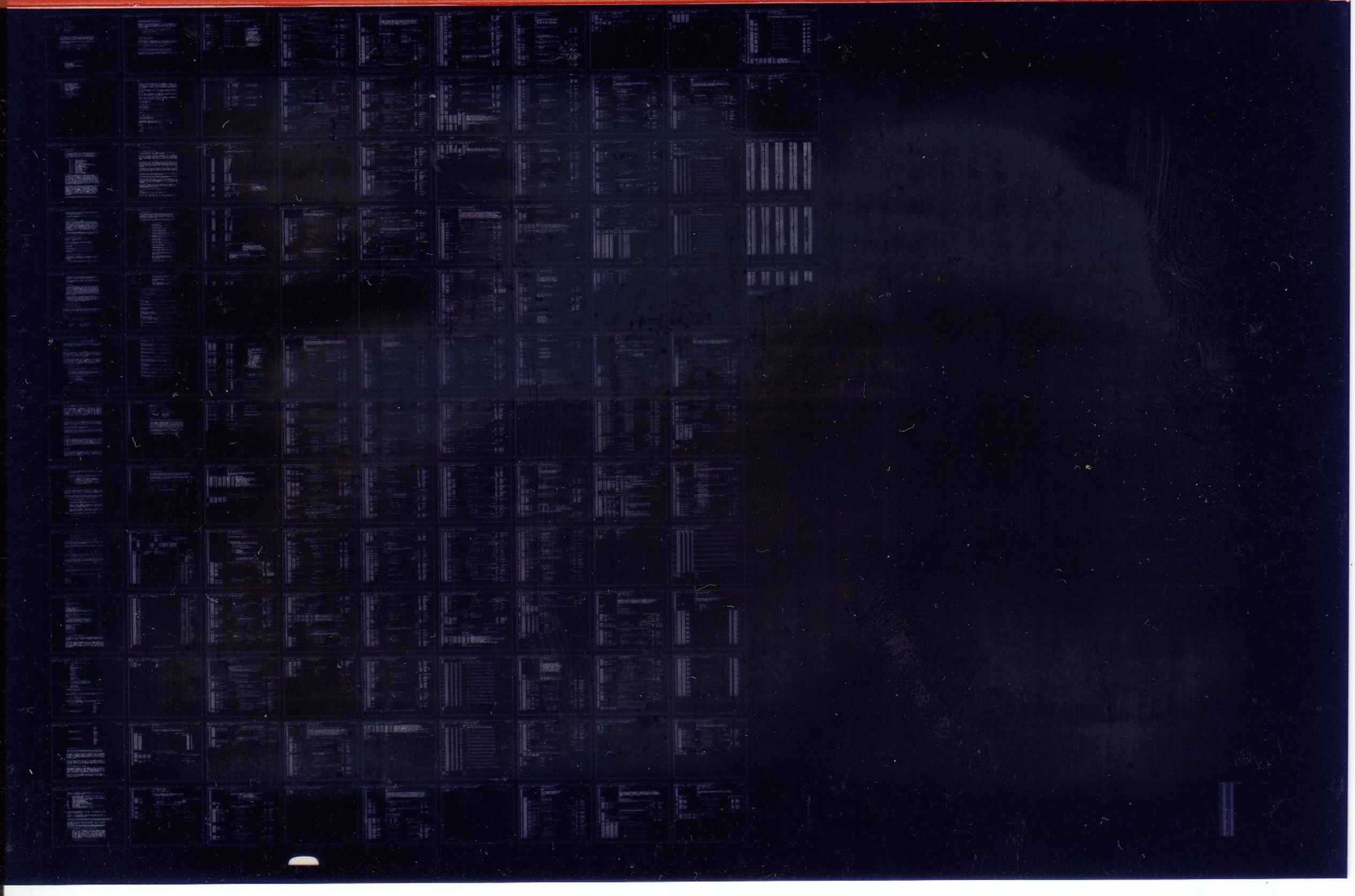
LP25

LP25 DIAG.
CZLPLA0

AH-E635A-MC

FICHE 1 OF 1

JAN 1980
digital
MADE IN USA



.REM &

IDENTIFICATION

PRODUCT CODE: AC-E634A-MC

PRODUCT NAME: CZLPLAO LP25 DIAG

MAINTAINER: SMALL SYSTEMS DIAGNOSTICS

AUTHORS: JOHN CHATALIAN
DONALD RICE

DATE: 27-SEP-79

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC PROGRAM VERIFIES PROPER PROPER OPERATION OF THE LP25 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

TEST 1 VERIFIES OPERATION OF THE INTERFACE LOGIC. TESTS 2 AND 3 ARE MANUAL INTERVENTION TESTS TO CHECK PRINTER MANUAL FUNCTIONS. TEST 4 DETERMINES THE TIME INTERVAL AUTOMATICALLY FOR CALCULATION OF THE PRINTING SPEED BY MEANS OF THE INTERNAL CLOCK. PROVISION IS INCLUDED TO PERFORM THE PRINTING SPEED MEASUREMENT MANUALLY IF A CLOCK IS NOT INSTALLED IN THE SYSTEM. TESTS 5, 6, AND 7 VERIFY PROPER OPERATION OF THE DAVFU (DIRECT ACCESS VERTICAL FORMAT UNIT) OPTION. TESTS 5 AND 6 OF THE DAVFU GROUP INVOLVE MANUAL INTERVENTION. THE DAVFU TESTS ARE OMITTED IF THE PRINTER UNDER TEST DOES NOT HAVE A DAVFU. TESTS 8 THROUGH 16 COMPRISE THE PRINTING TESTS.

THIS DIAGNOSTIC IS INTERFACED TO A FRONT-END PIECE OF SOFTWARE KNOWN AS THE DIAGNOSTIC RUNTIME SERVICES (DRS) MODULE WHICH BRINGS TOGETHER ALL THE CODE FORMERLY INCLUDED WITHIN THE DIAGNOSTIC FOR INTERFACING TO THE ENVIRONMENT. THE DIAGNOSTIC SUPERVISOR ALLOWS THE OPERATOR TO SPECIFY HARDWARE/SOFTWARE PARAMETERS AND TO PERFORM TEST AND UNIT SELECTION. IN THE DRS COMMAND MODE, THE OPERATOR CAN ISSUE COMMANDS TO CONTROL THE OPERATION OF THE DIAGNOSTIC AND CAN, BY MEANS OF SWITCHES ON THESE COMMANDS, INVOKE DIFFERENT FLAGS WHICH REPLACE THE HARDWARE SWITCH REGISTER WHICH IS NOT ACCESSIBLE UNDER THE SUPERVISOR.

SINCE THE PROGRAM ONLY MONITORS THE CURRENT CONDITION OF THE READY AND DEMAND LINES AND DOES NOT RECEIVE ANY OTHER STATUS

INFORMATION FROM THE LINE PRINTER, THE OPERATOR MUST EXAMINE THE PRINT PATTERNS PRODUCED BY THE VARIOUS TEST ROUTINES TO VERIFY PROPER PRINTER OPERATION.

THIS DIAGNOSTIC PROGRAM HAS THE CAPABILITY TO TEST UP TO 16 LINE PRINTERS AT THE SAME TIME.

1.2 SYSTEM REQUIREMENTS

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

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

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

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

DATAPRODUCTS 300 LPM LINE PRINTER FIELD MAINTENANCE
GUIDE (PRELIMINARY)

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

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

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE SIX STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP PROCEDURES. THE START COMMAND SHOULD NOT SPECIFY AN ADDRESS, BECAUSE THE DIAGNOSTIC HAS THE PROPER TRANSFER ADDRESS CODED INTO IT.

WHEN THIS DIAGNOSTIC IS STARTED, THE FOLLOWING STEPS WILL OCCUR:

****
* STEP 1 *
****

A SHORT SERIES OF "HARDCORE QUESTIONS" WILL BE ASKED:

QUESTION	MEANING
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
50HZ (L) N ?	IS THE POWER 50 CYCLES (AS IN EUROPE)?
LSI (L) N ?	IS MACHINE AN LSI?

```
LPT (L) N ?          IS THERE A LINE PRINTER?
MEM (K) (D) 16 ?     HOW MANY K OF MEMORY ARE THERE?
```

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

****
* STEP 2 *
****

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT "DS-B>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE XXDP "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE XXDP DOT PROMPT. THIS "START" COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF THESE ARE SET FORTH IN '2.3 DETAILS OF COMMANDS AND SYNTAX". HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

```
STA/PASS:1/FLAGS:HOE
```

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DS-B>" LEVEL NEED TO BE TYPED.

2. THE "PASS" SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.

3. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

```
LOE     LOOP ONE ERROR
HOE     HALT ON ERROR
IER     INHIBIT ERROR PRINTOUT
```

   THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

****
* STEP 3 *
****

WHEN YOU HAVE TYPED IN A "START" COMMAND, THE DIAGNOSTIC WILL COME

BACK WITH THE QUESTION ''# UNITS?'' TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE ''HEADER'' STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS ''HEADER'' STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

****
* STEP 4 *
****

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE ''HARDWARE QUESTIONS''. THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED ''HARDWARE P-TABLES''. ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS WILL IN THE FUTURE NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES. INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

****
* STEP 5 *
****

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS FOR ALL THE UNITS, YOU WILL BE ASKED ''CHANGE SW?'' IF YOU WANT TO BE ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS PROGRAM, TYPE 'Y''. IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE QUESTIONS, TYPE 'N''. IF YOU TYPE 'Y'' YOU WILL BE ASKED THE SOFTWARE QUESTIONS, AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.

****
* STEP 6 *
****

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE START COMMAND. CONSIDER THE POSSIBILITIES:

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DS-B>).

2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

   HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.
   LOE SET: THE DIAGNOSTIC WILL LOOP ENLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.
   NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURED.

2.1.2 SAMPLE RUN-THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND "STA/PASS:1/FLAGS:HOE". THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE REISSUED.

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN.

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 2, 3, 4, 5, AND 6 AGAIN)

2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)

3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED.

4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT ON ERROR).

THE MOST TYPICAL THING TO DO HERE IS TO ISSUE THE PROCEED, BUT WITH DIFFERENT FLAG SETTINGS. PROBABLY YOU WOULD WANT TO SAY

```
PRO/FLAGS:IER:LOE:HOE=0
```

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG

2. TURN ON THE LOE FLAG

3. TURN OFF THE HOE FLAG

4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

THE DIAGNOSTIC WILL NOW LOOP ON THE BLOCK OF CODE THAT DETECTED AND REPORTED THE ERROR, BUT NO ERROR PRINTOUT WILL OCCUR. THUS YOU CAN STUDY THE ERROR OR SCOPE IT OR WHATEVER.

WHEN YOU'VE SEEN ENOUGH, YOU MAY HIT CONTROL/C. THIS WILL TAKE YOU OUT OF THE LOOP AND PUT YOU BACK INTO COMMAND MODE. YOU NOW HAVE THREE CHOICES:

1. START

2. RESTART

3. CONTINUE

LET'S SAY YOU'VE REPAIRED THE DEFECT FOUND ABOVE AND WANT TO FINISH RUNNING THE DIAGNOSTIC. YOU WOULD TYPE

CON/FLAGS:HOE:IER=0:LOE=0

THIS WILL RESTORE THE FLAGS TO THEIR ORIGINAL VALUES AND RESUME EXECUTION AT THE BEGINNING OF THE HARDWARE TEST YOU WERE IN. IF THE ERROR DOES NOT RECUR, THE EXECUTION WILL FLOW RIGHT ON THRU TO THE NEXT ERROR OR TO END OF PASS.

IF AT END OF PASS YOU WANT TO RUN THE DIAGNOSTIC AGAIN, YOU HAVE TWO CHOICES:

1. START

2. RESTART

YOU WOULD CHOOSE ONE, DEPENDING ON WHETHER YOU WANTED TO ANSWER THE HARDWARE QUESTIONS AGAIN.

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS:

```
                                                        BY
                                                        WHOM
                                                        ENTERED:
.R CZLPLA                                               O
CZLPLA                                                  D
L-CLK (L) N ? Y                                         D,O
50HZ (L) N ?                                            D
LSI (L) N ?                                             D
LPT (L) N ?                                             D
MEM (K) (D) 16 ?                                        D
```

```
DS-B>STA/PASS:1/FLAGS:HOE                                          D,O

# UNITS (D) ? 1                                                    D,O

UNIT 1                                                             D
LP11 ADDRESS (O) 177514 ?                                          D
INTERRUPT VECTOR (O) 200 ?                                         D

CHANGE SW (L) ? Y                                                  D,O

RUN MANUAL INTERVENTION TESTS (L) ? Y                              D,O
96 CHARACTER BAND (L) ? N
DAVFU OPTION INSTALLED (L) N ?                                     D
PERFORM MANUAL PRINTING SPEED MEASUREMENT (L) N ? Y                D,O
DESIRED TIME INTERVAL FOR PRINTING SPEED CALCULATION (D) 4 ?       D
TESTING IN U.S.A. (L) ? Y

LP25 HRD ERR   00009 TST 004 SUB 000 PC: 003604                    D
ERROR AT CSR 177514 UNIT  1                                        D
ERR HLT                                                            D
DS-B>PRO/FLAGS:IER:LOE:HOE=0                                       D,O

**************************************************
AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE
ERROR WITHOUT PRINTING ANYTHING.  YOU CAN SCOPE
THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT.
**************************************************

^C                                                                 O
DS-B>CON/FLAGS:HOE:IER:LOE=0                                       D,O
CHANGE SW (L) ? N                                                  D,O
CZLPLA EOP 1                                                       D
DS-B>RESTART/PASS:1                                                D,O
CHANGE SW (L) ? N                                                  D,O
------------------
------------------
------------------
------------------
------------------
```

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION 'BIN'' INSTEAD OF 'BIC''. THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND ''CCI'' ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B>. THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION 'BIC''.

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```
.R UPD2
RESTART:  XXXXXX
*CLR
*LOAD DIAG.BIN
XFER:200   CORE:0,60602
*START 200
L-CLK (L) N ?
------------
------------
------------
DS-B>CCI
# UNITS (D) ? 4
------------
------------
------------
CHANGE SW (L) ? N
PTAB END:  60632

****************************************
*AT THIS POINT THE MACHINE HALTS AND*
*YOU MUST RESTART AT ADDRESS  XXXXXX*
****************************************

*HICORE 60632
CORE:  0,60632
*DUMP DK0: DIAG.BIC
```

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN FILE CONTAINING THE XXDP COMMAND

```
.R DIAG.BIC
```

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

THERE ARE FOUR WAYS OF ENTERING DIAGNOSTIC COMMAND MODE, AND DIFFERENT SUBSETS OF THE DIAG COMMAND SET ARE AVAILABLE WITH EACH:

	HOW ENTERED	LEGAL COMMANDS
1.	OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS
2.	DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT

DISPLAY
FLAGS
ZFLAGS

3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C

START
RESTART
CONTINUE
PRINT
DISPLAY
FLAGS
ZFLAGS

4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET

START
RESTART
CONTINUE
PROCEED
PRINT
DISPLAY
FLAGS
ZFLAGS

2.3.2 COMMAND SYNTAX

```
**************************************************************************
STA(RT)/TESTS:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR
**************************************************************************
```

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE ''# UNITS?'' IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED 'RUN DIAGNOSTIC'' B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C.

AFTER THE OPERATOR RESPONDS TO ''# UNITS?'', THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS ''CHANGE SW?'' IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

''TEST-LIST'' IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

''PASS-CNT'' IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUMBITTED. THE DEFAULT IS NON-ENDING EXECUTION.

'FLAG-LIST'' IS A SEQUENCE OF ELEMENTS OF THE
FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG>
HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.

'EOP-INCR'' IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS.

****
RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/UNITS:UNIT-LIST
****

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. HOWEVER, NEW P-TABLES ARE NOT BUILT. INSTEAD, THE ONES IN CORE ARE USED.
THE QUESTION ''CHANGE SW?'' IS ASKED,AND THE ANSWERS IF GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. 'UNIT-LIST'' IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING WHICH UNITS ARE TO BE TESTED. THE LOGICAL UNIT NUMBER DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING TO THE ORDER IN WHICH THEY WERE BUILT. THE UNITS SPECIFIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND. THE UNIT-LIST DEFAULTS TO ''ALL THAT HAVE NOT BEEN DROPPED BY OPERATOR COMMAND''. THE EFFECT OF THE UNIT-LIST LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO ''ALL'') OR THE NEXT RESTART.

2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

B 2

****
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>
****

COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART

2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

****
PRO(CEED)/FLAGS:<FLAG-LIST>
****

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

****
CCI/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR
****

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A "START" (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE: THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A "RESTART" (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND ".R DIAG". THE COMMAND PROMPT "DS-B>" WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

****
DRO(P)/UNITS:UNIT-LIST
****

THIS COMMAND IS NOT UTILIZED IN THIS DIAGNOSTIC.

****
ADD/UNITS:UNIT-LIST
****

THIS COMMAND IS NOT UTILIZED IN THIS DIAGNOSTIC.

****
PRI(NT)
****

THIS COMMAND IS NOT UTILIZED IN THIS DIAGNOSTIC.

****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED.
OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

****
FLA(GS)
****

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

****
ZFL(AGS)
****

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO-ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 4 LP25'S, AND THAT THERE ARE TWO HARDWARE PARAMETERS FOR EACH (TWO SLOTS IN THE P-TABLE, TWO HARDWARE QUESTIONS IN THE DIALOGUE).

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 4
UNIT 1
LP11 ADDRESS: (O) ? 177514, 177520, 177524, 177530
INTERRUPT VECTOR: (O) ? 200, 210, 220, 230
```

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 4 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION.

3.0 ERROR INFORMATION

ERRORS ARE REPORTED ON THE CONSOLE TERMINAL WHEN AN ERROR IS DETECTED BY THE DIAGNOSTIC. THE DEVICE ADDRESS OF THE FAILING UNIT AND THE NATURE OF THE ERROR IS GIVEN IN THE ERROR MESSAGE. OTHER INFORMATION INCLUDED ARE THE TEST NUMBER AND THE PROGRAM COUNTER ADDRESS WHERE THE ERROR OCCURRED.

THE ERROR MESSAGES AID IN IDENTIFYING THE PROBLEM. THE FOLLOWING LIST PROVIDES A BRIEF DESCRIPTION OF EACH OF THE ERRORS.

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  CZLPLA0 LP25 DIAGNOSTIC
                        .ENABL AMA
                        .SBTTL  IDENTIFICATION

                        ; PRODUCT CODE:          AC-E634A-MC

                        ; PRODUCT NAME:          CZLPLA0 LP25 DIAG

                        ; MAINTAINER:            SMALL SYSTEMS DIAGNOSTICS

                        ; AUTHORS:               JOHN CHATALIAN
                        ;                        DONALD RICE
                        ;                        RALPH SCHAUBER

                        ; DATE                  . 27-SEP-79

                        ;COPYRIGHT (C) 1979
                        ;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.
```

```
1055
1056                                    ;++
1057                                    ; FUNCTIONAL DESCRIPTION
1058                                    ;
1059                                    ;THIS DIAGNOSTIC PROGRAM VERIFIES PROPER OPERATION OF THE LP25 LINE PRINTER
1060                                    ;AND ITS ASSOCIATED M7258 CONTROL UNIT WHICH IS INTERFACED TO A PDP-11 CPU.
1061                                    ;THE DIAGNOSTIC HAS PROVISION TO TEST UP TO SIXTEEN UNITS AT A TIME.
1062                                    ;
1063                                    ;THE PROGRAM CONSISTS OF SIXTEEN TESTS, THREE OF WHICH ARE FOR THE DAVFU OPTION.
1064                                    ;THREE OF THE PRINTER TESTS INVOLVE MANUAL INTERVENTION. TWO OF THE DAVFU TESTS
1065                                    ;REQUIRE INTERVENTION BY THE OPERATOR.
1066                                    ;
1067                                    ;THE PROGRAM IS COMPATIBLE TO THE PDP-11 DIAGNOSTIC SUPERVISOR, ACT/SLIDE, AND
1068                                    ;XXDP+.
1069                                    ;
1070                                    ;--
1071
1072
1073
1074                                    ; VERSION       A-0    27-SEP-79        R. SCHAUBER
1075
1076
1077                                    ; HISTORY       REV. A-0   INITIAL RELEASE
1078                                    ;               REV-C SUPERVISOR / XXDP+ COMPATABLE
1079
1080
```

```
                                        .TITLE  CZLPLA0 LP25 TEST
                                .SBTTL PROGRAM HEADER

                                        .MCALL  SVC
000000'                                 SVC                             ;INITIALIZE SUPERVISOR MACROS
                                        .MCALL  STRUCT
000000'                                 STRUCT                          ;STRUCTURED MACRO PACKAGE
        000001                          $LSTIN=         1
        000001                          $LSTTAG=        1

        000001                  SVCINS= 1       ;LIST INSTRUCTIONS, SHIFTED RIGHT
        000001                  SVCTST= 1       ;LIST TEST TAGS   , SHIFTED RIGHT
        000001                  SVCSUB= 1       ;LIST SUBTEST TAGS, SHIFTED RIGHT
        000001                  SVCGBL= 1       ;LIST GLOBAL TAGS , SHIFTED RIGHT
        000001                  SVCTAG= 1       ;LIST OTHER TAGS  , SHIFTED RIGHT

                                .ENABL  ABS,AMA
        002000                          .=2000

002000                                  BGNMOD
002000                                  POINTER BGNSW,BGNSFT

002000                                  HEADER  CZLPL,A,0,60,1,340
(4)  002000                                                             L$NAME::
(4)  002000     103                                                             .ASCII /C/
(4)  002001     132                                                             .ASCII /Z/
(4)  002002     114                                                             .ASCII /L/
(4)  002003     120                                                             .ASCII /P/
(4)  002004     114                                                             .ASCII /L/
(6)  002005     000                                                             .BYTE   0
(6)  002006     000                                                             .BYTE   0
(5)  002007     000                                                             .BYTE   0
(5)  002010                                                             L$REV::
(4)  002010     101                                                             .ASCII  /A/
(5)  002011                                                             L$DEPO::
(4)  002011     060                                                             .ASCII  /0/
(5)  002012                                                             L$UNIT::
(4)  002012  000000                                                             .WORD   0
(5)  002014                                                             L$TIML::
(4)  002014  000060                                                             .WORD   60
(5)  002016                                                             L$HPCP::
(4)  002016  032276                                                             .WORD   L$HARD
(5)  002020                                                             L$SPCP::
(4)  002020  032356                                                             .WORD   L$SOFT
(5)  002022                                                             L$HPTP::
(4)  002022  002232                                                             .WORD   L$HW
(5)  002024                                                             L$SPTP::
(4)  002024  002244                                                             .WORD   L$SW
(5)  002026                                                             L$LADP::
(4)  002026  033020                                                             .WORD   L$LAST
(5)  002030                                                             L$STA::
(4)  002030  000000                                                             .WORD   0
(5)  002032                                                             L$CO::
(4)  002032  000000                                                             .WORD   0
(5)  002034                                                             L$DTYP::
(4)  002034  000001                                                             .WORD   1
```

