254 G
L$ICP   002104 G
L$INIT  005676 G
L$LADP  002026 G
L$LAST  041736 G
L$LOAD  002100 G
L$LUN   002074 G
L$MREV  002050 G
L$NAME  002000 G
L$PRIO  002042 G
L$PROT  002122 G
L$PRT   002112 G
L$REPP  002062 G
L$REV   002010 G
L$SOFT  041324 G
L$SPC   002056 G
L$SPCP  002020 G
L$SPTP  002024 G
L$STA   002030 G
L$SW    002272 G
L$TEST  002114 G
L$TIML  002014 G
L$UNIT  002012 G
L10001  002264
L10002  002266
L10003  002306
L10004  007264
L10005  007544
L10006  010724
L10007  010450
L10010  010456
L10011  013702
L10012  016160
L10013  022326
L10014  023714
L10015  022666
L10016  023264
L10017  025420
L10020  026670
L10021  027302
L10022  030130
L10023  031462
L10024  033774
L10025  034444
L10026  035602
L10027  037304
L10030  037762
L10031  040464
L10033  041136
L10034  041206
L10035  041400
L26M64  021550
L26M96  021631
MANSPD  002276
MAXERR  002304
MGTINT  041400
MOVMSG  015754
MOVMS1  016051
MOVMS2  016147
MRESET  006773
MSGADR  002724
MSGCNT  002624
MULINE  037633
NMLFLS  015402
NOCLCK  007127
NOCLK   004161
NOINTR= 000003
NONBUF  031065
NONCHR  030652
NOSTOP  023415
NOTIM   007171
NRGT16  006654
NRGT17  006737
NSTTBL  023705
OFFLIN  021051
ONEFIL= 000001
ONLIN1  021111
ONLIN2  021212
ONLIN3  021310
OUTBUF  003172
OUTTIM  003720
O$APTS= 000000
O$AU  = 000000
O$BGNR= 000000
O$BGNS= 000001
O$DU  = 000000
O$ERRT= 000000
O$GNSW= 000001
O$POIN= 000001
O$SETU= 000000
PAPCHK  015102
PAPOUT  012540
PAPRDY  012404
PAPRSW  012256
PAPSWI  003460
PAPSW1  012326
PAPTST  012506
PATCH   041672
PATTER  027300
PERIOD  002300
PLOC    007262
PNT   = 001000 G
PRI   = 002000 G
PRINTR  002332
PRINT4  012572
PRIO0 = 000000 G
PRIO1 = 000040 G
PRIO2 = 000100 G
PRIO3 = 000140 G
PRIO4 = 000200 G
PRIO5 = 000240 G
PRIO6 = 000300 G
PRIO7 = 000340 G
PRTCHR  030070
PRTCTL  035242
PRTSPD  021341
PTABAD  002330
QUIET   005466
RDYERR  003436
READY   007072
REFLIN  015174
REFLI1  015271
REFLI2  015366
REPCNT  002664
REPLUP  017632
RESET   023460
RESVEC  007270 G
SECMSG  025044
SFPTBL  002272 G
SKIP3   031060
SPCCNT  034400
SPED1   020744
SPED2   020767
SPED3   020773
SPED4   021003
STACHR  037614
STATER= 000001
STATUS  002524
STRCNT  002320
SVCGBL= 000000
SVCINS= 000000
SVCSUB= 000000
SVCTAG= 000000
SVCTST= 000000
S$LSYM= 010000
TABA64  021712
TABA96  022120
TABLDA  004664
TABLE1  035512
TABLE2  035546
TABSTR  037616
TAB64   036316
TAB96   036474
TICK    041142
TIME    041140
TIMOUT= 000002
TOF   = 000014
TXERR   003673
TXNOIN  003745
T$ARGC= 000001
T$CODE= 005052
T$ERRN= 000014
T$EXCP= 000000
T$FLAG= 000040
T$GMAN= 000000
T$HILI= 000377
T$LAST= 000001
T$LOLI= 000001
T$LSYM= 010000
T$LTNO= 000020
T$NEST= 000000
T$NSO = 000010
T$NS1 = 000000
T$NS2 = 000005
T$PTNU= 000000
T$SAVL= 177777
T$SEGL= 177777
T$SUBN= 000000
T$TAGL= 177777
T$TAGN= 010036
T$TEMP= 000000
T$TEST= 000020
T$TSTM= 177777
T$TSTS= 000001
T$$AUT= 010002
T$$CLE= 010005
T$$HAR= 010034
T$$HW = 010001
T$$INI= 010004
T$$PRO= 010000
T$$SOF= 010035
T$$SRV= 010033
T$$SUB= 010016
T$$SW = 010003
T$$TES= 010031
T1      007546 G
T1A     007562
T1B     010264
T1C     007566
T10     030132 G
T11     031464 G
T12     033776 G
T13     034446 G
T14     035604 G
T15     037306 G
T16     037764 G
T2      010726 G
T2664   037076
T2696   036670
T3      013704 G
T3MOV   014574
T3SET   014572
T4      016162 G
T5      022330 G
T5.1    022426
```