```
(5)  002036
(4)  002036  000000
(5)  002040
(4)  002040  002132
(5)  002042
(4)  002042  000340
(5)  002044
(4)  002044  000000
(5)  002046
(4)  002046  000000
(5)  002050
(4)  002050     003
(3)  002051     003
(5)  002052
(4)  002052  000000
(5)  002054  000000
(5)  002056
(4)  002056  000000
(5)  002060
(4)  002060  002222
(5)  002062
(4)  002062  000000
(5)  002064
(4)  002064  000000
(5)  002066
(4)  002066  000000
(5)  002070
(4)  002070  000000
(5)  002072
(4)  002072  000000
(5)  002074
(4)  002074  000000
(5)  002076
(4)  002076  002172
(5)  002100
(4)  002100  104035
(5)  002102
(4)  002102  000000
(5)  002104
(4)  002104  005020
(5)  002106
(4)  002106  006266
(5)  002110
(4)  002110  002236
(5)  002112
(4)  002112  002122
(5)  002114
(4)  002114  000000
(5)  002116
(4)  002116  000000
(5)  002120
(4)  002120  000000
1105
1106
1107                    ;
1108                    ; THE FOLLOWING IS A LOAD PROTECTION TABLE
                        ;

L$APT::
        .WORD   0
L$DTP::
        .WORD   L$DISPATCH
L$PRIO::
        .WORD   340
L$ENVI::
        .WORD   0
L$EXP1::
        .WORD   0
L$MREV::
        .BYTE   C$REVISION
        .BYTE   C$EDIT
L$EF::
        .WORD   0
        .WORD   0
L$SPC::
        .WORD   0
L$DEVP::
        .WORD   L$DVTYP
L$REPP::
        .WORD   0
L$EXP4::
        .WORD   0
L$EXP5::
        .WORD   0
L$AUT::
        .WORD   0
L$DUT::
        .WORD   0
L$LUN::
        .WORD   0
L$DESP::
        .WORD   L$DESC
L$LOAD::
        EMT     E$LOAD
L$ETP::
        .WORD   0
L$ICP::
        .WORD   L$INIT
L$CCP::
        .WORD   L$CLEAN
L$ACP::
        .WORD   L$AUTO
L$PRT::
        .WORD   L$PROT
L$TEST::
        .WORD   0
L$DLY::
        .WORD   0
L$HIME::
        .WORD   0
```

```
 1109  002122                                  BGNPROT
  (3)  002122                                                  L$PROT::
 1110  002122  000000                          .WORD   0
 1111  002124  177777                          .WORD   -1
 1112  002126  177777                          .WORD   -1
 1113  002130                                  ENDPROT
```

```
1115                                    .SBTTL  DISPATCH TABLE
1116
1117                                    ;++
1118                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1119                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1120                                    ;--
1121
1122  002130                                    DISPATCH        16              ;X= NUMBER OF TESTS
  (4) 002130  000020                                                                    .WORD   16
  (3) 002132                                                                  L$DISPATCH::
  (6) 002132  006406                                                                    .WORD   T1
  (6) 002134  007404                                                                    .WORD   T2
  (6) 002136  011452                                                                    .WORD   T3
  (6) 002140  013646                                                                    .WORD   T4
  (6) 002142  016330                                                                    .WORD   T5
  (6) 002144  017634                                                                    .WORD   T6
  (6) 002146  021226                                                                    .WORD   T7
  (6) 002150  022360                                                                    .WORD   T8
  (6) 002152  022736                                                                    .WORD   T9
  (6) 002154  023246                                                                    .WORD   T10
  (6) 002156  024614                                                                    .WORD   T11
  (6) 002160  026330                                                                    .WORD   T12
  (6) 002162  026712                                                                    .WORD   T13
  (6) 002164  027334                                                                    .WORD   T14
  (6) 002166  030212                                                                    .WORD   T15
  (6) 002170  030636                                                                    .WORD   T16
1123                                    ;
1124                                    ;FOR USE ON REVISION C OF THE SUPERVISOR
1125                                    ;
1126  002172                                    DESCRIP         <LP25 PRINTER DIAGNOSTIC>
  (4) 002172                                                                  L$DESC::
  (3) 002172  050114  032462  050040                                                .ASCIZ  /LP25 PRINTER DI
  (3) 002200  044522  052116  051105
  (3) 002206  042040  040511  047107
  (3) 002214  051517  044524  000103
  (2)                                                                               .EVEN
1127  002222                                    DEVTYP          <LP25>
  (4) 002222                                                                  L$DVTYP::
  (3) 002222  050114  032462      000                                               .ASCIZ  /LP25/
  (2)       002230                                                                  .EVEN
1128
1129
1130
```

```
                        .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.
                        ;--

002230                          BGNHW   DFPTBL
(3) 002230  000002                                                      .WORD   L10001-L$HW/2
(3) 002232                                                          L$HW::
(3) 002232                                                          DFPTBL::
002232  177514                  .WORD   177514              ;LP25 REGISTER ADDRESS
002234  000200                  .WORD   200                 ;LP25 INTERRUPT VECTOR
                        ;
                        ;INTERRUPT VECTOR PRIORITY IS 4 AND CANNOT BE CHANGED


002236                          ENDHW
(3) 002236                                                          L10001:


002236                          BGNAUTO
(3) 002236                                                          L$AUTO::

002236  000240                  NOP             ; NOT USED

002240                          ENDAUTO
(3) 002240                                                          L10002:
(3) 002240  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.
                        ;--

002242                          BGNSW   SFPTBL
(3) 002242  000007                                              .WORD   L10003-L$SW/2
(3) 002244                                                      L$SW::
(3) 002244                                                      SFPTBL::

002244  000000          INHINT: .WORD   0               ;0 IF NO INTERVENTION TESTS
                                                        ;1 IF MANUAL INTERVENTION TESTS
                                                        ;DEFAULT IS NO
002246  000001          CHRBND: .WORD   1               ;0 IF 64 CHARACTER BAND
                                                        ;1 IF 96 CHARACTER BAND
                                                        ;96 CHARACTER BAND DEFAULT
002250  000000          VFUOPT: .WORD   0               ;0 IF NO DAVFU OPTION
                                                        ;1 IF DAVFU OPTION INSTALLED
                                                        ;NO DAVFU DEFAULT
002252  000000          MANSPD: .WORD   0               ;0 FOR AUTOMATIC PRINT SPEED
                                                        ;1 FOR MANUAL PRINT SPEED TEST
                                                        ;AUTOMATIC DEFAULT VALUE
002254  000004          PERIOD: .WORD   4               ;OPERATOR TO SELECT TIMING VALUE
                                                        ;FROM 4 TO 60 SECONDS. INITIAL
                                                        ;DEFAULT VALUE IS 4 SECONDS.

002256  000001          USA:    .WORD   1               ; 1 FOR TESTING IN U.S.A.
                                                        ; 0 FOR TESTING IN G.B./EUROPE
                                                        ; * DIFFERENT BAND PATTERNS *
002260  000005          MAXERR: .WORD   5               ; AUTODROP ERROR COUNT
                        ; IF ERROR COUNT EXCEEDS MAXERR THE UNIT WILL BE DROPPED FROM TEST

002262                          ENDSW
(3) 002262                                                      L10003:
```

```
                        .SBTTL  I/O MACRO DEFINITIONS

                        .MACRO  OUTPUT  ADD,BFCNT,ERR
                                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
                                JSR     PC,IOCTRL               ;CALL THE DRIVER
                        .ENDM   OUTPUT


                        .MACRO  OUTPUI  ADD,BFCNT,ERR,UNIT
                                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
                                MOV     UNIT,PRINTR             ; SUPPLY UNIT NUMBER
                                JSR     PC,IOCTRL               ;CALL THE DRIVER
                        .ENDM


002262                          ENDMOD
```

```
                                .SBTTL  GLOBAL AREAS

002262                                  BGNMOD

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

002262                                  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
```

```
(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
1236
1240           ;
1241           ;PRIORITY LEVEL DEFINITIONS
1242           ;
1243   000340  PRI07== 340
1244   000300  PRI06== 300
1245   000240  PRI05== 240
1246   000200  PRI04== 200
1247   000140  PRI03== 140
1248   000100  PRI02== 100
1249   000040  PRI01== 40
1250   000000  PRI00== 0
1251
1252           ;
1253           ;GLOBAL ERROR CODES FOR USE BY GENERAL ERROR ROUTINE
1254           ;
1255   000001  STATER= 1                    ;TRANSMITTER STATUS ERROR IN OUTPUT
1256   000002  TIMOUT= 2                    ;TIMEOUT ERROR IN IO DRIVER MODULE
1257                                        ;THIS ERROR INDICATES THE LAST CHARACTER
1258                                        ;WAS NOT TRANSMITTED WITHIN A GIVEN TIME
1259   000003  NOINTR= 3                    ;GROSS TIME OUT ERROR. THE SPECIFIED DID NOT
1260                                        ;INTERRRUPT. THEREFORE IO DRIVER MODULE WAS
1261                                        ;NOT CALLED
1262           ;
1263           .SBTTL  GENERAL REGISTER USAGE DEFINITIONS
1264           ;
1265           ;R0     RESERVED FOR USE BY THE MACRO PACKAGES
1266           ;R1     MAXIMUM NUMBER OF UNITS TO TEST L$UNIT-1
1267           ;R2     UNIT NUMBER BY 2. USED TO CALCULATE OFFSET INTO PROPER
1268           ;       PRINTER TABLE
1269           ;R3     TEMPORARY STORAGE
1270           ;R4          "     "
1271           ;R5
1272           ;R6     STACK POINTER
```

```
1273                                    ;R7     PROGRAM COUNTER
1274                                    ;
1275
1276
1277                                    ;
1278                                    ; LP STATUS TABLE BIT DEFINITIONS
1279                                    ;
1280            100000                  ACTIVE = 100000
1281            000200                  DROPED = 000200
1282            002000                  ERROR  = 002000
1283            000100                  OVFLO  = 000100
```

```
1285                                    .SBTTL  GLOBAL DATA SECTION
1286
1287
1288
1289  002262  000000                    FLAG:   .WORD   0               ;<CR> FLAG FOR USE BY SUPERVISOR
1290  002264  000000                    LINCNT: .WORD   0               ;LINE COUNTER
1291  002266  000000                    LSTCNT: .WORD   0
1292  002270  000000                    CCNT:   .WORD   0
1293  002272  000000                    STRCNT: .WORD   0
1294  002274  000000                    CHRGEN: .WORD   0
1295  002276  000000                    UNIT:   .WORD   0               ;UNIT COUNTER FOR SINGLE UNIT TESTING
1296  002300  000000                    LUNIT:  .WORD   0               ;UNIT COUNTER FOR ERRORS
1297                                                                    ;AND TESTS NOT USING THE OUTPUT
1298                                                                    ;MACROS.
1299  002302  000000                    PTABAD: .WORD   0               ;P-TABLE ADDRESS RETURNED BY GPHARD
1300  002304  000000                    PRINTR: .WORD   0               ;SELECTED LINE NO.
1301                                                                    ;MACRO
1302  002306  000000                    CLKTYP: .WORD   0               ;CLOCK TYPE CONTROL WORD
1303                                                                    ;1= NO CLOCK AVAILABLE
1304                                                                    ;2= KW11-L LINE CLOCK
1305                                                                    ;3= KW11-P PROGRAMABLE CLOCK
1306  002310  000000                    CLOCKP: .WORD   0               ; CLOCK P-TABLE ADDRESS
1307  002312  000000                    CLKCSR: .WORD   0               ;CLOCK CSR ADDRESS
1308  002314  000000                    CLKSET: .WORD   0               ; CLOCK TIME SET REG ADDRESS
1309  002316  000000                    CLKVEC: .WORD   0               ;CLOCK VECTOR ADDRESS
1310  002320  000000                    CLKENA: .WORD   0               ;CLOCK ENABLE BITS
1311  002322  000000                    ERRCOD: .WORD   0               ;ERROR CODE TYPE FOR GENERAL
1312                                                                    ;ERROR ROUTINE
1313  002324  000000                    ERRFLG: .WORD 0                 ;EXPECTED ERROR INDICATOR
1314  002326  000000                    UUT:    .WORD   0               ; # UNITS ACTUALLY UNDER TEST
1315                                                                    ;EXITS BACK TO IO DRIVER EQUAL
1316                                                                    ;1 IF ERROR WAS EXPECTED.
1317
1318  002330  000000                    INDEX:  .WORD   0
1319  002332  000000                    VFUCMD: .WORD   0
1320                                    ;
1321                                    ;MACRO  VARIABLES
1322
1323  002334  000000                    BUFADD: .WORD   0               ;BUFFER ADDRESS OF DATA TO BE SENT
1324                                                                    ;TO THE LP25
1325  002336  000000                    BUFCNT: .WORD   0               ;NUMBER OF BYTES TO TRANSFER
1326                                    ;
1327                                    ;
1328                                    ;
1329                                    ;LP25 PARAMETER WORD TABLES
1330                                    ;
1331  002340  000020                    LPCSR:  .REPT   16.             ; ADDRESS OF CSR FOR EACH LP11
1332                                            .WORD   0
1333                                            .ENDR
1334  002400  000016                    LPVEC:  .REPT   16              ; INTERRUPT VECTOR ADDRESS
1335                                            .WORD   0
1336                                            .ENDR
1337  002434  000020                    LPBUF:  .REPT 16.               ; DATA BUFFER REGISTER ADDRESS
1338                                            .WORD   0
1339                                            .ENDR
1340  002474  000020                    STATUS: .REPT 16.               ; UNIT STATUS
```

```
1341                                            .WORD   0
1342                                            .ENDR
1343  002534  000020                  CURADD:   .REPT 16.             ; CURRENT ADDRESS OF OUTPUT DATA
1344                                            .WORD   0
1345                                            .ENDR
1346  002574  000020                  CURCNT:   .REPT 16.             ; CURRENT COUNT TO OUTPUT
1347                                            .WORD   -1
1348                                            .ENDR
1349  002634  000020                  LPINTR:   .REPT 16.             ; INTERRUPT ROUTINE ADDRESS
1350                                            .WORD   0
1351                                            .ENDR
1352  002674  000000                  ERRSVC:   .WORD   0             ; ERROR ROUTINE DISPATCH ADDRESS
1353  002676  000020                  ERRTBL::          .REPT 16.     ; ERROR COUNT FOR EACH UNIT
1354                                            .WORD   0
1355                                            .ENDR
1356
1357  002736  000000                  WORK::    .WORD   0             ; WORK AREA
1358  002740  000000                  WORK1:    .WORD   0
1359
1360
1361
1362                                  .SBTTL  OUTPUT BUFFER
1363                                  ;
1364                                  ;140 BYTES IS RESERVED FOR THE OUTPUT BUFFER AREA
1365                                  ;
1366
1367  002742  000226                  OUTBUF:   .REPT   150.
1368                                            .BYTE   0
1369                                            .ENDR
1370
```

```
1372                                    .SBTTL  GLOBAL TEXT SECTION
1373
1374                                    .NLIST BEX
1375                                    ;++
1376                                    ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1377                                    ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1378                                    ; MORE THAN ONE TEST.
1379                                    ;--
1380  003170  051120  047111  042524    CSRERR: .ASCIZ /PRINTER ERROR/
1381  003206  051120  047111  042524    RDYERR: .ASCIZ /PRINTER NOT READY/
1382  003230  040520  042520  020122    PAPSWI: .ASCIZ /PAPER LOW INTERLOCK SWITCH FAILURE/
1383  003273     110  046501  042515    BNKSWI: .ASCIZ /HAMMER BANK INTERLOCK SWITCH FAILURE/
1384  003340  044103  051101  041501    BNDSWI: .ASCIZ  /CHARACTER BAND INTERLOCK SWITCH FAILURE/
1385  003410  051124  047101  046523    INTER1: .ASCIZ  /TRANSMIT INTERRUPT TIMEOUT/
1386  003443     120  044522  052116    TXERR:  .ASCIZ  /PRINTER STATUS ERROR/
1387  003470  052517  050124  052125    OUTTIM: .ASCIZ  /OUTPUT TIMEOUT ERROR/
1388  003515     125  044516  020124    TXNOIN: .ASCIZ  /UNIT FAILED TO INTERRUPT/
1389  003546  046101  020114  047125    UUTEQ0: .ASCIZ  /ALL UNITS HAVE BEEN DROPPED..RESTART../
1390  003615     045  022516  044501    VFUSEL: .ASCII /%N%AINSURE THAT VFU-FLS SWITCH ON EACH UNIT IS IN THE /
1391  003703     045  022516  021101    VFUSE1: .ASCIZ /%N%A'VFU'' POSITION.%N/
1392        003732                      .EVEN
1393
1394
1395                                    ;
1396                                    ;
1397
1398                                    .LIST BEX
1399                                    ;
1400                                    ; FORMAT STATEMENTS USED IN PRINT CALLS
1401                                    ;
1402
1403  003732  040445  050114  030461    LPDROP: .ASCIZ  /%ALP11 UNIT %D2%A DROPPED FROM TEST%N/
      003740  052440  044516  020124
      003746  042045  022462  020101
      003754  051104  050117  042520
      003762  020104  051106  046517
      003770  052040  051505  022524
      003776  000116
1404
1405
1406
1407
```

```
                                .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.
                                ;

004000                          LPERR:  SELECT ERRCOD OF 3 VERIFY       ;SELECT PROPER MESSAGE FORMAT
(2)  004000  013746  002322                                                     MOV     ERRCOD,-(SP)
(4)  004004  003403                                                             BLE     50000$
(3)  004006  021627  000003                                                     CMP     (SP),#3
(6)  004012  003402                                                             BLE     50001$
(3)  004014                                                             50000$:
(3)  004014  012716  000004                                                     MOV     #4,(SP)
(3)  004020                                                             50001$:
(2)  004020  006316                                                             ASL     (SP)
(2)  004022  060716                                                             ADD     PC,(SP)
(2)  004024  063607                                                             ADD     @(SP)+,PC
(3)  004026                                                             50002$:
(4)  004026  000010                                                             .WORD   50006$-50002$
(4)  004030  000036                                                             .WORD   50005$-50002$
(4)  004032  000064                                                             .WORD   50004$-50002$
(3)  004034  000110                                                             .WORD   50003$-50002$

004036                                  CASE 1                  ;STATUS ERROR
(4)  004036                                                             50006$:
004036                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
(6)  004036  005262  002676                                                     INC     ERRTBL(R2)
004042                                  LET L$LUN := R2 SHIFT -1
(4)  004042  010237  002074                                                     MOV     R2,L$LUN
(7)  004046  006237  002074                                                     ASR     L$LUN
```

```
1446  004052                          ERRHRD  14,TXERR
 (4)  004052  104456                                                    TRAP    C$ERHRD
 (5)  004054  000016                                                    .WORD   14
 (5)  004056  003443                                                    .WORD   TXERR
 (5)  004060  000000                                                    .WORD   0
1447
1448  004062                          CASE 2                  ;OUTPUT TIMEOUT ERROR
 (4)  004062  000425                                                    BR      50003$
 (4)  004064                                                  50005$:
1449  004064                          LET ERRTBL(R2) := ERRTBL(R2) + #1
 (6)  004064  005262  002676                                            INC     ERRTBL(R2)
1450  004070                          LET L$LUN := R2 SHIFT -1
 (4)  004070  010237  002074                                            MOV     R2,L$LUN
 (7)  004074  006237  002074                                            ASR     L$LUN
1451  004100                          ERRHRD  15,OUTTIM       ;
 (4)  004100  104456                                                    TRAP    C$ERHRD
 (5)  004102  000017                                                    .WORD   15
 (5)  004104  003470                                                    .WORD   OUTTIM
 (5)  004106  000000                                                    .WORD   0
1452
1453  004110                          CASE 3
 (4)  004110  000412                                                    BR      50003$
 (4)  004112                                                  50004$:
1454                                          ;  NEVER RECIEVED THE INTERRUPT
1455  004112                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
 (6)  004112  005262  002676                                            INC     ERRTBL(R2)
1456  004116                                  LET L$LUN := R2 SHIFT -1
 (4)  004116  010237  002074                                            MOV     R2,L$LUN
 (7)  004122  006237  002074                                            ASR     L$LUN
1457  004126                                  ERRHRD 16,TXNOIN
 (4)  004126  104456                                                    TRAP    C$ERHRD
 (5)  004130  000020                                                    .WORD   16
 (5)  004132  003515                                                    .WORD   TXNOIN
 (5)  004134  000000                                                    .WORD   0
1458
1459
1460
1461  004136                  ENDSELECT
 (3)  004136                                                  50003$:
1462
1463  004136                          IF ERRTBL(R2) GT MAXERR THEN
 (6)  004136  026237  002676  002260                                    CMP     ERRTBL(R2),MAXER
 (9)  004144  003402                                                    BLE     50007$
1464  004146  004737  004656                  JSR PC,DROPIT       ; MAXIMUM ERROR COUNT EXCEEDED !
1465  004152                          ENDIF
 (4)  004152                                                  50007$:
1466  004152                          LET STATUS(R2) := STATUS(R2) CLR.BY #ERROR
 (6)  004152  042762  002000  002474                                    BIC     #ERROR,STATUS(R2
1467  004160                          LET ERRCOD := #0
 (4)  004160  005037  002322                                            CLR     ERRCOD
1468  004164  000207                  RTS     PC              ;AND EXIT
```

```
                        .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.
                        ;
004166                  IODRV:  IF #BIT15 NOTSETIN @LPCSR(R2) THEN
(6) 004166 032772 100000 002340                                         BIT     #BIT15,@LPCSR(R2
(9) 004174 001045                                                       BNE     50010$
                        ;
                        ; IF COUNT NOT ZERO SEND NEXT BYTE
                        ;
004176 000240                       NOP ;***************************************************
004200                              IF CURCNT(R2) GT #0 THEN
(6) 004200 005762 002574                                                TST     CURCNT(R2)
(9) 004204 003417                                                       BLE     50011$
004206                                  LET @LPBUF(R2) :B= @CURADD(R2)
(4) 004206 117272 002534 002434                                         MOVB    @CURADD(R2),@LPB
004214                                  LET CURADD(R2) := CURADD(R2) + #1
(6) 004214 005262 002534                                                INC     CURADD(R2)
                        ;
                        ; ENABLE INTERRUPT FOR NEXT BYTE
                        ;
004220                              LET STATUS(R2) := STATUS(R2) SET.BY #ACTIVE
(6) 004220 052762 100000 002474                                         BIS     #ACTIVE,STATUS(R
004226                              LET CURCNT(R2) := CURCNT(R2) - #1
(6) 004226 005362 002574                                                DEC     CURCNT(R2)
004232 000240                       NOP     ;*******************************************
004234                              LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100
(6) 004234 052772 000100 002340                                         BIS     #100,@LPCSR(R2)
004242                          ELSE
(4) 004242 000421                                                       BR      50012$
(3) 004244                                                          50011$:
004244                              IF CURCNT(R2) EQ #0 THEN
(6) 004244 005762 002574                                                TST     CURCNT(R2)
(9) 004250 001010                                                       BNE     50013$
004252                                  LET @LPBUF(R2) :B= #0       ; PAD WITH TRAILING NUL
(4) 004252 105072 002434                                                CLRB    @LPBUF(R2)
004256                                  LET CURCNT(R2) := CURCNT(R2) - #1
(6) 004256 005362 002574                                                DEC     CURCNT(R2)
004262                                  LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100
(6) 004262 052772 000100 002340                                         BIS     #100,@LPCSR(R2)
                                    ; INTERRUPT ONE MORE TIME
004270                              ELSE
(4) 004270 000406                                                       BR      50014$
(3) 004272                                                          50013$:
```

```
1508
1509  004272                                    ; ALL DONE, CLEAR ACTIVE & DISABLE INTERRUPTS
                                                LET STATUS(R2) := STATUS(R2) CLR.BY #ACTIVE
 (6)  004272  042762  100000  002474                                          BIC     #ACTIVE,STATUS(R
1510  004300
                                                LET @LPCSR(R2) := @LPCSR(R2) CLR.BY #100
 (6)  004300  042772  000100  002340                                          BIC     #100,@LPCSR(R2)
1511  004306                                        ENDIF
 (4)  004306                                                          50014$:
1512  004306                                    ENDIF
 (4)  004306                                                          50012$:
1513  004306                                ELSE
 (4)  004306  000406                                                          BR      50015$
 (3)  004310                                                          50010$:
1514
1515                                            ; CLEAR ERROR CONDITION, ENABLE INTERRUPTS
                                                ; SET ERROR FLAG
1516  004310                                    LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
 (6)  004310  052762  002000  002474                                          BIS     #ERROR,STATUS(R2
1517  004316                                    LET @LPCSR(R2) := #100
 (4)  004316  012772  000100  002340                                          MOV     #100,@LPCSR(R2)
1518  004324                                ENDIF
 (4)  004324                                                          50015$:
1519  004324                                POP R2
 (2)  004324  012602                                                          MOV     (SP)+,R2
1520  004326  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
                        ;--
                        IOCTRL: PUSH R2,R3
                          (2)   010246                                          MOV     R2,-(SP)
                          (3)   010346                                          MOV     R3,-(SP)
                        ;
                        ; IF PRINTR IS -1 QUE OUTPUT TO ALL PRINTERS SELECTED
                        ; OTHERWISE TO UNIT NUMBER IN PRINTR.
                        ;
                              IF PRINTR EQ #-1 THEN
                          (6)   023727  002304  177777                          CMP     PRINTR,#-1
                          (9)   001005                                          BNE     50016$
                                  LET R3 := L$UNIT
                          (4)   013703  002012                                  MOV     L$UNIT,R3
                                  LET L$LUN := #0
                          (4)   005037  002074                                  CLR     L$LUN
                              ELSE
                          (4)   000405                                          BR      50017$
                                                                        50016$:
                                  LET R3 := #1
                          (4)   012703  000001                                  MOV     #1,R3
                                  LET L$LUN := PRINTR
                          (4)   013737  002304  002074                          MOV     PRINTR,L$LUN
                              ENDIF
                                                                        50017$:
                        ;
                        ; REPEAT TILL R3 = 0
                        ;
                        CTLLOP:
                              IF R3 EQ #0 THEN
                          (6)   005703                                          TST     R3
                          (9)   001002                                          BNE     50020$
                                  INLINE <JMP CTLEND>
                          (2)   000137  004650                                  JMP     CTLEND
                              ENDIF
                                                                        50020$:
                        ;
                        ; USE R2 AS AN INDEX INTO THE UNIT TABLES
                        ;
                                  LET R2 := L$LUN SHIFT 1
                          (4)   013702  002074                                  MOV     L$LUN,R2
                          (7)   006302                                          ASL     R2
                                  LET ERRCOD := #0
                          (4)   005037  002322                                  CLR     ERRCOD
                        ;
```

```
1560                                    ; IF THE UNIT HAS BEEN DROPPED SELECT THE NEXT UNIT
1561                                    ;
1562  004412                                    IF #DROPED NOTSETIN STATUS(R2) THEN
 (6)  004412  032762  000200  002474                                                    BIT     #DROPED,STATUS(R
 (9)  004420  001103                                                                    BNE     50021$
1563                                    ;
1564                                    ;TEST FOR DVC ERROR BIT SET
1565                                    ;
1566  004422                                        IF #BIT15 SETIN @LPCSR(R2) THEN
 (6)  004422  032772  100000  002340                                                    BIT     #BIT15,@LPCSR(R2
 (9)  004430  001404                                                                    BEQ     50022$
1567  004432                                            LET ERRCOD := #STATER       ; STATUS REG ERROR BIT 15 SET IN CSR
 (4)  004432  012737  000001  002322                                                    MOV     #STATER,ERRCOD
1568  004440                                        ELSE
 (4)  004440  000450                                                                    BR      50023$
 (3)  004442                                                                50022$:
1569                                    ;
1570                                    ; MAKE SURE PREVIOUS MSG IS DONE
1571                                    ;
1572  004442                                        IF CURCNT(R2) GE #0 THEN
 (6)  004442  005762  002574                                                            TST     CURCNT(R2)
 (9)  004446  002445                                                                    BLT     50024$
1573  004450                                            IF #ACTIVE NOTSETIN STATUS(R2) THEN
 (6)  004450  032762  100000  002474                                                    BIT     #ACTIVE,STATUS(R
 (9)  004456  001004                                                                    BNE     50025$
1574                                    ;
1575                                    ; OUTPUT WAS QUEUED BUT I/O DRIVER WAS NEVER INVOKED (VIA INTERRUPT)
1576                                    ;
1577  004460                                            LET ERRCOD := #NOINTR       ; NO INTERRUPT
 (4)  004460  012737  000003  002322                                                    MOV     #NOINTR,ERRCOD
1578  004466                                    ELSE
 (4)  004466  000435                                                                    BR      50026$
 (3)  004470                                                                50025$:
1579  004470                                        WHILE #ACTIVE SETIN STATUS(R2) DO
 (4)  004470                                                                50027$:
 (6)  004470  032762  100000  002474                                                    BIT     #ACTIVE,STATUS(R
 (9)  004476  001431                                                                    BEQ     50030$
1580  004500                                            DELAY 100.
 (2)  004500  012727  000144                                                            MOV     #100.,(PC)+
 (2)  004504  000000                                                                    .WORD   0
 (2)  004506  013727  002116                                                            MOV     L$DLY,(PC)+
 (2)  004512  000000                                                                    .WORD   0
 (2)  004514  005367  177772                                                            DEC     -6(PC)
 (2)  004520  001375                                                                    BNE     .-4
 (2)  004522  005367  177756                                                            DEC     -22(PC)
 (2)  004526  001367                                                                    BNE     .-20
1581                                    ;
1582                                    ; ALLOW APPROX 2.5 SEC FOR BAND STARTUP
1583                                    ;
1584  004530                                            LET STATUS(R2) := STATUS(R2) + #1
 (6)  004530  005262  002474                                                            INC     STATUS(R2)
1585  004534                                            IF #OVFLO SETIN STATUS(R2) THEN
 (6)  004534  032762  000100  002474                                                    BIT     #OVFLO,STATUS(R2
 (9)  004542  001406                                                                    BEQ     50031$
1586  004544                                                LET ERRCOD := #TIMOUT
 (4)  004544  012737  000002  002322                                                    MOV     #TIMOUT,ERRCOD
```

```
1587  004552                                                    LET STATUS(R2) := STATUS(R2) CLR.BY #ACTIVE
 (6)  004552  042762  100000  002474                                                BIC     #ACTIVE,STATUS(R
1588  004560                                                        ENDIF
 (4)  004560                                                                 50031$:
1589  004560                                                    ENDDO
 (4)  004560  000743                                                         BR      50027$
 (3)  004562                                                                 50030$:
1590  004562                                                ENDIF
 (4)  004562                                                                 50026$:
1591  004562                                            ENDIF                 50024$:
 (4)  004562
1592  004562                                        ENDIF                     50023$:
 (4)  004562
1593  004562                                        IF ERRCOD NE #0 THEN
 (6)  004562  005737  002322                                                 TST     ERRCOD
 (9)  004566  001403                                                         BEQ     50032$
1594
1595                                ;
                                    ; REPORT THE ERROR
1596                                ;
1597  004570  004777  176100                                JSR     PC,@ERRSVC
1598  004574                                            ELSE
 (4)  004574  000415                                                         BR      50033$
 (3)  004576                                                                 50032$:
1599                                ;
1600                                ; Q UP THE MESSAGE AND ENABLE INTERRUPTS
1601                                ; THE I/O DRIVER WILL PICK UP FROM HERE.
1602                                ;
1603  004576                                                LET CURADD(R2) := BUFADD
 (4)  004576  013762  002334  002534                                        MOV     BUFADD,CURADD(R2
1604  004604                                                LET CURCNT(R2) := BUFCNT
 (4)  004604  013762  002336  002574                                        MOV     BUFCNT,CURCNT(R2
1605  004612                                                LET STATUS(R2) := STATUS(R2) CLR.BY #177
 (6)  004612  042762  000177  002474                                        BIC     #177,STATUS(R2)
1606  004620  000240                                       NOP     ;***************************************
1607  004622                                                LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100
 (6)  004622  052772  000100  002340                                        BIS     #100,@LPCSR(R2)
1608  004630                                            ENDIF
 (4)  004630                                                                 50033$:
1609  004630                                        ENDIF
 (4)  004630                                                                 50021$:
1610                                ;
1611                                ; CLEAR OUT ANY TIMEOUT COUNT
1612                                ;
1613  004630                                        LET STATUS(R2) := STATUS(R2) CLR.BY #177
 (6)  004630  042762  000177  002474                                        BIC     #177,STATUS(R2)
1614                                ;
1615                                ; SELECT THE NEXT UNIT AND DECRIMENT THE LINECOUNT
1616                                ;
1617  004636                                        LET R3 := R3 - #1
 (6)  004636  005303                                                        DEC     R3
1618  004640                                        LET L$LUN := L$LUN + #1
 (6)  004640  005237  002074                                                INC     L$LUN
1619  004644  000137  004370                    JMP     CTLLOP
1620  004650                                CTLEND:
1621  004650                                    POP  R3,R2
 (2)  004650  012603                                                        MOV     (SP)+,R3
```

```
   (3)  004652  012602                                          MOV     (SP)+,R2
1622    004654  000207                          RTS     PC
```

```
1624                            ;=============================================================
1625                            ; DROPIT        FUNCTIONAL DESCRIPTION :
1626                            ;
1627                            ; THIS SUBROUTINE IS USED TO DROP A BAD PRINTER FROM THE TEST
1628                            ; DISABLE ANY INTERRUPTS FROM THE PRINTER, AND NOTIFY THE
1629                            ; OPERATOR THAT THE PRINTER WAS DROPPED.
1630                            ;=============================================================
1631
1632  004656                    DROPIT: LET STATUS(R2) := #DROPED
 (4)  004656  012762  000200  002474                                    MOV     #DROPED,STATUS(R
1633  004664                          LET CURCNT(R2) := #-1
 (4)  004664  012762  177777  002574                                    MOV     #-1,CURCNT(R2)
1634  004672                          LET @LPCSR(R2) := #0
 (4)  004672  005072  002340                                            CLR     @LPCSR(R2)
1635  004676                          PRINTF #LPDROP, L$LUN
 (8)  004676  013746  002074                                            MOV     L$LUN,-(SP)
 (7)  004702  012746  003732                                            MOV     #LPDROP,-(SP)
 (6)  004706  012746  000002                                            MOV     #2,-(SP)
 (3)  004712  010600                                                    MOV     SP,R0
 (4)  004714  104417                                                    TRAP    C$PNTF
 (4)  004716  062706  000006                                            ADD     #6,SP
1636  004722                          LET ERRTBL(R2) := #0
 (4)  004722  005062  002676                                            CLR     ERRTBL(R2)
1637  004726                          LET UUT := UUT - #1
 (6)  004726  005337  002326                                            DEC     UUT
1638  004732                          IF UUT EQ #0 THEN
 (6)  004732  005737  002326                                            TST     UUT
 (9)  004736  001011                                                    BNE     50034$
1639  004740                          PRINTF #UUTEQ0
 (7)  004740  012746  003546                                            MOV     #UUTEQ0,-(SP)
 (6)  004744  012746  000001                                            MOV     #1,-(SP)
 (3)  004750  010600                                                    MOV     SP,R0
 (4)  004752  104417                                                    TRAP    C$PNTF
 (4)  004754  062706  000004                                            ADD     #4,SP
1640  004760                              DOCLN       ; NOTHING TO TEST
 (3)  004760  104444                                                    TRAP    C$DCLN
1641  004762                          ENDIF
 (4)  004762                                                    50034$:
1642  004762  000207                   RTS     PC
1643
1644
1645
1646                            ;=============================================================
1647                            ; FAKE          FUNCTIONAL DESCRIPTION:
1648                            ;
1649                            ; THIS SUBROUTINE IS REQUIRED TO INSURE PROPER PASS COUNT REPORTS
1650                            ; IN A MULTI UNIT MODE OF OPERATION.
1651                            ;=============================================================
1652
1653  004764                    FAKE:   LET L$LUN := #0
 (4)  004764  005037  002074                                            CLR     L$LUN
1654  004770                          WHILE L$LUN LT L$UNIT DO
 (4)  004770                                                    50035$:
 (6)  004770  023737  002074  002012                                    CMP     L$LUN,L$UNIT
 (9)  004776  002007                                                    BGE     50036$
1655  005000                              GPHARD      L$LUN, R3
```

```
  (3)  005000  013700  002074                                                    MOV     L$LUN,R0
  (3)  005004  104442                                                            TRAP    C$GPHRD
  (3)  005006  010003                                                            MOV     R0,R3
1656   005010                                LET L$LUN := L$LUN + #1
  (6)  005010  005237  002074                                                    INC     L$LUN
1657   005014                            ENDDO
  (4)  005014  000765                                                            BR      50035$
  (3)  005016                                                           50036$:
1658   005016  000207                    RTS     PC
1659
1660
1661   005020                            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.
                        ;--
005020                  BGNMOD
005020                  BGNINIT
 (3) 005020                                                                  L$INIT::
                        ;RESET EXTERNAL BUS IF START EVENT FLAG IS SET
                        ;OR POWER FAIL RESTART
005020                          READEF  #EF.START               ;TEST START EF INDICATOR
 (3) 005020  012700  000040                                                 MOV     #EF.START,R0
 (3) 005024  104447                                                         TRAP    C$REFG
005026                          BCOMPLETE 1$                    ;BRANCH IF FROM START UP
 (2) 005026  103410                                                         BCS     1$
005030                          READEF  #EF.RESTART             ;NOW THE RESTARTFLAG
 (3) 005030  012700  000037                                                 MOV     #EF.RESTART,R0
 (3) 005034  104447                                                         TRAP    C$REFG
005036                          BCOMPLETE 1$                    ;IF EITHER START OR POWER FAIL RESTART
 (2) 005036  103404                                                         BCS     1$
                                                                ;DO A BUS RESET
005040  004737  004764          JSR     PC,FAKE                 ; UPDATE PASS COUNT
005044                          EXIT INIT                       ; ELSE EXIT INIT CODE
 (3) 005044  104432                                                         TRAP    C$EXIT
 (3) 005046  001134                                                         .WORD   L10004-.
                        ;
                        ;POWER UP RESTART OR START COMMAND ISSUED
                        ;
005050                  1$:     BRESET                          ;RESET THE BUS
 (3) 005050  104433                                                         TRAP    C$RESET
005052                          IF L$UNIT GT #16. THEN
 (6) 005052  023727  002012  000020                                         CMP     L$UNIT,#16.
 (9) 005060  003420                                                         BLE     50037$
005062                              PRINTF #NRGT16
 (7) 005062  012746  005612                                                 MOV     #NRGT16,-(SP)
 (6) 005066  012746  000001                                                 MOV     #1,-(SP)
 (3) 005072  010600                                                         MOV     SP,R0
 (4) 005074  104417                                                         TRAP    C$PNTF
 (4) 005076  062706  000004                                                 ADD     #4,SP
005102                              PRINTF #NRGT17
 (7) 005102  012746  005675                                                 MOV     #NRGT17,-(SP)
 (6) 005106  012746  000001                                                 MOV     #1,-(SP)
 (3) 005112  010600                                                         MOV     SP,R0
 (4) 005114  104417                                                         TRAP    C$PNTF
 (4) 005116  062706  000004                                                 ADD     #4,SP
005122                          ENDIF
 (4) 005122                                                     50037$:
005122                          MANUAL          ; WAIT FOR CR IF IN MANUAL MODE
 (3) 005122  104450                                                         TRAP    C$MANI
005124                          BNCOMPLETE      100$
 (2) 005124  103016                                                         BCC     100$
005126                          PRINTF  #MRESET                 ;PRINT RESET MESSAGE
```

```
   (7)  005126  012746  005731                                                                    MOV     #MRESET,-(SP)
   (6)  005132  012746  000001                                                                    MOV     #1,-(SP)
   (3)  005136  010600                                                                            MOV     SP,R0
   (4)  005140  104417                                                                            TRAP    C$PNTF
   (4)  005142  062706  000004                                                                    ADD     #4,SP
1694                                ;
1695                                ;WAIT FOR A ''CR'' BEFORE GOING ON
1696                                ;
1697  005146                                LET FLAG := #0
   (4)  005146  005037  002262                                                                    CLR     FLAG
1698  005152                                LET ERRCOD := #0
   (4)  005152  005037  002322                                                                    CLR     ERRCOD
1699  005156                                LET UUT := #0
   (4)  005156  005037  002326                                                                    CLR     UUT
1700  005162                          100$:
1701  005162                                GMANIL  READY,FLAG,100000,YES   ;GET MANUAL PARAMETERS
   (3)  005162  104443                                                                            TRAP    C$GMAN
   (3)  005164  000404                                                                            BR      10000$
   (4)  005166  002262                                                                            .WORD   FLAG
   (5)  005170  000130                                                                            .WORD   T$CODE
   (5)  005172  006010                                                                            .WORD   READY
   (5)  005174  100000                                                                            .WORD   100000
   (3)  005176                                                                          10000$:
1702                                ;
1703                                ;REQUEST P-TABLE FOR PRINTERS UNDER TEST
1704                                ;
1705  005176                          2$:   LET R1 := L$UNIT - #1           ;MAXIMUM NUMBER OF UNITS
   (4)  005176  013701  002012                                                                    MOV     L$UNIT,R1
   (6)  005202  005301                                                                            DEC     R1
1706  005204                                INCR L$LUN FROM #0 TO R1 BY #1
   (4)  005204  005037  002074                                                                    CLR     L$LUN
   (5)  005210  000402                                                                            BR      50040$
   (4)  005212                                                                          50041$:
   (7)  005212  005237  002074                                                                    INC     L$LUN
   (5)  005216                                                                          50040$:
   (5)  005216  023701  002074                                                                    CMP     L$LUN,R1
   (7)  005222  003071                                                                            BGT     50042$
1707  005224                                GPHARD L$LUN,R3             ;REQUEST P-TABLE ADDRESS
   (3)  005224  013700  002074                                                                    MOV     L$LUN,R0
   (3)  005230  104442                                                                            TRAP    C$GPHRD
   (3)  005232  010003                                                                            MOV     R0,R3
1708  005234                                BNCOMPLETE 3$               ;BRANCH IF DEVICE NOT PRESENT
   (2)  005234  103060                                                                            BCC     3$
1709  005236                                LET R2 := L$LUN SHIFT 1
   (4)  005236  013702  002074                                                                    MOV     L$LUN,R2
   (7)  005242  006302                                                                            ASL     R2
1710                                ;
1711                                ;CLEAR STATUS WORD IN TABLE, CLEAR ANY ERROR COUNT, OUTPUT COUNT
1712                                ;
1713  005244                                LET STATUS(R2) := #0
   (4)  005244  005062  002474                                                                    CLR     STATUS(R2)
1714  005250                                LET ERRTBL(R2) := #0
   (4)  005250  005062  002676                                                                    CLR     ERRTBL(R2)
1715  005254                                LET CURCNT(R2) := #-1
   (4)  005254  012762  177777  002574                                                            MOV     #-1,CURCNT(R2)
1716                                ;
```