```
T5.2    022670          X$FALS= 000040      $F$GOO= 000400      $LSTTA= 000000      $$CASE= 000404
T6      023716 G        X$OFFS= 000400      $F$IF = 000110      $NESTL= 177777      $$DST = 000037
T7      025422 G        X$TRUE= 000020      $F$INC= 000210      $NSK0 = 000110      $$ELOC= 000402
T8      026672 G        X0      035333      $F$LOO= 000200      $NSK1 = 000120      $$ERFL= 000000
T9      027304 G        X1      035345      $F$NAM= 000160      $NSK2 = 000110      $$FLAG= 000001
UAM   = 000200 G        X11     035477      $F$NO = 000403      $NSK3 = 000110      $$FROM= 000000
UNIT    002324          X2      035357      $F$OR = 000320      $NSK4 = 000110      $$LOC = 041126
USA     002302          X3      035371      $F$RTI= 000350      $NSK5 = 000110      $$LOCN= 000000
UUT     002354          X4      035403      $F$RTN= 000300      $SAVLE= 177777      $$REG = 177777
UUTEQO  003776          X5      035415      $F$SEL= 000140      $SSK0 = 050413      $$RETU= 000000
VFUCHL  026524          X6      035427      $F$THE= 000330      $TAGLE= 177777      $$RTN1= 000000
VFUCMD  002360          X7      035441      $F$TRU= 000404      $TAGNU= 050417      $$RTN2= 000000
VFUDAT  026606          X8      035453      $F$UNT= 000130      $TEMP = 050416      $$SRC = 000027
VFUERR  023304          X9      035465      $F$WHI= 000120      $TSK0 = 050416      $$TGSV= 050145
VFULCT  024654          $BGNLE= 177777      $F$YES= 000402      $TSK1 = 050413      $$TGS1= 000001
VFUOPT  002274          $ERFLG= 000400      $IFLEV= 177777      $TSK2 = 050410      $$TGS2= 000000
VFUSEL  004045          $F$AND= 000310      $ISK0 = 000001      $TSK3 = 050400      $$TO  = 000000
VFUSE1  004133          $F$BAD= 000401      $ISK1 = 000001      $TSK4 = 050402      $$$TAG= 050000
VFUTBL  024765          $F$BLA= 000170      $ISK2 = 000001      $TSK5 = 050203      .     = 041736
WORK    003166 G        $F$CAS= 000150      $ISK3 = 000001      $TSK6 = 050204
WORK1   003170          $F$DEC= 000220      $ISK4 = 000001      $TSK7 = 050174
X     = 000040          $F$DO = 000340      $LOCTA= 177777      $$ARGC= 000000
X$ALWA= 000000          $F$FAL= 000405      $LSTIN= 000000      $$BYTE= 000403

. ABS.  041736     000

 ERRORS DETECTED:  0

CZLPLC,CZLPLC.SEQ/DOC=SVC/ML,SPMAC/ML,CZLPLC.P11
RUN-TIME: 390 369 2 SECONDS
RUN-TIME RATIO: 858/762=1.1
CORE USED:  25K  (49 PAGES)

DOCUMENT PAGES:  125
```