```
1717                                    ;LOAD CSR ADDRESS INTO TABLE
1718                                    ;
1719  005262                                    LET LPCSR(R2) := (R3)+      ;SET UP CSR ADDRESS FOR DEVICE
 (4)  005262  012362  002340                                                MOV     (R3)+,LPCSR(R2)
1720  005266                                    LET LPBUF(R2) := LPCSR(R2) + #2
 (4)  005266  016262  002340  002434                                        MOV     LPCSR(R2),LPBUF(
 (6)  005274  062762  000002  002434                                        ADD     #2,LPBUF(R2).
1721                                    ;
1722                                    ;SET UP VECTOR ADDRESS INTO GIVEN TABLE
1723                                    ;
1724  005302                                    LET LPVEC(R2) := (R3)+
 (4)  005302  012362  002400                                                MOV     (R3)+,LPVEC(R2)
1725                                    ;
1726                                    ;SET UP DEVICE INTERRUPT VECTOR INFORMATION
1727                                    ;
1728  005306                                LET WORK := R2 SHIFT 3
 (4)  005306  010237  002736                                                MOV     R2,WORK
 (7)  005312  006337  002736                                                ASL     WORK
 (7)  005316  006337  002736                                                ASL     WORK
 (7)  005322  006337  002736                                                ASL     WORK
1729  005326                                LET WORK := WORK + #INT00
 (6)  005326  062737  031616  002736                                        ADD     #INT00,WORK
1730  005334                                LET LPINTR(R2) := WORK
 (4)  005334  013762  002736  002634                                        MOV     WORK,LPINTR(R2)
1731  005342                                SETVEC LPVEC(R2), LPINTR(R2), #PRI04
 (7)  005342  012746  000200                                                MOV     #PRI04,-(SP)
 (6)  005346  016246  002634                                                MOV     LPINTR(R2),-(SP)
 (5)  005352  016246  002400                                                MOV     LPVEC(R2),-(SP)
 (4)  005356  012746  000003                                                MOV     #3,-(SP)
 (3)  005362  104437                                                        TRAP    C$SVEC
 (2)  005364  062706  000010                                                ADD     #10,SP
1732                                    ;
1733                                    ; ADD ONE TO UNIT UNDER TEST COUNT
1734                                    ;
1735  005370                                LET UUT := UUT + #1
 (6)  005370  005237  002326                                                INC     UUT
1736  005374  000403                        BR  4$
1737                                    ;
1738                                    ;INDICATE L$LUN NOT AVAILABLE FOR TESTING
1739                                    ;
1740  005376                            3$:     LET STATUS(R2) := STATUS(R2) SET.BY #DROPED
 (6)  005376  052762  000200  002474                                        BIS     #DROPED,STATUS(R
1741  005404                            4$:   ENDINC                        ;GO BACK AND DO IT AGAIN
 (4)  005404  000702                                                        BR      50041$
 (3)  005406                                                            50042$:
1742                                    ;DETERMINE IF MANUAL PRINT SPEED TO BE DONE AND PROCESS ACCORDINGLY
1743                                    ;
1744                                    ;
1745                                    ;THIS IF STATEMENT HAS BEEN HARD CODED BECAUSE OF
1746                                    ;NON-LOCAL SYMBOLS IN THE LOOP
1747                                    ;
1748                                    ;IF MANSPD EQ #0 THEN               ; IF 0 AUTO MATIC PRINT SPEED TEST
1749                                    ;
1750  005406  005737  002252                    TST     MANSPD
1751  005412  001405                            BEQ     CK1                 ;BRANCH IF AUTOMATIC TEST TO BE RUN
1752  005414                                    LET CLKTYP := #4
```

```
  (4)  005414  012737  000004  002306                                                    MOV     #4,CLKTYP
1753   005422                          EXIT INIT
  (3)  005422  104432                                                                    TRAP    C$EXIT
  (3)  005424  000556                                                                    .WORD   L10004-.
1754                                ;
1755
1756                                ;END OF INITIAL SET UP SECTION. NOW FIND OUT WHAT TYPE OF CLOCK IS
1757                                ;AVAILABLE IF ANY.
1758                                ;
1759   005426                       CK1:    ; SEE IF THERE IS A 'P' CLOCK
1760   005426                               CLOCK P,R4
  (3)  005426  012700  000120                                                           MOV     #'P,R0
  (3)  005432  104462                                                                   TRAP    C$CLCK
  (3)  005434  010004                                                                   MOV     R0,R4
1761                                ;IF CARRY CLEAR 'P' CLOCK DOESN'T EXIST
1762   005436                       IFCOND CC THEN
  (6)  005436  103435                                                                   BCS     50043$
1763                                    ; SO TEST FOR AN 'L' CLOCK
1764   005440                           CLOCK L,R4
  (3)  005440  012700  000114                                                           MOV     #'L,R0
  (3)  005444  104462                                                                   TRAP    C$CLCK
  (3)  005446  010004                                                                   MOV     R0,R4
1765                                    ; IF CARRY CLEAR THEN 'L' CLOCK DOESN'T EXIST
1766   005450                           IFCOND CC THEN
  (6)  005450  103424                                                                   BCS     50044$
1767   005452                               LET CLKTYP := #1
  (4)  005452  012737  000001  002306                                                   MOV     #1,CLKTYP
1768   005460                               PRINTF #NOCLCK
  (7)  005460  012746  006045                                                           MOV     #NOCLCK,-(SP)
  (6)  005464  012746  000001                                                           MOV     #1,-(SP)
  (3)  005470  010600                                                                   MOV     SP,R0
  (4)  005472  104417                                                                   TRAP    C$PNTF
  (4)  005474  062706  000004                                                           ADD     #4,SP
1769   005500                               PRINTF #NOTIM
  (7)  005500  012746  006107                                                           MOV     #NOTIM,-(SP)
  (6)  005504  012746  000001                                                           MOV     #1,-(SP)
  (3)  005510  010600                                                                   MOV     SP,R0
  (4)  005512  104417                                                                   TRAP    C$PNTF
  (4)  005514  062706  000004                                                           ADD     #4,SP
1770   005520                           ELSE
  (4)  005520  000403                                                                   BR      50045$
  (3)  005522                                                                   50044$:
1771                                        ;WE HAVE AN 'L' CLOCK
1772   005522                               LET CLKTYP := #2
  (4)  005522  012737  000002  002306                                                   MOV     #2,CLKTYP
1773   005530                           ENDIF
  (4)  005530                                                                   50045$:
1774   005530                       ELSE
  (4)  005530  000403                                                                   BR      50046$
  (3)  005532                                                                   50043$:
1775                                        ;WE HAVE A 'P' CLOCK
1776   005532                               LET CLKTYP := #3
  (4)  005532  012737  000003  002306                                                   MOV     #3,CLKTYP
1777   005540                       ENDIF
  (4)  005540                                                                   50046$:
1778   005540                       IF CLKTYP EQ #2 OR CLKTYP EQ #3 THEN
```

```
  (6)  005540  023727  002306  000002                                                    CMP     CLKTYP,#2
  (8)  005546  001404                                                                    BEQ     50047$
  (6)  005550  023727  002306  000003                                                    CMP     CLKTYP,#3
  (9)  005556  001010                                                                    BNE     50050$
  (6)  005560                                                                         50047$:
1779   005560                          LET CLOCKP := R4
  (4)  005560  010437  002310                                                         MOV     R4,CLOCKP
1780   005564                          LET CLKCSR := @CLOCKP
  (4)  005564  017737  174520  002312                                                MOV     @CLOCKP,CLKCSR
1781   005572  012777  000000  174512          MOV     #0,@CLKCSR
1782   005600                          ENDIF
  (4)  005600                                                                         50050$:
1783   005600                          SETPRI #PRI00
  (3)  005600  012700  000000                                                         MOV     #PRI00,R0
  (3)  005604  104441                                                                 TRAP    C$SPRI
1784   005606                  EXIT INIT
  (3)  005606  104432                                                                 TRAP    C$EXIT
  (3)  005610  000372                                                                 .WORD   L10004-.
1785                            .NLIST BEX
1786
1787   005612  047045  040445  052516  NRGT16: .ASCIZ  /%N%ANUMBER OF LINE PRINTERS UNDER TEST EXCEEDS 16./
1788   005675     045  022516  047501  NRGT17: .ASCIZ  /%N%AONLY 16 WILL BE TESTED./
1789   005731     045  022516  051101  MRESET: .ASCIZ  /%N%ARESET LINE PRINTER(S) AND PLACE ON LINE.%N/
1790
1791   006010  042504  051120  051505  READY:  .ASCIZ  /DEPRESS 'RETURN'' WHEN READY./
1792   006045     045  022516  044101  NOCLCK: .ASCIZ  /%N%AHARDWARE CLOCK NOT AVAILABLE./
1793   006107     045  022516  040501  NOTIM:  .ASCIZ  /%N%AAUTO PRINTING SPEED MEASUREMENT CANNOT BE PERFORMED./
1794                            .EVEN
1795   006200  000000          PLOC:   .WORD   0
1796
1797                            .LIST BEX
1798   006202                   ENDINIT
  (3)  006202                                                                         L10004:
  (3)  006202  104411                                                                 TRAP    C$INIT
1799
1800
1801                            ;-------------------------------------------------------------
1802                            ;
1803                            ; RESVEC        FUNCTIONAL DESCRIPTION
1804                            ;
1805                            ;       THIS SUBROUTINE WILL SETUP ALL UNITS VECTOR AREAS
1806                            ;       TO THE 'NORMAL' INTERRUPT ROUTINES STARTING AT INT00.
1807                            ;-------------------------------------------------------------
1808
1809   006204                   RESVEC::        PUSH    R3,R4
  (2)  006204  010346                                                                 MOV     R3,-(SP)
  (3)  006206  010446                                                                 MOV     R4,-(SP)
1810   006210                          LET  R4 := #0
  (4)  006210  005004                                                                 CLR     R4
1811   006212                          LET R3 := L$UNIT
  (4)  006212  013703  002012                                                         MOV     L$UNIT,R3
1812   006216                          WHILE R3 GT #0 DO
  (4)  006216                                                                         50051$:
  (6)  006216  005703                                                                 TST     R3
  (9)  006220  003417                                                                 BLE     50052$
1813   006222                              SETVEC LPVEC(R4), LPINTR(R4), #PRI04
```

```
  (7)  006222  012746  000200                                        MOV     #PRI04,-(SP)
  (6)  006226  016446  002634                                        MOV     LPINTR(R4),-(SP)
  (5)  006232  016446  002400                                        MOV     LPVEC(R4),-(SP)
  (4)  006236  012746  000003                                        MOV     #3,-(SP)
  (3)  006242  104437                                                TRAP    C$SVEC
  (2)  006244  062706  000010                                        ADD     #10,SP
1814   006250                        LET R4 := R4 + #2
  (6)  006250  062704  000002                                        ADD     #2,R4
1815   006254                        LET R3 := R3 - #1
  (6)  006254  005303                                                DEC     R3
1816   006256                      -ENDDO
  (4)  006256  000757                                                BR      50051$
  (3)  006260                                                50052$:
1817   006260                        POP R4,R3
  (2)  006260  012604                                                MOV     (SP)+,R4
  (3)  006262  012603                                                MOV     (SP)+,R3
1818   006264  000207                RTS     PC
1819
```

```
1821                                    .SBTTL  CLEANUP CODING SECTION
1822  006266                            STARS
 (2)                                    ;;*****************************************************************
1823                                    ;++
1824                                    ;THE PURPOSE OF THE CLEANUP SECTION IS TO CLEANUP ALL PRINTERS UNDER TEST
1825                                    ;AND RETEST ANY UNITS WHICH HAVE BEEN DROPPED FROM  TESTING TO INSURE THAT
1826                                    ;THEY HAVE NOT COME BACK ON LINE. IF THE DEVICE HAS COME BACK ON LINE
1827                                    ;TESTING WILL BE RESTARTED ON THE DEVICE. THIS INSURES THAT
1828                                    ;IN THE EVENT A PAPER OUT OCCURRED AND THE OPERATOR HAS PUT ADDITIONAL PAPER
1829                                    ;INTO THE UNIT UNDER TEST, THE INITIALIZATION SEQUENCE DOES NOT
1830                                    ;HAVE TO BE DONE AGAIN IN ORDER TO GET THE DEVICE ACTIVE.
1831                                    ;--
1832  006266                            STARS
 (2)                                    ;;*****************************************************************
1833  006266                            BGNCLN
 (3)  006266                                                                    L$CLEAN::
1834  006266                                    BRESET
 (3)  006266  104433                                                                TRAP    C$RESET
1835
1836  006270                            CLEAN:  LET R1 := L$UNIT - #1   ;NUMBER OF UNITS-1
 (4)  006270  013701  002012                                                        MOV     L$UNIT,R1
 (6)  006274  005301                                                                DEC     R1
1837  006276                                    INCR L$LUN FROM #0 TO R1 BY #1
 (4)  006276  005037  002074                                                        CLR     L$LUN
 (5)  006302  000402                                                                BR      50053$
 (4)  006304                                                                50054$:
 (7)  006304  005237  002074                                                        INC     L$LUN
 (5)  006310                                                                50053$:
 (5)  006310  023701  002074                                                        CMP     L$LUN,R1
 (7)  006314  003013                                                                BGT     50055$
1838                                    ; DISABLE ALL INTERRUPTS, SELECT ALL LINES
1839                                    ; ZERO ALL ERROR COUNTS
1840  006316                                LET R2 := L$LUN SHIFT 1
 (4)  006316  013702  002074                                                        MOV     L$LUN,R2
 (7)  006322  006302                                                                ASL     R2
1841  006324                                LET STATUS(R2) := #0
 (4)  006324  005062  002474                                                        CLR     STATUS(R2)
1842  006330                                LET CURCNT(R2) := #-1
 (4)  006330  012762  177777  002574                                                MOV     #-1,CURCNT(R2)
1843  006336                                LET ERRTBL(R2) := #0
 (4)  006336  005062  002676                                                        CLR     ERRTBL(R2)
1844  006342                            ENDINC
 (4)  006342  000760                                                                BR      50054$
 (3)  006344                                                                50055$:
1845  006344  004737  006204           JSR     PC,RESVEC               ; RESET THE VECTORS
1846  006350                            IF CLKTYP EQ #2 OR CLKTYP EQ #3 THEN
 (6)  006350  023727  002306  000002                                                CMP     CLKTYP,#2
 (8)  006356  001404                                                                BEQ     50056$
 (6)  006360  023727  002306  000003                                                CMP     CLKTYP,#3
 (9)  006366  001006                                                                BNE     50057$
 (6)  006370                                                                50056$:
1847  006370                                CLRVEC @CLKVEC
 (3)  006370  017700  173722                                                        MOV     @CLKVEC,R0
 (3)  006374  104436                                                                TRAP    C$CVEC
1848  006376  012777  000000  173706      MOV     #0,@CLKCSR
1849  006404                            ENDIF
```

```
  (4)  006404                                                    50057$:
1850   006404                        ENDCLN
  (3)  006404                                                    L10005:
  (3)  006404  104412                                                   TRAP    C$CLEAN
1851
1852   006406                        ENDMOD
```

```
                                .SBTTL  INTERFACE LOGIC

006406                          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.
                                ;
                                ;
                                ;--
006406                          BGNTST  1
(3)  006406                                                             T1::
006406                          LET R1 := L$UNIT - #1                   ;MAX NUMBER OF UNITS ON SYSTEM
(4)  006406  013701  002012                                                     MOV     L$UNIT,R1
(6)  006412  005301                                                             DEC     R1
                                ;
                                ;HARD CODED INCREMEMNT LOOP
                                ;INCR LUNIT FROM #0 TO R1 BY #1         ;START LOOP
                                ;
006414  005037  002300                  CLR     LUNIT                   ;UNIT TO 0
006420  000402                         BR      T1C                     ;DO COMPARE
006422                          T1A:
006422  005237  002300                  INC     LUNIT                   ;UPDATE UNIT NUMBER
006426                          T1C:
006426  023701  002300                  CMP     LUNIT,R1                ;DO COMPARISON OF UNIT NUMBER
006432  003402                         BLE     1$                      ;ONTO NEXT UNIT
006434  000137  007116                 JMP     T1B                     ;EXIT LOOP
006440                          1$:
006440                                  LET R2 := LUNIT SHIFT 1
(4)  006440  013702  002300                                                     MOV     LUNIT,R2
(7)  006444  006302                                                             ASL     R2
006446                              IF #BIT15 SETIN @LPCSR(R2) THEN
(6)  006446  032772  100000  002340                                             BIT     #BIT15,@LPCSR(R2
(9)  006454  001416                                                             BEQ     50060$
006456                                  LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
(6)  006456  052762  002000  002474                                             BIS     #ERROR,STATUS(R2
006464                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
(6)  006464  005262  002676                                                     INC     ERRTBL(R2)
006470                                  LET L$LUN := LUNIT
(4)  006470  013737  002300  002074                                             MOV     LUNIT,L$LUN
006476                                  ERRHRD 1,CSRERR         ;ERROR BIT WAS SET. SAY SO
(4)  006476  104456                                                             TRAP    C$ERHRD
(5)  006500  000001                                                             .WORD   1
(5)  006502  003170                                                             .WORD   CSRERR
(5)  006504  000000                                                             .WORD   0
006506                                  LET @LPCSR(R2) := #0
(4)  006506  005072  002340                                                     CLR     @LPCSR(R2)
006512                              ENDIF
(4)  006512                                                             50060$:
                                ;TIME DELAY
                                        ; IF NOT READY ALLOW 3 SECONDS TO COME UP
006512                                  IF #BIT7 NOTSETIN @LPCSR(R2) THEN
(6)  006512  032772  000200  002340                                             BIT     #BIT7,@LPCSR(R2)
```

```
  (9)  006520  001014                                                                    BNE     50061$
1893   006522                        DELAY 30.
  (2)  006522  012727  000036                                                          MOV     #30.,(PC)+
  (2)  006526  000000                                                                  .WORD   0
  (2)  006530  013727  002116                                                          MOV     L$DLY,(PC)+
  (2)  006534  000000                                                                  .WORD   0
  (2)  006536  005367  177772                                                          DEC     -6(PC)
  (2)  006542  001375                                                                  BNE     .-4
  (2)  006544  005367  177756                                                          DEC     -22(PC)
  (2)  006550  001367                                                                  BNE     .-20
1894   006552                    ENDIF
  (4)  006552                                                                  50061$:
1895                            ;
1896                            ;NOW TEST FOR PRINTER READY
1897                            ;
1898   006552                       IF #BIT07 NOTSETIN @LPCSR(R2) THEN          ;TEST FOR THE READY BIT
  (6)  006552  032772  000200  002340                                                  BIT     #BIT07,@LPCSR(R2
  (9)  006560  001014                                                                  BNE     50062$
1899   006562                           LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
  (6)  006562  052762  002000  002474                                                  BIS     #ERROR,STATUS(R2
1900   006570                           LET L$LUN := LUNIT
  (4)  006570  013737  002300  002074                                                  MOV     LUNIT,L$LUN
1901   006576                           LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  006576  005262  002676                                                          INC     ERRTBL(R2)
1902   006602                           ERRHRD 2,RDYERR             ;REPORT AN ERROR
  (4)  006602  104456                                                                  TRAP    C$ERHRD
  (5)  006604  000002                                                                  .WORD   2
  (5)  006606  003206                                                                  .WORD   RDYERR
  (5)  006610  000000                                                                  .WORD   0
1903   006612                       ENDIF
  (4)  006612                                                                  50062$:
1904                            ;
1905                            ;INSURE LOADING CHARACTER CAUSES PRINTER READY TO GO AWAY
1906                            ;
1907   006612                       LET @LPBUF(R2) := #12
  (4)  006612  012772  000012  002434                                                  MOV     #12,@LPBUF(R2)
1908   006620                       IF #BIT07 SETIN @LPCSR(R2) THEN
  (6)  006620  032772  000200  002340                                                  BIT     #BIT07,@LPCSR(R2
  (9)  006626  001416                                                                  BEQ     50063$
1909   006630                           LET STATUS(R2) := STATUS(R2) SET.BY #ERROR
  (6)  006630  052762  002000  002474                                                  BIS     #ERROR,STATUS(R2
1910   006636                           LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  006636  005262  002676                                                          INC     ERRTBL(R2)
1911   006642                           LET L$LUN := LUNIT
  (4)  006642  013737  002300  002074                                                  MOV     LUNIT,L$LUN
1912   006650                           ERRHRD 3,ERR11              ;REPORT AN ERROR
  (4)  006650  104456                                                                  TRAP    C$ERHRD
  (5)  006652  000003                                                                  .WORD   3
  (5)  006654  007172                                                                  .WORD   ERR11
  (5)  006656  000000                                                                  .WORD   0
1913   006660                           LET @LPCSR(R2) := #0
  (4)  006660  005072  002340                                                          CLR     @LPCSR(R2)
1914   006664                       ENDIF
  (4)  006664                                                                  50063$:
1915                            ;
1916                            ;VERIFY THAT THE PRINTER WILL NOT INTERRUPT IF IT IS AT A PRIORITY LEVEL
```

```
                                ;THE SAME AS THE CPU
                                ;
                                    SETPRI #PRI04                       ;CPU TO PRIORITY 4
(3)  006664  012700  000200                                                     MOV     #PRI04,R0
(3)  006670  104441                                                             TRAP    C$SPRI
                                    SETVEC LPVEC(R2),#INTERR,#PRI04     ;LP VECTOR SET UP
(7)  006672  012746  000200                                                     MOV     #PRI04,-(SP)
(6)  006676  012746  007134                                                     MOV     #INTERR,-(SP)
(5)  006702  016246  002400                                                     MOV     LPVEC(R2),-(SP)
(4)  006706  012746  000003                                                     MOV     #3,-(SP)
(3)  006712  104437                                                             TRAP    C$SVEC
(2)  006714  062706  000010                                                     ADD     #10,SP
                                    LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100    ;INTERRUPT ENABLE
(6)  006720  052772  000100  002340                                             BIS     #100,@LPCSR(R2)
                                        DELAY   30                  ; ALLOW 3 SEC FOR DELAY
(2)  006726  012727  000030                                                     MOV     #30,(PC)+
(2)  006732  000000                                                             .WORD   0
(2)  006734  013727  002116                                                     MOV     L$DLY,(PC)+
(2)  006740  000000                                                             .WORD   0
(2)  006742  005367  177772                                                     DEC     -6(PC)
(2)  006746  001375                                                             BNE     .-4
(2)  006750  005367  177756                                                     DEC     -22(PC)
(2)  006754  001367                                                             BNE     .-20
                                ;
                                ;NOW TEST THAT THE PRINTER WILL INTERRUPT IF THE PRIORITY IS LOWER THAN
                                ;THE CPU
                                ;
                                    LET @LPCSR(R2) := @LPCSR(R2) CLR.BY #100    ;CLEAR INTERRUPT ENABLE
(6)  006756  042772  000100  002340                                             BIC     #100,@LPCSR(R2)
                                    SETPRI #PRI03                       ;CPU TO PRIORITY 3
(3)  006764  012700  000140                                                     MOV     #PRI03,R0
(3)  006770  104441                                                             TRAP    C$SPRI
                                    SETVEC LPVEC(R2),#INTHDL,#PRI04
(7)  006772  012746  000200                                                     MOV     #PRI04,-(SP)
(6)  006776  012746  007164                                                     MOV     #INTHDL,-(SP)
(5)  007002  016246  002400                                                     MOV     LPVEC(R2),-(SP)
(4)  007006  012746  000003                                                     MOV     #3,-(SP)
(3)  007012  104437                                                             TRAP    C$SVEC
(2)  007014  062706  000010                                                     ADD     #10,SP
                                LET @LPCSR(R2) := @LPCSR(R2) SET.BY #100    ;INTERRUPT ENABLE
(6)  007020  052772  000100  002340                                             BIS     #100,@LPCSR(R2)
                                    DELAY 30                    ; ALLOW 3 SEC DELAY
(2)  007026  012727  000030                                                     MOV     #30,(PC)+
(2)  007032  000000                                                             .WORD   0
(2)  007034  013727  002116                                                     MOV     L$DLY,(PC)+
(2)  007040  000000                                                             .WORD   0
(2)  007042  005367  177772                                                     DEC     -6(PC)
(2)  007046  001375                                                             BNE     .-4
(2)  007050  005367  177756                                                     DEC     -22(PC)
(2)  007054  001367                                                             BNE     .-20
                                    LET ERRTBL(R2) := ERRTBL(R2) + #1
(6)  007056  005262  002676                                                     INC     ERRTBL(R2)
                                    LET L$LUN := LUNIT
(4)  007062  013737  002300  002074                                             MOV     LUNIT,L$LUN
                                ERRHRD 4,ERR13
(4)  007070  104456                                                             TRAP    C$ERHRD
```

```
  (5)  007072  000004                                                                    .WORD   4
  (5)  007074  007331                                                                    .WORD   ERR13
  (5)  007076  000000                                                                    .WORD   0
1935   007100                        END2:   LET @LPCSR(R2) := @LPCSR(R2) CLR.BY #100          ;MAKE SURE INTERRUPT ENABLE IS C
  (6)  007100  042772  000100  002340                                                    BIC     #100,@LPCSR(R2)
1936   007106                                LET STATUS(R2) := #0
  (4)  007106  005062  002474                                                          CLR     STATUS(R2)
1937                                  ;
1938                                  ;END OF HARD CODED INCREMENT LOOP
1939                                  ;ENDINC
1940                                  ;
1941   007112  000137  006422                JMP     T1A                       ;UPDATE UNIT #
1942   007116  004737  006204          T1B:  JSR     PC,RESVEC       ; RESET STANDARD VECTORS
1943   007122                                SETPRI #PRI00
  (3)  007122  012700  000000                                                          MOV     #PRI00,R0
  (3)  007126  104441                                                                  TRAP    C$SPRI
1944   007130                        EXIT TST                                  ;EXIT THE TEST
  (3)  007130  104432                                                                  TRAP    C$EXIT
  (3)  007132  000250                                                                  .WORD   L10006-.
1945                                  ;
1946                                  ;INTERRUPT HANDLER TO SERVICE FAULTY INTERRUPT FROM LP INTERFACE.
1947                                  ;THIS ROUTINE IS ENTERED ONLY WHEN THE LP INTERRUPTS AT THE SAME LEVEL AS
1948                                  ;THE CPU AND IS CONSIDERED AN ERROR.
1949                                  ;
1950   007134                         BGNSRV
1951   007134                         INTERR: LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  007134  005262  002676                                                          INC     ERRTBL(R2)
1952   007140                                LET L$LUN := LUNIT
  (4)  007140  013737  002300  002074                                                  MOV     LUNIT,L$LUN
1953   007146                                ERRHRD 5,ERR12
  (4)  007146  104456                                                                  TRAP    C$ERHRD
  (5)  007150  000005                                                                  .WORD   5
  (5)  007152  007246                                                                  .WORD   ERR12
  (5)  007154  000000                                                                  .WORD   0
1954   007156                        LET (SP) := #END2
  (4)  007156  012716  007100                                                          MOV     #END2,(SP)
1955   007162                        ENDSRV
  (3)  007162                                                                  L10007:
  (2)  007162  000002                                                                  RTI
1956                                  ;
1957                                  ;INTERRUPT HANDLER FOR EXPECTED INTERRUPT
1958                                  ;
1959   007164                         BGNSRV
1960                                  ;
1961   007164                         INTHDL: LET (SP) := #END2
  (4)  007164  012716  007100                                                          MOV     #END2,(SP)
1962   007170                        ENDSRV
  (3)  007170                                                                  L10010:
  (2)  007170  000002                                                                  RTI
1963                                  ;
1964                                  ;
1965                                  ;ERROR MESSAGES ASSOCIATED WITH THIS TEST
1966                                  ;
1967                                  .NLIST BEX
1968   007172  047514  042101  047111  ERR11:  .ASCIZ  /LOADING PRINTER BUFFER DOES NOT CLEAR READY/
1969   007246  051120  047111  042524  ERR12:  .ASCIZ  /PRINTER INTERRUPTED AT SAME LEVEL AS THE PROCESSOR/
```

```
 1970  007331      120  044522  052116  ERR13:  .ASCIZ  /PRINTER DID NOT INTERRUPT AT PRIORITY 3/
 1971          007402                  .EVEN
 1972  007402                          ENDTST
  (3)  007402                                                                  L10006:
  (3)  007402  104401                                                                  TRAP    C$ETST
 1973                                  .LIST BEX
 1974  007404                          ENDMOD
 1975
```

```
                        .SBTTL  READY LINE INTERLOCKS

                        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.
                        ;--

                        BGNTST  2
                                                                T2::

                        ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
                                MANUAL
                                                                TRAP    C$MANI
                                BCOMPLETE 11$
                                                                BCS     11$
                                EXIT TST
                                                                TRAP    C$EXIT
                                                                .WORD   L10011-.

                        ;EXIT TEST IF MANUAL INTERVENTION TESTS ARE NOT SPECIFIED
                        11$:    IF INHINT EQ #0 THEN
                                                                TST     INHINT
                                                                BNE     50064$
                                EXIT TST
                                                                TRAP    C$EXIT
                                                                .WORD   L10011-.

                        ENDIF
                                                        50064$:
                                LET FLAG := #0
                                                                CLR     FLAG
                                LET R1 := L$UNIT - #1
                                                                MOV     L$UNIT,R1
                                                                DEC     R1

                        ;CHECK FOR ERROR IN EACH PRINTER UNDER TEST
                            INCR LUNIT FROM #0 TO R1 BY #1
                                                                CLR     LUNIT
                                                                BR      50065$
                                                        50066$:
                                                                INC     LUNIT
                                                        50065$:
                                                                CMP     LUNIT,R1
                                                                BGT     50067$
                                LET R2 := LUNIT SHIFT 1
                                                                MOV     LUNIT,R2
                                                                ASL     R2
                                IF #BIT15 SETIN @LPCSR(R2) THEN
                                                                BIT     #BIT15,@LPCSR(R2
                                                                BEQ     50070$
                                LET ERRTBL(R2) := ERRTBL(R2) + #1
                                                                INC     ERRTBL(R2)
                                    ERRHRD 6, CSRERR
```

```
  (4)  007502  104456                                                                    TRAP    C$ERHRD
  (5)  007504  000006                                                                    .WORD   6
  (5)  007506  003170                                                                    .WORD   CSRERR
  (5)  007510  000000                                                                    .WORD   0
2008   007512                                  LET @LPCSR(R2) := #0
  (4)  007512  005072  002340                                                            CLR     @LPCSR(R2)
2009   007516                              ENDIF
  (4)  007516                                                                  50070$:
2010   007516                          ENDINC
  (4)  007516  000753                                                                    BR      50066$
  (3)  007520                                                                  50067$:
2011                                ;CHECK FOR READY IN EACH PRINTER UNDER TEST
2012   007520                           INCR LUNIT FROM #0 TO R1 BY #1
  (4)  007520  005037  002300                                                            CLR     LUNIT
  (5)  007524  000402                                                                    BR      50071$
  (4)  007526                                                                  50072$:
  (7)  007526  005237  002300                                                            INC     LUNIT
  (5)  007532                                                                  50071$:
  (5)  007532  023701  002300                                                            CMP     LUNIT,R1
  (7)  007536  003016                                                                    BGT     50073$
2013   007540                                  LET R2 := LUNIT SHIFT 1
  (4)  007540  013702  002300                                                            MOV     LUNIT,R2
  (7)  007544  006302                                                                    ASL     R2
2014
2015   007546                                  IF #BIT07 NOTSETIN @LPCSR(R2) THEN
  (6)  007546  032772  000200  002340                                                    BIT     #BIT07,@LPCSR(R2
  (9)  007554  001006                                                                    BNE     50074$
2016   007556                                  LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  007556  005262  002676                                                            INC     ERRTBL(R2)
2017   007562                                      ERRHRD 7, RDYERR
  (4)  007562  104456                                                                    TRAP    C$ERHRD
  (5)  007564  000007                                                                    .WORD   7
  (5)  007566  003206                                                                    .WORD   RDYERR
  (5)  007570  000000                                                                    .WORD   0
2018   007572                                  ENDIF
  (4)  007572                                                                  50074$:
2019   007572                              ENDINC
  (4)  007572  000755                                                                    BR      50072$
  (3)  007574                                                                  50073$:
2020                                ;VERIFY OPERATION OF PAPER LOW INTERLOCK SWITCH
2021                                ;HARD CODED INCREMENT LOOP
2022                                ;
2023   007574                           LET ERRFLG := #0
  (4)  007574  005037  002324                                                            CLR     ERRFLG
2024   007600  005037  002300            CLR LUNIT
2025   007604  000405                    BR  1$
2026   007606                      2$:
2027   007606  005237  002300            INC LUNIT
2028   007612                           LET R2 := LUNIT SHIFT 1
  (4)  007612  013702  002300                                                            MOV     LUNIT,R2
  (7)  007616  006302                                                                    ASL     R2
2029   007620                      1$:
2030   007620  023701  002300            CMP LUNIT,R1
2031   007624  003402                    BLE 3$
2032   007626  000137  010110            JMP 4$
2033   007632                      3$:
```

```
 2034  007632                          LET FLAG := #0
  (4)  007632  005037  002262                                    CLR     FLAG
 2035  007636                          PRINTF #PAPRSW,LUNIT
  (8)  007636  013746  002300                                    MOV     LUNIT,-(SP)
  (7)  007642  012746  010542                                    MOV     #PAPRSW,-(SP)
  (6)  007646  012746  000002                                    MOV     #2,-(SP)
  (3)  007652  010600                                            MOV     SP,R0
  (4)  007654  104417                                            TRAP    C$PNTF
  (4)  007656  062706  000006                                    ADD     #6,SP
 2036  007662                          PRINTF #PAPSW1
  (7)  007662  012746  010612                                    MOV     #PAPSW1,-(SP)
  (6)  007666  012746  000001                                    MOV     #1,-(SP)
  (3)  007672  010600                                            MOV     SP,R0
  (4)  007674  104417                                            TRAP    C$PNTF
  (4)  007676  062706  000004                                    ADD     #4,SP
 2037  007702                          GMANIL READY, FLAG, 100000, YES
  (3)  007702  104443                                            TRAP    C$GMAN
  (3)  007704  000404                                            BR      10000$
  (4)  007706  002262                                            .WORD   FLAG
  (5)  007710  000130                                            .WORD   T$CODE
  (5)  007712  006010                                            .WORD   READY
  (5)  007714  100000                                            .WORD   100000
  (3)  007716                                                10000$:
 2038  007716                          LET LINCNT := #0
  (4)  007716  005037  002264                                    CLR     LINCNT
 2039  007722                          LET OUTBUF := #14
  (4)  007722  012737  000014  002742                            MOV     #14,OUTBUF
 2040  007730                          REPEAT
  (3)  007730                                                50075$:
 2041  007730                                  OUTPUI #OUTBUF,#1,#5$,LUNIT
 2042  007764                                  LET LINCNT := LINCNT + #1
  (6)  007764  005237  002264                                    INC     LINCNT
 2043  007770                          UNTIL LINCNT EQ #3 OR ERRFLG NE #0
  (4)  007770  023727  002264  000003                            CMP     LINCNT,#3
  (6)  007776  001403                                            BEQ     50076$
  (4)  010000  005737  002324                                    TST     ERRFLG
  (7)  010004  001751                                            BEQ     50075$
  (4)  010006                                                50076$:
 2044  010006                          IF ERRFLG EQ #0 THEN
  (6)  010006  005737  002324                                    TST     ERRFLG
  (9)  010012  001006                                            BNE     50077$
 2045  010014                          LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  010014  005262  002676                                    INC     ERRTBL(R2)
 2046  010020                                  ERRHRD 8, PAPSWI
  (4)  010020  104456                                            TRAP    C$ERHRD
  (5)  010022  000010                                            .WORD   8
  (5)  010024  003230                                            .WORD   PAPSWI
  (5)  010026  000000                                            .WORD   0
 2047  010030                          ENDIF
  (4)  010030                                                50077$:
 2048  010030                          PRINTF #PAPRDY,LUNIT
  (8)  010030  013746  002300                                    MOV     LUNIT,-(SP)
  (7)  010034  012746  010670                                    MOV     #PAPRDY,-(SP)
  (6)  010040  012746  000002                                    MOV     #2,-(SP)
  (3)  010044  010600                                            MOV     SP,R0
  (4)  010046  104417                                            TRAP    C$PNTF
```

```
  (4)  010050  062706  000006                                                    ADD     #6,SP
2049   010054                        LET FLAG := #0
  (4)  010054  005037  002262                                                    CLR     FLAG
2050   010060                        GMANIL READY,FLAG,100000,YES
  (3)  010060  104443                                                            TRAP    C$GMAN
  (3)  010062  000404                                                            BR      10001$
  (4)  010064  002262                                                            .WORD   FLAG
  (5)  010066  000130                                                            .WORD   T$CODE
  (5)  010070  006010                                                            .WORD   READY
  (5)  010072  100000                                                            .WORD   100000
  (3)  010074                                                           10001$:
2051   010074  000137  007606                JMP 2$
2052                                ;EXPECTED ERROR HANDLER.
2053                                ;JUST SET EXPECTED ERROR INDICATOR.
2054                                ;
2055   010100                       5$:      LET ERRFLG := #1
  (4)  010100  012737  000001  002324                                            MOV     #1,ERRFLG
2056   010106  000207                RTS PC                                ;AND RETURN
2057                                ;VERIFY OPERATION OF HAMMER BANK INTERLOCK SWITCH
2058   010110                       4$:     INCR LUNIT FROM #0 TO R1 BY #1
  (4)  010110  005037  002300                                                    CLR     LUNIT
  (5)  010114  000402                                                            BR      50100$
  (4)  010116                                                           50101$:
  (7)  010116  005237  002300                                                    INC     LUNIT
  (5)  010122                                                           50100$:
  (5)  010122  023701  002300                                                    CMP     LUNIT,R1
  (7)  010126  003077                                                            BGT     50102$
2059   010130                        LET R2 := LUNIT SHIFT 1
  (4)  010130  013702  002300                                                    MOV     LUNIT,R2
  (7)  010134  006302                                                            ASL     R2
2060   010136                        LET L$LUN := LUNIT
  (4)  010136  013737  002300  002074                                           MOV     LUNIT,L$LUN
2061   010144                        LET FLAG := #0
  (4)  010144  005037  002262                                                    CLR     FLAG
2062   010150                        PRINTF #HAMRSW,LUNIT
  (8)  010150  013746  002300                                                    MOV     LUNIT,-(SP)
  (7)  010154  012746  010753                                                    MOV     #HAMRSW,-(SP)
  (6)  010160  012746  000002                                                    MOV     #2,-(SP)
  (3)  010164  010600                                                            MOV     SP,R0
  (4)  010166  104417                                                            TRAP    C$PNTF
  (4)  010170  062706  000006                                                    ADD     #6,SP
2063   010174                        PRINTF #HAMSW1
  (7)  010174  012746  011037                                                    MOV     #HAMSW1,-(SP)
  (6)  010200  012746  000001                                                    MOV     #1,-(SP)
  (3)  010204  010600                                                            MOV     SP,R0
  (4)  010206  104417                                                            TRAP    C$PNTF
  (4)  010210  062706  000004                                                    ADD     #4,SP
2064   010214                        GMANIL READY, FLAG, 100000, YES
  (3)  010214  104443                                                            TRAP    C$GMAN
  (3)  010216  000404                                                            BR      10002$
  (4)  010220  002262                                                            .WORD   FLAG
  (5)  010222  000130                                                            .WORD   T$CODE
  (5)  010224  006010                                                            .WORD   READY
  (5)  010226  100000                                                            .WORD   100000
  (3)  010230                                                           10002$:
2065   010230                        IF #BIT15 SETIN @LPCSR(R2) THEN
```

```
  (6)  010230  032772  100000  002340                                          BIT     #BIT15,@LPCSR(R2
  (9)  010236  001421                                                          BEQ     50103$
2066   010240                        PRINTF #HAMRDY,LUNIT
  (8)  010240  013746  002300                                                  MOV     LUNIT,-(SP)
  (7)  010244  012746  011114                                                  MOV     #HAMRDY,-(SP)
  (6)  010250  012746  000002                                                  MOV     #2,-(SP)
  (3)  010254  010600                                                          MOV     SP,R0
  (4)  010256  104417                                                          TRAP    C$PNTF
  (4)  010260  062706  000006                                                  ADD     #6,SP
2067   010264                        GMANIL READY, FLAG, 100000, YES
  (3)  010264  104443                                                          TRAP    C$GMAN
  (3)  010266  000404                                                          BR      10003$
  (4)  010270  002262                                                          .WORD   FLAG
  (5)  010272  000130                                                          .WORD   T$CODE
  (5)  010274  006010                                                          .WORD   READY
  (5)  010276  100000                                                          .WORD   100000
  (3)  010300                                                          10003$:
2068   010300                        ELSE
  (4)  010300  000411                                                          BR      50104$
  (3)  010302                                                          50103$:
2069   010302                        LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  010302  005262  002676                                                  INC     ERRTBL(R2)
2070   010306                        LET L$LUN := LUNIT
  (4)  010306  013737  002300  002074                                          MOV     LUNIT,L$LUN
2071
2072   010314                            ERRHRD 9, BNKSWI
  (4)  010314  104456                                                          TRAP    C$ERHRD
  (5)  010316  000011                                                          .WORD   9
  (5)  010320  003273                                                          .WORD   BNKSWI
  (5)  010322  000000                                                          .WORD   0
2073   010324                        ENDIF
  (4)  010324                                                          50104$:
2074   010324                    ENDINC
  (4)  010324  000674                                                          BR      50101$
  (3)  010326                                                          50102$:
2075
2076   010326                ;VERIFY OPERATION OF CHARACTER BAND INTERLOCK SWITCH
                                INCR LUNIT FROM #0 TO R1 BY #1
  (4)  010326  005037  002300                                                  CLR     LUNIT
  (5)  010332  000402                                                          BR      50105$
  (4)  010334                                                          50106$:
  (7)  010334  005237  002300                                                  INC     LUNIT
  (5)  010340                                                          50105$:
  (5)  010340  023701  002300                                                  CMP     LUNIT,R1
  (7)  010344  003074                                                          BGT     50107$
2077   010346                        LET R2 := LUNIT SHIFT 1
  (4)  010346  013702  002300                                                  MOV     LUNIT,R2
  (7)  010352  006302                                                          ASL     R2
2078   010354                            LET FLAG := #0
  (4)  010354  005037  002262                                                  CLR     FLAG
2079   010360                            PRINTF #BANDSW,LUNIT
  (8)  010360  013746  002300                                                  MOV     LUNIT,-(SP)
  (7)  010364  012746  011212                                                  MOV     #BANDSW,-(SP)
  (6)  010370  012746  000002                                                  MOV     #2,-(SP)
  (3)  010374  010600                                                          MOV     SP,R0
  (4)  010376  104417                                                          TRAP    C$PNTF
  (4)  010400  062706  000006                                                  ADD     #6,SP
```

```
 2080  010404                                PRINTF #BNDSW1
  (7)  010404  012746  011300                                                MOV     #BNDSW1,-(SP)
  (6)  010410  012746  000001                                                MOV     #1,-(SP)
  (3)  010414  010600                                                        MOV     SP,R0
  (4)  010416  104417                                                        TRAP    C$PNTF
  (4)  010420  062706  000004                                                ADD     #4,SP
 2081  010424                                GMANIL READY, FLAG, 100000, YES
  (3)  010424  104443                                                        TRAP    C$GMAN
  (3)  010426  000404                                                        BR      10004$
  (4)  010430  002262                                                        .WORD   FLAG
  (5)  010432  000130                                                        .WORD   T$CODE
  (5)  010434  006010                                                        .WORD   READY
  (5)  010436  100000                                                        .WORD   100000
  (3)  010440                                                          10004$:
 2082  010440                                IF #BIT15 SETIN @LPCSR(R2) THEN
  (6)  010440  032772  100000  002340                                        BIT     #BIT15,@LPCSR(R2
  (9)  010446  001421                                                        BEQ     50110$
 2083  010450                                    PRINTF #BNDRDY,LUNIT
  (8)  010450  013746  002300                                                MOV     LUNIT,-(SP)
  (7)  010454  012746  011347                                                MOV     #BNDRDY,-(SP)
  (6)  010460  012746  000002                                                MOV     #2,-(SP)
  (3)  010464  010600                                                        MOV     SP,R0
  (4)  010466  104417                                                        TRAP    C$PNTF
  (4)  010470  062706  000006                                                ADD     #6,SP
 2084  010474                                    GMANIL READY, FLAG, 100000, YES
  (3)  010474  104443                                                        TRAP    C$GMAN
  (3)  010476  000404                                                        BR      10005$
  (4)  010500  002262                                                        .WORD   FLAG
  (5)  010502  000130                                                        .WORD   T$CODE
  (5)  010504  006010                                                        .WORD   READY
  (5)  010506  100000                                                        .WORD   100000
  (3)  010510                                                          10005$:
 2085  010510                                ELSE
  (4)  010510  000411                                                        BR      50111$
  (3)  010512                                                          50110$:
 2086  010512                                    LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  010512  005262  002676                                                INC     ERRTBL(R2)
 2087  010516                                    LET L$LUN := LUNIT
  (4)  010516  013737  002300  002074                                        MOV     LUNIT,L$LUN
 2088  010524                                        ERRHRD 10, BNDSWI
  (4)  010524  104456                                                        TRAP    C$ERHRD
  (5)  010526  000012                                                        .WORD   10
  (5)  010530  003340                                                        .WORD   BNDSWI
  (5)  010532  000000                                                        .WORD   0
 2089  010534                                    ENDIF
  (4)  010534                                                          50111$:
 2090  010534                                ENDINC
  (4)  010534  000677                                                        BR      50106$
  (3)  010536                                                          50107$:
 2091  010536                                EXIT TST
  (3)  010536  104432                                                        TRAP    C$EXIT
  (3)  010540  000710                                                        .WORD   L10011-.
 2092
 2093                                  .NLIST BEX
 2094
 2095  010542  047045  040445  042524  PAPRSW: .ASCIZ /%N%ATEAR OFF PAPER JUST BELOW LUNIT %D2/
```

```
 2096  010612  040445  052040  020117  PAPSW1: .ASCIZ  /%A TO CHECK PAPER LOW %N%AINTERLOCK SWITCH.%N/
 2097  010670  047045  040445  042522  PAPRDY: .ASCIZ  /%N%ARESTORE PAPER AND PLACE LUNIT %D2%A ON LINE.%N/
 2098  010753     045  022516  042101  HAMRSW: .ASCIZ  /%N%ADISENGAGE HAMMER BANK LATCH SWITCH ON LUNIT %D2/
 2099  011037     045  022516  052101  HAMSW1: .ASCIZ  /%N%ATO CHECK HAMMER BANK INTERLOCK SWITCH.%N/
 2100  011114  047045  040445  047105  HAMRDY: .ASCIZ  /%N%AENGAGE HAMMER BANK LATCH AND PLACE LUNIT %D2%A ON LINE.%N/
 2101  011212  047045  040445  050117  BANDSW: .ASCIZ  /%N%AOPEN CHARACTER BAND COVER ON LUNIT %D2%A TO CHECK/
 2102  011300  047045  040445  044103  BNDSW1: .ASCIZ  /%N%ACHARACTER BAND INTERLOCK SWITCH.%N/
 2103  011347     045  022516  041501  BNDRDY: .ASCIZ/%N%ACLOSE CHARACTER BAND COVER ON LUNIT %D2%A AND PLACE ON LINE./
 2104                                          .EVEN
 2105
 2106                                          .LIST BEX
 2107  011450                                  ENDTST
   (3) 011450                                                                  L10011:
   (3) 011450  104401                                                                TRAP    C$ETST
 2108
 2109  011452                                  ENDMOD
 2110
```

```
                                .SBTTL FORMS LENGTH SELECTION
011452                          BGNMOD
                                ;++
                                ;THIS TEST CHECKS ALL POSITIONS OF THE FORM LENGTH SELECT SWITCH. THE
                                ;PROGRAM INDICATES THE SPECIFIED SETTING OF THE FORM LENGTH SELECT SWITCH
                                ;AND WAITS FOR THE OPERATOR TO SET THE SWITCH ON THE PRINTER. THE PAPER
                                ;IS THEN ADVANCED UNDER PROGRAM CONTROL. THE PRINTER OUTPUT IS VISUALLY
                                ;INSPECTED AFTER ALL SWITCH SETTINGS HAVE BEEN RUN THROUGH BY THE OPERATOR
                                ;TO VERIFY THAT THE PROPER PAPER MOVEMENT HAS OCCURRED FOR EACH SWITCH
                                ;SETTING.
                                ;--
011452                          BGNTST  3
(3) 011452                                                                      T3::
                                ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
011452                                  MANUAL
(3) 011452  104450                                                              TRAP    C$MANI
                                        BCOMPLETE 1$
011454
(2) 011454  103402                                                              BCS     1$
                                        EXIT TST
011456
(3) 011456  104432                                                              TRAP    C$EXIT
(3) 011460  002164                                                              .WORD   L10012-.
                                ;EXIT TEST IF MANUAL INTERVENTION TESTS ARE NOT SPECIFIED
011462  005737  002244          1$:     TST INHINT
011466  001002                          BNE 2$
011470                                  EXIT TST
(3) 011470  104432                                                              TRAP    C$EXIT
(3) 011472  002152                                                              .WORD   L10012-.
                                ;PRINT TEST IDENTIFICATION
011474                          2$:     OUTPUT  #FRMLTH,#25.
                                ;
                                ;HARD CODE INCREMENT LOOP
                                ;
011530  005037  002300                  CLR     LUNIT
011534  000402                          BR      4$                      ;COMPARE LOOP
011536  005237  002300          5$:     INC     LUNIT
011542  023701  002300          4$:     CMP     LUNIT,R1
011546  003402                          BLE     6$                      ;EXIT ONLY IF GREATER THAN
011550  000137  012276                  JMP     7$                      ;EXIT
011554                          6$:
011554                                  LET R2 := LUNIT SHIFT 1
(4) 011554  013702  002300                                                      MOV     LUNIT,R2
(7) 011560  006302                                                              ASL     R2
                                    PRINTF #LINSWI,LUNIT        ;PRINT LUNIT MESSAGE
(8) 011562  013746  002300                                                      MOV     LUNIT,-(SP)
(7) 011566  012746  012340                                                      MOV     #LINSWI,-(SP)
(6) 011572  012746  000002                                                      MOV     #2,-(SP)
(3) 011576  010600                                                              MOV     SP,R0
(4) 011600  104417                                                              TRAP    C$PNTF
(4) 011602  062706  000006                                                      ADD     #6,SP
                                    PRINTF #LINSW1              ;SECOND PART OF MESSAGE
(7) 011606  012746  012424                                                      MOV     #LINSW1,-(SP)
(6) 011612  012746  000001                                                      MOV     #1,-(SP)
(3) 011616  010600                                                              MOV     SP,R0
(4) 011620  104417                                                              TRAP    C$PNTF
(4) 011622  062706  000004                                                      ADD     #4,SP
                                    PRINTF #FLSSEL,LUNIT        ;SET TO "FLS" POSITION
```

```
  (8)  011626  013746  002300                                                    MOV     LUNIT,-(SP)
  (7)  011632  012746  012504                                                    MOV     #FLSSEL,-(SP)
  (6)  011636  012746  000002                                                    MOV     #2,-(SP)
  (3)  011642  010600                                                            MOV     SP,R0
  (4)  011644  104417                                                            TRAP    C$PNTF
  (4)  011646  062706  000006                                                    ADD     #6,SP
2148   011652                        INCR R4 FROM #0 TO #20. BY #2
  (4)  011652  005004                                                            CLR     R4
  (5)  011654  000402                                                            BR      50112$
  (4)  011656                                                           50113$:
  (7)  011656  062704  000002                                                    ADD     #2,R4
  (5)  011662                                                           50112$:
  (5)  011662  020427  000024                                                    CMP     R4,#20.
  (7)  011666  003126                                                            BGT     50114$
2149   011670                        LET R3 := R4                    ;INDEX INTO SWITCH SETTING TABLE
  (4)  011670  010403                                                            MOV     R4,R3
2150   011672  006303                ASL R3                          ;ACTUAL OFFSET FOR SWITCH SETTINGS
2151   011674                        LET T3SET := #FFSET + R3
  (4)  011674  012737  013175  012302                                           MOV     #FFSET,T3SET
  (6)  011702  060337  012302                                                    ADD     R3,T3SET
2152   011706                        PRINTF #FLSMSG,LUNIT,T3SET              ;SELECT MESSAGE
  (9)  011706  013746  012302                                                    MOV     T3SET,-(SP)
  (8)  011712  013746  002300                                                    MOV     LUNIT,-(SP)
  (7)  011716  012746  013251                                                    MOV     #FLSMSG,-(SP)
  (6)  011722  012746  000003                                                    MOV     #3,-(SP)
  (3)  011726  010600                                                            MOV     SP,R0
  (4)  011730  104417                                                            TRAP    C$PNTF
  (4)  011732  062706  000010                                                    ADD     #10,SP
2153   011736                        PRINTF #FLSMS1
  (7)  011736  012746  013343                                                    MOV     #FLSMS1,-(SP)
  (6)  011742  012746  000001                                                    MOV     #1,-(SP)
  (3)  011746  010600                                                            MOV     SP,R0
  (4)  011750  104417                                                            TRAP    C$PNTF
  (4)  011752  062706  000004                                                    ADD     #4,SP
2154   011756                        LET FLAG := #0                  ;CLEAR FLAG INDICATOR FOR MANUAL
  (4)  011756  005037  002262                                                    CLR     FLAG
2155   011762                        GMANIL READY,FLAG,100000,YES    ;WAIT FOR OPERATOR
  (3)  011762  104443                                                            TRAP    C$GMAN
  (3)  011764  000404                                                            BR      10000$
  (4)  011766  002262                                                            .WORD   FLAG
  (5)  011770  000130                                                            .WORD   T$CODE
  (5)  011772  006010                                                            .WORD   READY
  (5)  011774  100000                                                            .WORD   100000
  (3)  011776                                                           10000$:
2156   011776                        OUTPUI #REFLIN,#133.,,LUNIT     ;OUTPUT REFERENCE LINE AND TERMINATOR
2157   012032                        OUTPUI T3SET,#3,,LUNIT  ;# OF LINES FOR SPACING
2158   012066                        OUTPUI #MOVMSG,#130.,,LUNIT     ;FINAL REFERENCE LINE
2159   012122                        LET FLAG := #0
  (4)  012122  005037  002262                                                    CLR     FLAG
2160   012126                        GMANIL READY,FLAG,100000,YES
  (3)  012126  104443                                                            TRAP    C$GMAN
  (3)  012130  000404                                                            BR      10001$
  (4)  012132  002262                                                            .WORD   FLAG
  (5)  012134  000130                                                            .WORD   T$CODE
  (5)  012136  006010                                                            .WORD   READY
  (5)  012140  100000                                                            .WORD   100000
```

```
  (3)  012142                                                                  10001$:
2161   012142                          ENDINC
  (4)  012142  000645                                                          BR      50113$
  (3)  012144                                                                  50114$:
2162                                ;SET FORMS LENGTH SELECT SWITCH TO ITS 'REGULAR' SETTING
2163   012144                           PRINTF #NMLFLS,LUNIT
  (8)  012144  013746  002300                                                  MOV     LUNIT,-(SP)
  (7)  012150  012746  013103                                                  MOV     #NMLFLS,-(SP)
  (6)  012154  012746  000002                                                  MOV     #2,-(SP)
  (3)  012160  010600                                                          MOV     SP,R0
  (4)  012162  104417                                                          TRAP    C$PNTF
  (4)  012164  062706  000006                                                  ADD     #6,SP
2164   012170                           LET OUTBUF := #14
  (4)  012170  012737  000014  002742                                          MOV     #14,OUTBUF
2165   012176                           OUTPUI #OUTBUF,#1,,LUNIT
2166   012232                           PRINTF #PAPCHK                ;MAKE SURE MOVEMENT WAS RIGHT
  (7)  012232  012746  012603                                                  MOV     #PAPCHK,-(SP)
  (6)  012236  012746  000001                                                  MOV     #1,-(SP)
  (3)  012242  010600                                                          MOV     SP,R0
  (4)  012244  104417                                                          TRAP    C$PNTF
  (4)  012246  062706  000004                                                  ADD     #4,SP
2167   012252                           LET FLAG := #0                ;CLEAR <CR> FLAG
  (4)  012252  005037  002262                                                  CLR     FLAG
2168   012256                           GMANIL READY,FLAG,100000,YES  ;AND WAIT FOR RESPONSE
  (3)  012256  104443                                                          TRAP    C$GMAN
  (3)  012260  000404                                                          BR      10002$
  (4)  012262  002262                                                          .WORD   FLAG
  (5)  012264  000130                                                          .WORD   T$CODE
  (5)  012266  006010                                                          .WORD   READY
  (5)  012270  100000                                                          .WORD   100000
  (3)  012272                                                                  10002$:
2169   012272  000137  011536                JMP     5$                ;END OF HARDCODED INCREMENT LOOP
2170   012276                           7$:
2171   012276                           EXIT TST
  (3)  012276  104432                                                          TRAP    C$EXIT
  (3)  012300  001344                                                          .WORD   L10012-.
2172   012302  000000                   T3SET:  .WORD   0
2173   012304  000000                   T3MOV:  .WORD   0
2174                                    .NLIST BEX
2175   012306  047506  046522  020123  FRMLTH: .ASCIZ /FORMS LENGTH SELECTION/<12><12><12>
2176   012340  047045  040445  042523  LINSWI: .ASCIZ /%N%ASET LINES SWITCH ON UNIT %D2%A TO "6" TO SELECT/
2177   012424  047045  040445  020066  LINSW1: .ASCIZ /%N%A6 LINES PER INCH VERTICAL PRINTING DENSITY./
2178   012504  047045  022462  051501  FLSSEL: .ASCIZ /%N2%ASET VFU-FLS SWITCH ON UNIT %D2%A TO THE "FLS" POSITION.%N/
2179   012603     045  022516  053101  PAPCHK: .ASCIZ /%N%AVERIFY PROPER PAPER MOVEMENT FOR EACH SWITCH SETTING./
2180   012675     122  043105  051105  REFLIN: .ASCII /REFERENCE LINE FOR FORMS LENGTH SELECTION...................../
2181   012772  027056  027056  027056  REFLI1: .ASCII /.............................................................../
2182   013067     056  027056  027056  REFLI2: .ASCIZ /.........../<14>
2183   013103     045  022516  051501  NMLFLS: .ASCIZ /%N%ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO 11.%N/
2184                                    ;SWITCH SETTINGS FOR FORMS LENGTH MESSAGES
2185   013175     063  020040     000  FFSET:  .ASCIZ /3  /
2186   013201     063  032456     000          .ASCIZ /3.5/
2187   013205     064  020040     000          .ASCIZ /4  /
2188   013211     065  032456     000          .ASCIZ /5.5/
2189   013215     066  020040     000          .ASCIZ /6  /
2190   013221     067  020040     000          .ASCIZ /7  /
2191   013225     070  020040     000          .ASCIZ /8  /
```

```
 2192  013231     070  032456     000          .ASCIZ /8.5/
 2193  013235     061  020061     000          .ASCIZ /11 /
 2194  013241     061  020062     000          .ASCIZ /12 /
 2195  013245     061  020064     000          .ASCIZ /14 /
 2196  013251     045  022516  051501  FLSMSG: .ASCIZ /%N%ASET FORMS LENGTH SELECT SWITCH ON UNIT %D2%A TO %T%A./
 2197  013343     045  022516  042101  FLSMS1: .ASCIZ /%N%ADEPRESS ''ALARM CLEAR'', AND PLACE PRINTER BACK ON LINE.%N/
 2198  013440  044440  041516  042510  MOVMSG: .ASCII / INCHES SHOULD OCCUR BETWEEN THIS AND THE REFERENCE LINE...../
 2199  013535     056  027056  027056  MOVMS1: .ASCII /............................................................/
 2200  013633     056  027056  027056  MOVMS2: .ASCIZ /....../<12>
 2201        013644                          .EVEN
 2202                                  .LIST BEX
 2203  013644                          ENDTST
   (3) 013644                                                          L10012:
   (3) 013644  104401                                                          TRAP    C$ETST
 2204  013646                          ENDMOD
 2205
```

```
                        .SBTTL PRINTING SPEED MEASUREMENT
013646                  BGNMOD
                        ;++
                        ;PRINTING SPEED IS MEASURED BY OPERATING THE LINE PRINTER FOR A GIVEN PERIOD
                        ;OF TIME AND ON THE BASIS OF THE TIME DURATION AND THE NUMBER OF LINES PRINTED,
                        ;WHICH WILL BE PRINTED ON THE CONSOLE TERMINAL, THE PRINTING SPEED IS
                        ;CALCULATED. AVAILABILITY OF A KW11-L OR A KW11-P CLOCK ALLOWS PROGRAMMED
                        ;DETERMINATION OF THE TIME INTERVAL WHICH CAN BE SPECIFIED BY THE OPERATOR TO BE
                        ;ANY VALUE IN THE RANGE FROM 4 TO 60 SECONDS. IF A CLOCK IS NOT INSTALLED IN THE
                        ;SYSTEM, THEN THE OPERATOR CAN PERFORM  THE PRINTING SPEED MEASUREMENT MANUALLY.
                        ;--
013646                  BGNTST  4
013646                                                          T4::
                        ;
                        LET R1 := L$UNIT - #1                   ;NUMBER OF UNITS TO TEST
013646  013701  002012                                                  MOV     L$UNIT,R1
013652  005301                                                          DEC     R1
                        IF MANSPD NE #0 THEN
013654  005737  002252                                                  TST     MANSPD
013660  001416                                                          BEQ     50115$
                            MANUAL                              ;DETERMINE IF MANUAL INTERVENTION ALLOWED
013662  104450                                                          TRAP    C$MANI
                            BCOMPLETE 1$
013664  103402                                                          BCS     1$
                            EXIT TST
013666  104432                                                          TRAP    C$EXIT
013670  002436                                                          .WORD   L10013-.
                        1$:     IF INHINT EQ #0 THEN
013672  005737  002244                                                  TST     INHINT
013676  001003                                                          BNE     50116$
                                    EXIT TST
013700  104432                                                          TRAP    C$EXIT
013702  002424                                                          .WORD   L10013-.
                                ELSE
013704  000403                                                          BR      50117$
013706                  50116$:
                                    LET WORK := PERIOD
013706  013737  002254  002736                                          MOV     PERIOD,WORK
                                ENDIF
013714                  50117$:
                        ELSE
013714  000403                                                          BR      50120$
013716                  50115$:
                            LET WORK := #60.
013716  012737  000074  002736                                          MOV     #60.,WORK
                        ENDIF
013724                  50120$:
                        OUTPUT  #PRTSPD,#29.                ;PRINT TEST ID
                        INCR LUNIT FROM #0 TO R1 BY #1
013760  005037  002300                                                  CLR     LUNIT
013764  000402                                                          BR      50121$
013766                  50122$:
013766  005237  002300                                                  INC     LUNIT
013772                  50121$:
013772  023701  002300                                                  CMP     LUNIT,R1
013776  003161                                                          BGT     50123$
```

```
2235  014000                          LET ERRFLG := #0
 (4)  014000  005037  002324                                                CLR     ERRFLG
2236  014004                          LET R2 := LUNIT SHIFT 1
 (4)  014004  013702  002300                                                MOV     LUNIT,R2
 (7)  014010  006302                                                        ASL     R2
2237  014012                          SELECT CLKTYP OF 4 VERIFY       ;SET UP THE RIGHT CLOCK
 (2)  014012  013746  002306                                                MOV     CLKTYP,-(SP)
 (4)  014016  003403                                                        BLE     50124$
 (3)  014020  021627  000004                                                CMP     (SP),#4
 (6)  014024  003402                                                        BLE     50125$
 (3)  014026                                                          50124$:
 (3)  014026  012716  000005                                                MOV     #5,(SP)
 (3)  014032                                                          50125$:
 (2)  014032  006316                                                        ASL     (SP)
 (2)  014034  060716                                                        ADD     PC,(SP)
 (2)  014036  063607                                                        ADD     @(SP)+,PC
 (3)  014040                                                          50126$:
 (4)  014040  000012                                                        .WORD   50133$-50126$
 (4)  014042  000020                                                        .WORD   50132$-50126$
 (4)  014044  000070                                                        .WORD   50131$-50126$
 (4)  014046  000154                                                        .WORD   50130$-50126$
 (3)  014050  000156                                                        .WORD   50127$-50126$
2238
2239  014052                          CASE 1
 (4)  014052                                                          50133$:
2240  014052  000137  014454              JMP     END4               ;JUST EXIT TEST NO CLOCK AVAILBLE
2241
2242  014056                          CASE 2                          ;KW11-L LINE CLOCK SELECTED
 (4)  014056  000457                                                        BR      50127$
 (4)  014060                                                          50132$:
2243  014060                              LET CLKENA := #100          ;INTERRUPT ENABLE/ CLR MONITOR
 (4)  014060  012737  000100  002320                                        MOV     #100,CLKENA
2244  014066                              LET R4 := CLOCKP
 (4)  014066  013704  002310                                                MOV     CLOCKP,R4
2245  014072                              LET CLKVEC := 4(R4)         ; GET VECTOR FROM P-TABLE
 (4)  014072  016437  000004  002316                                        MOV     4(R4),CLKVEC
2246  014100                              SETVEC CLKVEC,#CLKTCK,#PRI06    ;SET UP INTERRUPT VECTOR
 (7)  014100  012746  000300                                                MOV     #PRI06,-(SP)
 (6)  014104  012746  032216                                                MOV     #CLKTCK,-(SP)
 (5)  014110  013746  002316                                                MOV     CLKVEC,-(SP)
 (4)  014114  012746  000003                                                MOV     #3,-(SP)
 (3)  014120  104437                                                        TRAP    C$SVEC
 (2)  014122  062706  000010                                                ADD     #10,SP
2247
2248  014126                          CASE 3                          ;KW11-P REAL TIME CLOCK
 (4)  014126  000433                                                        BR      50127$
 (4)  014130                                                          50131$:
2249  014130                              LET CLKSET := CLKCSR + #2
 (4)  014130  013737  002312  002314                                        MOV     CLKCSR,CLKSET
 (6)  014136  062737  000002  002314                                        ADD     #2,CLKSET
2250  014144                              LET CLKENA := #111          ;SET UP ENABLE BITS
 (4)  014144  012737  000111  002320                                        MOV     #111,CLKENA
2251                                    ; RUN, RATE = 10KHZ, REPEAT INTR, DOWN,INT ENABLE
2252  014152                              LET R4 := CLOCKP
 (4)  014152  013704  002310                                                MOV     CLOCKP,R4
2253  014156                              LET CLKVEC := 4(R4)         ; GET VECTOR FROM P-TABLE
```

```
 (4)  014156  016437  000004  002316
2254  014164                          SETVEC CLKVEC,#CLKTCK,#PRIO6    ;INTERRUPT VECTOR
 (7)  014164  012746  000300                                                MOV     4(R4),CLKVEC
 (6)  014170  012746  032216                                                MOV     #PRIO6,-(SP)
 (5)  014174  013746  002316                                                MOV     #CLKTCK,-(SP)
 (4)  014200  012746  000003                                                MOV     CLKVEC,-(SP)
 (3)  014204  104437                                                        MOV     #3,-(SP)
 (2)  014206  062706  000010                                                TRAP    C$SVEC
2255                                                                        ADD     #10,SP
2256  014212                          CASE 4
 (4)  014212  000401                                                        BR      50127$
 (4)  014214                                                        50130$:
2257  014214  000240                      NOP                       ;THIS IS JUST A DUMMY
2258  014216                          ENDSELECT
 (3)  014216                                                        50127$:
2259  014216                          LET LINCNT := #0              ;CLEAR LINE COUNTER
 (4)  014216  005037  002264                                                CLR     LINCNT
2260  014222                          LET TIME := #0                ;PRESET SECOND COUNTER TO 0
 (4)  014222  005037  032270                                                CLR     TIME
2261  014226                          LET TICK := #60.              ;SET UP INITIAL CLOCK VALUE
 (4)  014226  012737  000074  032272                                        MOV     #60.,TICK
2262  014234                          IF CHRBND EQ #0 THEN
 (6)  014234  005737  002246                                                TST     CHRBND
 (9)  014240  001017                                                        BNE     50134$
2263  014242                              LET BNDPAT := #TABA64     ;64 CHARACTER BAND
 (4)  014242  012737  015712  015152                                        MOV     #TABA64,BNDPAT
2264  014250                              LET WORK := #133.
 (4)  014250  012737  000205  002736                                        MOV     #133.,WORK
2265  014256                              PRINTF #LPM64             ; SHOULD BE 285 LPM.
 (7)  014256  012746  015547                                                MOV     #LPM64,-(SP)
 (6)  014262  012746  000001                                                MOV     #1,-(SP)
 (3)  014266  010600                                                        MOV     SP,R0
 (4)  014270  104417                                                        TRAP    C$PNTF
 (4)  014272  062706  000004                                                ADD     #4,SP
2266  014276                          ELSE
 (4)  014276  000416                                                        BR      50135$
 (3)  014300                                                        50134$:
2267  014300                              LET BNDPAT := #TABA96      ;96 CHARACTER BAND
 (4)  014300  012737  016120  015152                                        MOV     #TABA96,BNDPAT
2268  014306                              LET WORK := #133.
 (4)  014306  012737  000205  002736                                        MOV     #133.,WORK
2269  014314                              PRINTF #LPM96             ; SHOULD BE 204 LPM.
 (7)  014314  012746  015630                                                MOV     #LPM96,-(SP)
 (6)  014320  012746  000001                                                MOV     #1,-(SP)
 (3)  014324  010600                                                        MOV     SP,R0
 (4)  014326  104417                                                        TRAP    C$PNTF
 (4)  014330  062706  000004                                                ADD     #4,SP
2270  014334                          ENDIF
 (4)  014334                                                        50135$:
2271  014334  004737  014514      JSR PC,REPLUP                     ;DO THE OUTPUT
2272  014340                      ENDINC
 (4)  014340  000612                                                        BR      50122$
 (3)  014342                                                        50123$:
2273                                ;
2274                                ;IF MANUAL PRINT SPEED TESTS HAVE BEEN PERFORMED INSURE PRINTERS ARE
2275                                ;BACK ON LINE WHEN DONE
```

```
2276
2277  014342                          ;
                                      IF CLKTYP EQ #4 THEN
 (6)  014342  023727  002306  000004                                        CMP     CLKTYP,#4
 (9)  014350  001020                                                       BNE     50136$
2278  014352
 (4)  014352  005037  002262              LET FLAG := #0                   ;CLEAR <CR> FLAG
                                                                           CLR     FLAG
2279  014356                              PRINTF #MRESET
 (7)  014356  012746  005731                                               MOV     #MRESET,-(SP)
 (6)  014362  012746  000001                                               MOV     #1,-(SP)
 (3)  014366  010600                                                       MOV     SP,R0
 (4)  014370  104417                                                       TRAP    C$PNTF
 (4)  014372  062706  000004                                               ADD     #4,SP
2280  014376                              GMANIL READY,FLAG,100000,YES     ;WAIT FOR OPERATOR
 (3)  014376  104443                                                       TRAP    C$GMAN
 (3)  014400  000404                                                       BR      10000$
 (4)  014402  002262                                                       .WORD   FLAG
 (5)  014404  000130                                                       .WORD   T$CODE
 (5)  014406  006010                                                       .WORD   READY
 (5)  014410  100000                                                       .WORD   100000
 (3)  014412                                                          10000$:
2281  014412                          ENDIF
 (4)  014412                                                          50136$:
2282
2283  014412                          LET OUTBUF := #14
 (4)  014412  012737  000014  002742                                       MOV     #14,OUTBUF
2284  014420                          OUTPUT #OUTBUF,#1
2285
2286  014454                          END4:   IF CLKTYP EQ #2 OR CLKTYP EQ #3 THEN
 (6)  014454  023727  002306  000002                                        CMP     CLKTYP,#2
 (8)  014462  001404                                                       BEQ     50137$
 (6)  014464  023727  002306  000003                                        CMP     CLKTYP,#3
 (9)  014472  001006                                                       BNE     50140$
 (6)  014474                                                          50137$:
2287  014474                                  CLRVEC CLKVEC
 (3)  014474  013700  002316                                               MOV     CLKVEC,R0
 (3)  014500  104436                                                       TRAP    C$CVEC
2288  014502  012777  000000  165602              MOV     #0,@CLKCSR
2289  014510                                      ENDIF
 (4)  014510                                                          50140$:
2290                                  ;
2291  014510                          EXIT TST
 (3)  014510  104432                                                       TRAP    C$EXIT
 (3)  014512  001614                                                       .WORD   L10013-.
2292                                  ;
2293                                  ;
2294                                  ;THIS IS SUBROUTINED TO DECREASE THE SIZE OF THE INITIAL INCREMENT LOOP.
2295                                  ;
2296                                  ;
2297  014514                          REPLUP:
2298  014514                                  IF CLKTYP EQ #4 THEN
 (6)  014514  023727  002306  000004                                        CMP     CLKTYP,#4
 (9)  014522  001124                                                       BNE     50141$
2299  014524                                      PRINTF #OFFLIN          ;TELL OPERATOR TO PLACE PRINTERS OFFLINE
 (7)  014524  012746  015221                                               MOV     #OFFLIN,-(SP)
 (6)  014530  012746  000001                                               MOV     #1,-(SP)
 (3)  014534  010600                                                       MOV     SP,R0
```

```
  (4)  014536  104417                                                          TRAP    C$PNTF
  (4)  014540  062706  000004                                                  ADD     #4,SP
2300   014544                    LET FLAG := #0
  (4)  014544  005037  002262                                                  CLR     FLAG
2301   014550                    GMANIL READY,FLAG,100000,YES
  (3)  014550  104443                                                          TRAP    C$GMAN
  (3)  014552  000404                                                          BR      10001$
  (4)  014554  002262                                                          .WORD   FLAG
  (5)  014556  000130                                                          .WORD   T$CODE
  (5)  014560  006010                                                          .WORD   READY
  (5)  014562  100000                                                          .WORD   100000
  (3)  014564                                                                10001$:
2302   014564                    PRINTF #ONLIN1,LUNIT        ;PUT LUNIT TO TEST ON LINE
  (8)  014564  013746  002300                                                  MOV     LUNIT,-(SP)
  (7)  014570  012746  015261                                                  MOV     #ONLIN1,-(SP)
  (6)  014574  012746  000002                                                  MOV     #2,-(SP)
  (3)  014600  010600                                                          MOV     SP,R0
  (4)  014602  104417                                                          TRAP    C$PNTF
  (4)  014604  062706  000006                                                  ADD     #6,SP
2303   014610                    PRINTF #ONLIN2,LUNIT        ;END OF TEST.
  (8)  014610  013746  002300                                                  MOV     LUNIT,-(SP)
  (7)  014614  012746  015362                                                  MOV     #ONLIN2,-(SP)
  (6)  014620  012746  000002                                                  MOV     #2,-(SP)
  (3)  014624  010600                                                          MOV     SP,R0
  (4)  014626  104417                                                          TRAP    C$PNTF
  (4)  014630  062706  000006                                                  ADD     #6,SP
2304   014634                    PRINTF #ONLIN3,LUNIT
  (8)  014634  013746  002300                                                  MOV     LUNIT,-(SP)
  (7)  014640  012746  015460                                                  MOV     #ONLIN3,-(SP)
  (6)  014644  012746  000002                                                  MOV     #2,-(SP)
  (3)  014650  010600                                                          MOV     SP,R0
  (4)  014652  104417                                                          TRAP    C$PNTF
  (4)  014654  062706  000006                                                  ADD     #6,SP
2305   014660                    WHILE #BIT15 SETIN @LPCSR(R2) DO   ; WAIT FOR LP SET ON-LINE
  (4)  014660                                                                50142$:
  (6)  014660  032772  100000  002340                                          BIT     #BIT15,@LPCSR(R2
  (9)  014666  001402                                                          BEQ     50143$
2306   014670  000240                        NOP
2307   014672                    ENDDO
  (4)  014672  000772                                                          BR      50142$
  (3)  014674                                                                50143$:
2308   014674                    LET LINCNT := #0
  (4)  014674  005037  002264                                                  CLR     LINCNT
2309   014700                    WHILE #BIT15 NOTSETIN @LPCSR(R2) DO   ; REPEAT UNTIL LP GOES OFF-LINE
  (4)  014700                                                                50144$:
  (6)  014700  032772  100000  002340                                          BIT     #BIT15,@LPCSR(R2
  (9)  014706  001031                                                          BNE     50145$
2310   014710                        LET R5 := BNDPAT
  (4)  014710  013705  015152                                                  MOV     BNDPAT,R5
2311   014714                        LET R3 := WORK
  (4)  014714  013703  002736                                                  MOV     WORK,R3
2312   014720                        WHILE R3 GT #0 DO       ; PRINT R3 CHARACTERS
  (4)  014720                                                                50146$:
  (6)  014720  005703                                                          TST     R3
  (9)  014722  003417                                                          BLE     50147$
2313   014724                            WHILE #BIT7 NOTSETIN @LPCSR(R2) DO  ; WAIT FOR READY
```

```
  (4)  014724                                                                                50150$:
  (6)  014724  032772  000200  002340                                                            BIT     #BIT7,@LPCSR(R2)
  (9)  014732  001007                                                                            BNE     50151$
2314   014734                                  IF #BIT15 SETIN @LPCSR(R2) THEN
  (6)  014734  032772  100000  002340                                                            BIT     #BIT15,@LPCSR(R2
  (9)  014742  001402                                                                            BEQ     50152$
2315   014744  000137  015102                      JMP 99$   ; EXIT LOOP IF OFF-LINE AGAIN
2316   014750                                  ENDIF
  (4)  014750                                                                                50152$:
2317   014750                              ENDDO
  (4)  014750  000765                                                                            BR      50150$
  (3)  014752                                                                                50151$:
2318   014752                              LET @LPBUF(R2) :B= (R5)+   ; PUT CHAR INTO LP BUFFER
  (4)  014752  112572  002434                                                                    MOVB    (R5)+,@LPBUF(R2)
2319   014756                              LET R3 := R3 - #1      ; DECRIMENT CHAR COUNTER
  (6)  014756  005303                                                                            DEC     R3
2320   014760                          ENDDO
  (4)  014760  000757                                                                            BR      50146$
  (3)  014762                                                                                50147$:
2321   014762                          BREAK   ; ALLOW CTL-C ABORT
  (3)  014762  104422                                                                            TRAP    C$BRK
2322   014764                          LET LINCNT := LINCNT + #1
  (6)  014764  005237  002264                                                                    INC     LINCNT
2323   014770                      ENDDO
  (4)  014770  000743                                                                            BR      50144$
  (3)  014772                                                                                50145$:
2324   014772                  ELSE
  (4)  014772  000443                                                                            BR      50153$
  (3)  014774                                                                                50141$:
2325   014774                  IF CLKTYP EQ #3 THEN
  (6)  014774  023727  002306  000003                                                            CMP     CLKTYP,#3
  (9)  015002  001003                                                                            BNE     50154$
2326   015004                      LET @CLKSET := #1666.       ; 1/60 SEC.
  (4)  015004  012777  003202  165302                                                            MOV     #1666.,@CLKSET
2327   015012                  ENDIF
  (4)  015012                                                                                50154$:
2328   015012                  LET @CLKCSR := CLKENA   ;ENABLE THE CLOCK TO DO ITS THING
  (4)  015012  013777  002320  165272                                                            MOV     CLKENA,@CLKCSR
2329   015020                  LET LINCNT := #0
  (4)  015020  005037  002264                                                                    CLR     LINCNT
2330   015024                  WHILE TIME LT PERIOD DO         ; REPEAT UNTIL TIME EXHAUSTED
  (4)  015024                                                                                50155$:
  (6)  015024  023737  032270  002254                                                            CMP     TIME,PERIOD
  (9)  015032  002023                                                                            BGE     50156$
2331   015034                          LET R5 := BNDPAT
  (4)  015034  013705  015152                                                                    MOV     BNDPAT,R5
2332   015040                          LET R3 := WORK
  (4)  015040  013703  002736                                                                    MOV     WORK,R3
2333   015044                          WHILE R3 GT #0 DO       ; SEND R3 CHARACTERS
  (4)  015044                                                                                50157$:
  (6)  015044  005703                                                                            TST     R3
  (9)  015046  003412                                                                            BLE     50160$
2334   015050                              WHILE #BIT7 NOTSETIN @LPCSR(R2) DO  ; WAIT FOR READY
  (4)  015050                                                                                50161$:
  (6)  015050  032772  000200  002340                                                            BIT     #BIT7,@LPCSR(R2)
  (9)  015056  001002                                                                            BNE     50162$
```

```
2335  015060  000240                                        NOP
2336  015062                                              ENDDO
 (4)  015062  000772                                                    BR      50161$
 (3)  015064                                                        50162$:
2337  015064                                              LET @LPBUF(R2) :B= (R5)+          ; PUT DATA INTO BUFFER
 (4)  015064  112572  002434                                           MOVB    (R5)+,@LPBUF(R2)
2338  015070                                              LET R3 := R3 - #1           ; DECRIMENT CHAR COUNTER
 (6)  015070  005303                                                   DEC     R3
2339  015072                                          ENDDO
 (4)  015072  000764                                                    BR      50157$
 (3)  015074                                                        50160$:
2340  015074                                          LET LINCNT := LINCNT + #1
 (6)  015074  005237  002264                                           INC     LINCNT
2341  015100                                      ENDDO
 (4)  015100  000751                                                    BR      50155$
 (3)  015102                                                        50156$:
2342  015102                                  ENDIF
 (4)  015102                                                        50153$:
2343  015102                              99$:
2344  015102  012777  000000  165202          MOV     #0,@CLKCSR
2345                                        ;
2346                                        ; REPORT TOTAL NUMBER OF LINES PRINTED
2347                                        ;
2348  015110                                  PRINTB      #LINPER,LINCNT,LUNIT
 (9)  015110  013746  002300                                           MOV     LUNIT,-(SP)
 (8)  015114  013746  002264                                           MOV     LINCNT,-(SP)
 (7)  015120  012746  015154                                           MOV     #LINPER,-(SP)
 (6)  015124  012746  000003                                           MOV     #3,-(SP)
 (3)  015130  010600                                                   MOV     SP,R0
 (4)  015132  104414                                                   TRAP    C$PNTB
 (4)  015134  062706  000010                                           ADD     #10,SP
2349  015140  000207                                RTS PC                          ;GO BACK AND DO IT AGAIN
2350                                        ;
2351                                        ;EXPECTED ERROR HANDLER
2352                                        ;
2353  015142                                LPERR2: LET ERRFLG := #1                ;SET ERROR FOUND
 (4)  015142  012737  000001  002324                                   MOV     #1,ERRFLG
2354  015150  000207                                RTS     PC                      ;AND EXIT
2355                                        ;
2356                                        ;
2357  015152  000000                        BNDPAT: .WORD   0               ; CONTAINS ADDRESS OF PRINT PATTERN
2358                                        .NLIST BEX
2359                                        ;
2360                                        ;ASSOCIATED MESSAGES
2361                                        ;
2362  015154  047045  042045  022463       LINPER: .ASCIZ  /%N%D3%A LINES PRINTED ON LUNIT %D2%N/
2363  015221     045  022516  044501       OFFLIN: .ASCIZ /%N%AINSURE PRINTER(S) OFF LINE./
2364  015261     045  022516  050101       ONLIN1: .ASCIZ/%N%APLACE LUNIT %D2%A ON LINE TO INITIATE TIME PERIOD FOR MANUAL/
2365  015362  047045  040445  051120       ONLIN2: .ASCIZ  /%N%APRINTING SPEED MEASUREMENT AND BACK OFF LINE TO TERMINATE/
2366  015460  047045  040445  044124       ONLIN3: .ASCIZ /%N%ATHE TIME INTERVAL.%N/
2367  015511     120  044522  052116       PRTSPD: .ASCIZ  /PRINTING SPEED MEASUREMENT/<12><12><12>
2368  015547     045  022516  033101       LPM64:  .ASCIZ  /%N%A64 CHARACTER BAND SHOULD PRINT AT 285 LPM.%N/
2369  015630  047045  040445  033071       LPM96:  .ASCIZ  /%N%A96 CHARACTER BAND SHOULD PRINT AT 204 LPM.%N/
2370                                        .LIST BEX
2371          015712                        .EVEN
2372                                        ;64 CHARACTER BAND  PATTERN  285 LPM
```

```
2373                                    ;
2374                                    .SBTTL  PRINT SPEED TEST PATTERNS
2375
2376  015712  105  061  104  TABA64: .BYTE   105,061,104,075,064,041,103,136,102,060,163
      015715  075  064  041
      015720  103  136  102
      015723  060  163
2377  015725  042  062  134          .BYTE   042,062,134,054,124,101,133,101,133,043,135
      015730  054  124  101
      015733  133  101  133
      015736  043  135
2378  015740  041  105  061          .BYTE   041,105,061,100,075,077,041,056,136,074,060
      015743  100  075  077
      015746  041  056  136
      015751  074  060
2379  015753  076  042  073          .BYTE   076,042,073,042,073,134,055,124,044,133,057
      015756  042  073  134
      015761  055  124  044
      015764  133  057
2380  015766  135  054  105          .BYTE   135,054,105,072,100,050,077,052,056,051,056
      015771  072  100  050
      015774  077  052  056
      015777  051  056
2381  016001  051  074  046          .BYTE   051,074,046,076,071,073,045,055,053,044,137
      016004  076  071  073
      016007  045  055  053
      016012  044  137
2382  016014  057  070  054          .BYTE   057,070,054,132,072,131,072,131,050,067,052
      016017  132  072  131
      016022  072  131  050
      016025  067  052
2383  016027  130  051  127          .BYTE   130,051,127,046,066,071,126,045,125,053,065
      016032  046  066  071
      016035  126  045  125
      016040  053  065
2384  016042  137  123  137          .BYTE   137,123,137,123,070,122,132,121,131,064,067
      016045  123  070  122
      016050  132  121  131
      016053  064  067
2385  016055  120  130  117          .BYTE   120,130,117,124,063,066,116,126,115,126,115
      016060  124  063  066
      016063  116  126  115
      016066  126  115
2386  016070  125  062  065          .BYTE   125,062,065,114,123,113,122,061,121,112,064
      016073  114  123  113
      016076  122  061  121
      016101  112  064
2387  016103  111  120  110          .BYTE   111,120,110,117,060,117,060,063,107,116,106,012,015
      016106  117  060  117
      016111  060  063  107
      016114  116  106  012
      016117  015
2388                                    .EVEN
2389
2390                                .EVEN
2391                                ;
```

```
2392                                    ;96 CHARACTER BAND TABLE  204 LPM.
2393                                    ;
2394                                    ;       MINIMUM PRINT SPEED PATTERN 96 CHARACTER BAND
2395  016120  061  055  144  047  TABA96: .BYTE   061,055,144,047.143,043,142,041,060,052,100
        016123  047  143  043  142
        016126  142  041  060  052
        016131  052  100
2396  016133  075  140  174  176          .BYTE   075,140,174,176,041,056,054,056,054,136,042
        016136  176  041  056  054
        016141  054  056  054  136
        016144  136  042
2397  016146  176  134  173  133          .BYTE   176,134,173,133,175,135,055,164,047,100,043
        016151  133  175  135  055
        016154  055  164  047  100
        016157  100  043
2398  016161  077  041  074  052          .BYTE   077,041,074,041,074,052,062,075,076,174,073
        016164  041  074  052  062
        016167  062  075  076  174
        016172  174  073
2399  016174  041  053  054  071          .BYTE   041,053,054,071,042,057,134,072,133,050,133
        016177  071  042  057  134
        016202  134  072  133  050
        016205  050  133
2400  016207  050  135  051  164          .BYTE   050,135,051,164,070,100,046,124,045,123,044
        016212  164  070  100  046
        016215  046  124  045  123
        016220  123  044
2401  016222  122  067  064  137          .BYTE   122,067,064,137,073,132,073,132,053,131,071
        016225  137  073  132  073
        016230  073  132  053  131
        016233  131  071
2402  016235  066  057  130  072          .BYTE   066,057,130,072,127,050,120,151,125,070,065
        016240  072  127  050  120
        016243  120  151  125  070
        016246  070  065
2403  016250  046  124  046  124          .BYTE   046,124,046,124,045,123,044,122,067,064,137
        016253  124  045  123  044
        016256  044  122  067  064
        016261  064  137
2404  016263  121  132  120  131          .BYTE   121,132,120,131,117,066,063,130,116,130,116
        016266  131  117  066  063
        016271  063  130  116  130
        016274  130  116
2405  016276  127  115  126  114          .BYTE   127,115,126,114,125,113,065,062,124,112,123
        016301  114  125  113  065
        016304  065  062  124  112
        016307  112  123
2406  016311  111  122  110  064          .BYTE   111,122,110,064,061,064,061,121,107,102,106,012,015
        016314  064  061  064  061
        016317  061  121  107  102
        016322  102  106  012  015
        016325  015
2407
2408                                    .EVEN
2409
2410  016326                            ENDTST
```

```
   (3)  016326                                                  L10013:
   (3)  016326  104401                                                 TRAP    C$ETST
  2411  016330                        ENDMOD
```

```
                        .SBTTL DAVFU ERROR DETECTION
                        
016330                  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.
                        ;--
016330                  BGNTST  5
(3) 016330                                                          T5::
                        ;EXIT TEST IF DAVFU OPTION IS NOT SPECIFIED
016330                  IF VFUOPT EQ #0 THEN
(6) 016330  005737  002250                                              TST     VFUOPT
(9) 016334  001002                                                      BNE     50163$
016336                          EXIT TST
(3) 016336  104432                                                      TRAP    C$EXIT
(3) 016340  001272                                                      .WORD   L10014-.
016342                  ENDIF
(4) 016342                                                          50163$:
                        ;PRINT TEST IDENTIFICATION
016342                          OUTPUT #VFUERR, #25., LPERR
                        ;DETERMINE IF MANUAL INTERVENTION IS ALLOWED
016376                          MANUAL
(3) 016376  104450                                                      TRAP    C$MANI
016400                          BCOMPLETE 1$
(2) 016400  103402                                                      BCS     1$
                        ;EXIT TEST IF NEGATIVE DETERMINATION FOR MANUAL
                        ;INTERVENTION IS MADE
016402                          EXIT TST
(3) 016402  104432                                                      TRAP    C$EXIT
(3) 016404  001226                                                      .WORD   L10014-.
                        ;DETERMINE IF INTERVENTION IS INHIBITED
016406  005737  002244  1$:     TST INHINT
016412  001002                  BNE 2$
016414                          EXIT TST
(3) 016414  104432                                                      TRAP    C$EXIT
(3) 016416  001214                                                      .WORD   L10014-.
016420                  2$:     BGNSUB
(3) 016420                                                          T5.1:
(3) 016420  104402                                                      TRAP    C$BSUB
                        ;SEND MESSAGE TO OPERATOR TO SET VFU-FLS SWITCH(ES)
016422                          PRINTF  #VFUSEL
(7) 016422  012746  003615                                              MOV     #VFUSEL,-(SP)
(6) 016426  012746  000001                                              MOV     #1,-(SP)
(3) 016432  010600                                                      MOV     SP,R0
(4) 016434  104417                                                      TRAP    C$PNTF
(4) 016436  062706  000004                                              ADD     #4,SP
                        ;WAIT FOR OPERATOR RESPONSE
016442                          LET FLAG := #0
(4) 016442  005037  002262                                              CLR     FLAG
016446                          GMANIL READY,FLAG,100000,YES
(3) 016446  104443                                                      TRAP    C$GMAN
(3) 016450  000404                                                      BR      10000$
```

```
  (4)  016452  002262                                                                  .WORD   FLAG
  (5)  016454  000130                                                                  .WORD   T$CODE
  (5)  016456  006010                                                                  .WORD   READY
  (5)  016460  100000                                                                  .WORD   100000
  (3)  016462                                                                  10000$:
2447                                ;PERFORM INCOMPLETE DATA ERROR DETECTION TEST
2448   016462                             LET R1 := L$UNIT - #1
  (4)  016462  013701  002012                                                          MOV     L$UNIT,R1
  (6)  016466  005301                                                                  DEC     R1
2449   016470                          INCR LUNIT FROM #0 TO R1 BY #1
  (4)  016470  005037  002300                                                          CLR     LUNIT
  (5)  016474  000402                                                                  BR      50164$
  (4)  016476                                                                  50165$:
  (7)  016476  005237  002300                                                          INC     LUNIT
  (5)  016502                                                                  50164$:
  (5)  016502  023701  002300                                                          CMP     LUNIT,R1
  (7)  016506  003056                                                                  BGT     50166$
2450   016510                             LET R2 := LUNIT SHIFT 1
  (4)  016510  013702  002300                                                          MOV     LUNIT,R2
  (7)  016514  006302                                                                  ASL     R2
2451   016516                             LET FLAG := #0
  (4)  016516  005037  002262                                                          CLR     FLAG
2452   016522                             OUTPUI #INCDAT, #41.., LUNIT
2453   016556                             OUTPUI #INCTBL, #5, #GETFLG, LUNIT
2454   016612                             IF FLAG EQ #0 THEN
  (6)  016612  005737  002262                                                          TST     FLAG
  (9)  016616  001011                                                                  BNE     50167$
2455   016620                             LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  016620  005262  002676                                                          INC     ERRTBL(R2)
2456   016624                             LET L$LUN := LUNIT
  (4)  016624  013737  002300  002074                                                  MOV     LUNIT,L$LUN
2457   016632                                 ERRHRD 11, ERR06
  (4)  016632  104456                                                                  TRAP    C$ERHRD
  (5)  016634  000013                                                                  .WORD   11
  (5)  016636  017502                                                                  .WORD   ERR06
  (5)  016640  000000                                                                  .WORD   0
2458   016642                             ENDIF
  (4)  016642                                                                  50167$:
2459   016642                          ENDINC
  (4)  016642  000715                                                                  BR      50165$
  (3)  016644                                                                  50166$:
2460   016644                          ENDSUB
  (3)  016644                                                                  L10015:
  (3)  016644  104403                                                                  TRAP    C$ESUB
2461
2462   016646                          BGNSUB
  (3)  016646                                                                  T5.2:
  (3)  016646  104402                                                                  TRAP    C$BSUB
2463                                ;PERFORM STOP CODE ERROR DETECTION TEST
2464   016650                          INCR LUNIT FROM #0 TO R1 BY #1
  (4)  016650  005037  002300                                                          CLR     LUNIT
  (5)  016654  000402                                                                  BR      50170$
  (4)  016656                                                                  50171$:
  (7)  016656  005237  002300                                                          INC     LUNIT
  (5)  016662                                                                  50170$:
  (5)  016662  023701  002300                                                          CMP     LUNIT,R1
```

```
  (7)  016666  003147                                                                    BGT     50172$
2465   016670                        LET R2 := LUNIT SHIFT 1
  (4)  016670  013702  002300                                                          MOV     LUNIT,R2
  (7)  016674  006302                                                                  ASL     R2
2466   016676                        LET FLAG := #0
  (4)  016676  005037  002262                                                          CLR     FLAG
2467   016702                        OUTPUI #NOSTOP, #35.,, LUNIT
2468   016736                        OUTPUI #NSTTBL, #6, #GETFLG, LUNIT
2469   016772                        INCR VFUCMD FROM #200 TO #213 BY #1
  (4)  016772  012737  000200  002332                                                  MOV     #200,VFUCMD
  (5)  017000  000402                                                                  BR      50173$
  (4)  017002                                                                  50174$:
  (7)  017002  005237  002332                                                          INC     VFUCMD
  (5)  017006                                                                  50173$:
  (5)  017006  023727  002332  000213                                                  CMP     VFUCMD,#213
  (7)  017014  003052                                                                  BGT     50175$
2470   017016                            LET OUTBUF := #15          ;''CR'' TO OUTPUT BUFFER
  (4)  017016  012737  000015  002742                                                  MOV     #15,OUTBUF
2471   017024                            OUTPUI #OUTBUF, #1, GETFLG, LUNIT
2472   017060                            DELAY 2
  (2)  017060  012727  000002                                                          MOV     #2,(PC)+
  (2)  017064  000000                                                                  .WORD   0
  (2)  017066  013727  002116                                                          MOV     L$DLY,(PC)+
  (2)  017072  000000                                                                  .WORD   0
  (2)  017074  005367  177772                                                          DEC     -6(PC)
  (2)  017100  001375                                                                  BNE     .-4
  (2)  017102  005367  177756                                                          DEC     -22(PC)
  (2)  017106  001367                                                                  BNE     .-20
2473   017110                            IF FLAG EQ #0 THEN
  (6)  017110  005737  002262                                                          TST     FLAG
  (9)  017114  001011                                                                  BNE     50176$
2474   017116                                LET ERRTBL(R2) := ERRTBL(R2) + #1
  (6)  017116  005262  002676                                                          INC     ERRTBL(R2)
2475   017122                                LET L$LUN := LUNIT
  (4)  017122  013737  002300  002074                                                  MOV     LUNIT,L$LUN
2476   017130                                    ERRHRD 12, ERR07
  (4)  017130  104456                                                                  TRAP    C$ERHRD
  (5)  017132  000014                                                                  .WORD   12
  (5)  017134  017553                                                                  .WORD   ERR07
  (5)  017136  000000                                                                  .WORD   0
2477   017140                                ENDIF
  (4)  017140                                                                  50176$:
2478   017140                            ENDINC
  (4)  017140  000720                                                                  BR      50174$
  (3)  017142                                                                  50175$:
2479   017142                            LET OUTBUF := #14          ;''FF'' TO OUTPUT BUFFER
  (4)  017142  012737  000014  002742                                                  MOV     #14,OUTBUF
2480   017150                            OUTPUI #OUTBUF, #1,,LUNIT
2481   017204                        ENDINC
  (4)  017204  000624                                                                  BR      50171$
  (3)  017206                                                                  50172$:
2482   017206                        EXIT TST
  (3)  017206  104432                                                                  TRAP    C$EXIT
  (3)  017210  000422                                                                  .WORD   L10014-.
2483   017212                        ENDSUB
  (3)  017212                                                                  L10016:
```

```
  (3)  017212  104403                                                          TRAP    C$ESUB
2484                          ;
2485                          ;EXPECTED ERROR ROUTINE FOR THIS TEST
2486                          ;
2487  017214                  GETFLG: GMANIL RESET, FLAG, 100000, YES
  (3)  017214  104443                                                          TRAP    C$GMAN
  (3)  017216  000404                                                          BR      10000$
  (4)  017220  002262                                                          .WORD   FLAG
  (5)  017222  000130                                                          .WORD   T$CODE
  (5)  017224  017377                                                          .WORD   RESET
  (5)  017226  100000                                                          .WORD   100000
  (3)  017230                                                              10000$:
2488  017230  000207                  RTS     PC                      ;RETURN
2489                          ;
2490                          ;
2491
2492                          .NLIST BEX
2493  017232  040504  043126  020125  VFUERR: .ASCIZ /DAVFU ERROR DETECTION/<12><12><12>
2494  017263     104  053101  052506  INCDAT: .ASCIZ /DAVFU INCOMPLETE DATA ERROR DETECTION/<12><12><12>
2495  017334  040504  043126  020125  NOSTOP: .ASCIZ /DAVFU STOP CODE ERROR DETECTION/<12><12><12>
2496  017377     124  051505  020124  RESET:  .ASCII /TEST O.K. - PLACE PRINTER ON LINE AND DEPRESS/
2497  017454  051042  052105  051125          .ASCIZ  /"RETURN" WHEN READY./<12>
2498
2499  017502  040504  043126  020125  ERR06:  .ASCIZ /DAVFU INCOMPLETE DATA ERROR NOT DETECTED/
2500  017553     104  053101  052506  ERR07:  .ASCIZ /DAVFU STOP CODE  ERROR NOT DETECTED/
2501  017617     356     001     002  INCTBL: .BYTE 356, 1, 2, 3, 357
2502  017624     356     000     000  NSTTBL: .BYTE 356, 0, 0, 0, 0, 357
2503                          .EVEN
2504
2505
2506
2507
2508
2509                          .LIST BEX
2510  017632                  ENDTST
  (3)  017632                                                              L10014:
  (3)  017632  104401                                                          TRAP    C$ETST
2511
2512  017634                  ENDMOD
```

```
                        .SBTTL DAVFU LINE COUNT PAPER CONTROL

                        BGNMOD
                        ;++
                        ;THIS TEST CHECKS THE LINE COUNT METHOD
                        ;OF PAPER ADVANCE USING THE DAVFU. THE
                        ;DAVFU MEMORY IS LOADED WITH DUMMY
                        ;DATA, AND THEN EACH OF THE LINE COUNT
                        ;SLEWING COMMANDS IS TESTED IN SEQUENCE
                        ;IN THE RANGE FROM ZERO TO 15 LINES.
                        ;--

                        BGNTST  6
                                                                T6::
                        ;EXIT TEST IF DAVFU OPTION IS NOT SPECIFIED
                                IF VFUOPT EQ #0 THEN
                                                                        TST     VFUOPT
                                                                        BNE     50177$
                                    EXIT TST
                                                                        TRAP    C$EXIT
                                                                        .WORD   L10017-.
                                ENDIF
                                                                50177$:
                        ;PRINT TEST IDENTIFICATION
                        OUTPUT #VFULCT, #33., LPERR
                        ;SEND MESSAGE TO OPERATOR TO SET VFU-FLS SWITCH(ES)
                        PRINTF  #VFUSEL
                                                                        MOV     #VFUSEL,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #4,SP
                        ;WAIT FOR OPERATOR RESPONSE
                        LET FLAG := #0
                                                                        CLR     FLAG
                        GMANIL READY,FLAG,100000,YES
                                                                        TRAP    C$GMAN
                                                                        BR      10000$
                                                                        .WORD   FLAG
                                                                        .WORD   T$CODE
                                                                        .WORD   READY
                                                                        .WORD   100000
                                                                10000$:
                        ;INITIALIZE PARAMETERS
                        LET VFUCMD := #200
                                                                        MOV     #200,VFUCMD
                        LET R4 := #0
                                                                        CLR     R4
                        ;LOAD DAVFU MEMORY
                        OUTPUT #VFUTBL, #18.
                        ;PRINT FIRST PART OF ZERO LINE SLEW MESSAGE
                        OUTPUT #FSTMSG, #29.
                        ;SEND ZERO LINE SLEW COMMAND
                        OUTPUT #VFUCMD, #1.
                        ;PRINT SECOND PART OF ZERO LINE SLEW MESSAGE
                        OUTPUT #SECMSG, #103.
```

```
2549  020132                        LET OUTBUF := #15                    ;''CR'' TO OUTPUT BUFFER
 (4)  020132  012737  000015  002742                                      MOV     #15,OUTBUF
2550  020140                        OUTPUT #OUTBUF, #1.
2551                                ;SEND OTHER DAVFU PAPER ADVANCE COMMANDS
2552  020174                        INCR VFUCMD FROM #221 TO #237 BY #1
 (4)  020174  012737  000221  002332                                      MOV     #221,VFUCMD
 (5)  020202  000402                                                      BR      50200$
 (4)  020204                                                        50201$:
 (7)  020204  005237  002332                                      INC     VFUCMD
 (5)  020210                                                        50200$:
 (5)  020210  023727  002332  000237                                      CMP     VFUCMD,#237
 (7)  020216  003101                                                      BGT     50202$
2553                                ;PERFORM PAPER MOVEMENT
2554  020220                            OUTPUT #VFUCMD, #1.
2555  020254                            LET OUTBUF := #15
 (4)  020254  012737  000015  002742                                      MOV     #15,OUTBUF
2556  020262                            OUTPUT #OUTBUF, #1.
2557                                ;INSERT PAPER MOVEMENT VALUE
2558  020316                            LET OUTBUF := LCTTBL(R4)
 (4)  020316  016437  020532  002742                                      MOV     LCTTBL(R4),OUTBU
2559  020324                             OUTPUT #OUTBUF,#2
2560                                ;APPEND MESSAGE AND PRINT MOVEMENT MESSAGE
2561  020360                            OUTPUT #LCTMSG, #131.
2562  020414                            LET R4 := R4 + #2
 (6)  020414  062704  000002                                              ADD     #2,R4
2563  020420                        ENDINC
 (4)  020420  000671                                                      BR      50201$
 (3)  020422                                                        50202$:
2564  020422                        LET OUTBUF := #14
 (4)  020422  012737  000014  002742                                      MOV     #14,OUTBUF
2565  020430                        OUTPUT #OUTBUF, #1.
2566  020464                        EXIT TST
 (3)  020464  104432                                                      TRAP    C$EXIT
 (3)  020466  000536                                                      .WORD   L10017-.
2567
2568                                .NLIST BEX
2569  020470  040504  043126  020125  VFULCT: .ASCIZ /DAVFU LINE COUNT PAPER CONTROL/ <12><12><12>
2570                                        .EVEN
2571  020532  030460  031060  031460  LCTTBL: .ASCIZ/0102030405060708091011121314 15/
2572
2573  020571     356     001     002  VFUTBL: .BYTE 356, 1, 2, 3, 4, 5, 6
2574  020600     007     010     011          .BYTE 7, 10, 11, 12, 13, 14
2575  020606     015     016     017          .BYTE 15, 16, 17, 20, 357
2576
2577  020613     124  044510  020123  FSTMSG: .ASCIZ /THIS LINE SHOULD BE PRINTED /
2578
2579  020650  046101  020114  047117  SECMSG: .ASCII /ALL ON ONE LINE IF SLEWED ZERO LINES/
2580  020714  027056  027056  027056          .ASCII  /............................................................/
2581  021012  027056  000056                  .ASCIZ  /.../
2582  021016  041040  040514  045516  LCTMSG: .ASCII / BLANK LINES SHOULD OCCUR BETWEEN THIS LINE AND THE/
2583  021101     040  051120  053105          .ASCII  / PREVIOUS LINE ................/
2584  021142  027056  027056  027056          .ASCII  /......................................./
2585  021206  027056  027056  027056          .ASCIZ  /.......... / <15>
2586                                .EVEN
2587                                .LIST BEX
2588  021224                        ENDTST
```

```
   (3)  021224
   (3)  021224  104401                                          L10017:
                                                                        TRAP    C$ETST
2589
2590  021226                        ENDMOD
```

```
                        .SBTTL DAVFU CHANNEL SELECTION PAPER CONTROL

021226                  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.
                        ;--

021226                  BGNTST  7
(3) 021226                                                      T7::
                        ;EXIT TEST IF DAVFU OPTION IS NOT SPECIFIED
021226                          IF VFUOPT EQ #0 THEN
(6) 021226  005737  002250                                              TST     VFUOPT
(9) 021232  001002                                                      BNE     50203$
                                    EXIT TST
021234
(3) 021234  104432                                                      TRAP    C$EXIT
(3) 021236  001120                                                      .WORD   L10020-.
                                ENDIF
021240
(4) 021240                                                      50203$:
                        ;PRINT TEST IDENTIFICATION
021240                  OUTPUT #VFUCHL, #40.
                        ;SEND MESSAGE TO OPERATOR TO SET VFU-FLS SWITCH(ES)
021274                  PRINTF  #VFUSEL
(7) 021274  012746  003615                                              MOV     #VFUSEL,-(SP)
(6) 021300  012746  000001                                              MOV     #1,-(SP)
(3) 021304  010600                                                      MOV     SP,R0
(4) 021306  104417                                                      TRAP    C$PNTF
(4) 021310  062706  000004                                              ADD     #4,SP
                        ;WAIT FOR RESPONSE FROM OPERATOR
021314                  LET FLAG := #0
(4) 021314  005037  002262                                              CLR     FLAG
021320                  GMANIL READY,FLAG,100000,YES
(3) 021320  104443                                                      TRAP    C$GMAN
(3) 021322  000404                                                      BR      10000$
(4) 021324  002262                                                      .WORD   FLAG
(5) 021326  000130                                                      .WORD   T$CODE
(5) 021330  006010                                                      .WORD   READY
(5) 021332  100000                                                      .WORD   100000
(3) 021334                                                      10000$:
                        ;LOAD DAVFU MEMORY
021334                  OUTPUT #VFUDAT, #50.
021370  012704  000002  MOV     #2,R4           ;SET UP ITERATION COUNTER
021374  000401          BR      1$
021376  005304  2$:     DEC     R4              ;BACK UP COUNTER
021400  020427  000001  1$:     CMP     R4,#1           ;TEST FOR LAST TIME THROUGH
021404  002002          BGE     3$              ;BRANCH IF LAST TIME THROUGH FALSE
021406  000137  022150          JMP     4$              ;EXIT TEST
                        ;SEND PAPER INSTRUCTIONS TO ALL 12 CHANNELS
021412                  3$: INCR INSTR FROM #200 TO #213 BY #1
(4) 021412  012737  000200  022216                                      MOV     #200,INSTR
(5) 021420  000402                                                      BR      50204$
(4) 021422                                                      50205$:
(7) 021422  005237  022216                                              INC     INSTR
(5) 021426                                                      50204$:
```

```
  (5)  021426  023727  022216  000213                                                    CMP     INSTR,#213
  (7)  021434  003125                                                                    BGT     50206$
2624
2625  021436                        ;PERFORM PAPER MOVEMENT
2626  021472                            OUTPUT #INSTR, #1
                                        LET OUTBUF := #15
  (4)  021472  012737  000015  002742                                                    MOV     #15,OUTBUF
2627  021500                            OUTPUT #OUTBUF, #1
2628                                ;GET SPACE COUNT
2629  021534                            LET R3 := INSTR
  (4)  021534  013703  022216                                                          MOV     INSTR,R3
2630  021540  142703  000360               BICB #360, R3
2631  021544                            WHILE R3 NE #0 DO
  (4)  021544                                                                50207$:
  (6)  021544  005703                                                                  TST     R3
  (9)  021546  001420                                                                  BEQ     50210$
2632  021550                                OUTPUT #CHARSP, #1
2633  021604                                LET R3 := R3 - #1
  (6)  021604  005303                                                                  DEC     R3
2634  021606                            ENDDO
  (4)  021606  000756                                                                  BR      50207$
  (3)  021610                                                                50210$:
2635  021610                            OUTPUT #CHARX, #1
2636                                ;SEND PRINT COMMAND TO PRINT SPACES AND X'S
2637  021644                                LET OUTBUF := #15
  (4)  021644  012737  000015  002742                                                    MOV     #15,OUTBUF
2638  021652                            OUTPUT #OUTBUF, #1.
2639  021706                        ENDINC
  (4)  021706  000645                                                                  BR      50205$
  (3)  021710                                                                50206$:
2640                                ;DUPLICATE TEST TO REVERSE OUTPUT PATTERN
2641  021710                            DECR INSTR FROM #213 TO #200 BY #1
  (4)  021710  012737  000213  022216                                                    MOV     #213,INSTR
  (5)  021716  000402                                                                  BR      50211$
  (4)  021720                                                                50212$:
  (7)  021720  005337  022216                                                          DEC     INSTR
  (5)  021724                                                                50211$:
  (5)  021724  023727  022216  000200                                                    CMP     INSTR,#200
  (7)  021732  002504                                                                  BLT     50213$
2642  021734                        OUTPUT #INSTR, #1.
2643  021770                        LET R3 := INSTR
  (4)  021770  013703  022216                                                          MOV     INSTR,R3
2644  021774  142703  000360           BICB #360, R3
2645  022000                        WHILE R3 NE #0 DO
  (4)  022000                                                                50214$:
  (6)  022000  005703                                                                  TST     R3
  (9)  022002  001420                                                                  BEQ     50215$
2646  022004                            OUTPUT #CHARSP, #1
2647  022040                            LET R3 := R3 - #1
  (6)  022040  005303                                                                  DEC     R3
2648  022042                        ENDDO
  (4)  022042  000756                                                                  BR      50214$
  (3)  022044                                                                50215$:
2649  022044                        OUTPUT #CHARX, #1
2650  022100                        LET OUTBUF := #15
  (4)  022100  012737  000015  002742                                                    MOV     #15,OUTBUF
2651  022106                        OUTPUT #OUTBUF, #1.
```

```
2652  022142                          ENDDEC
 (4)  022142  000666                                                    BR      50212$
 (3)  022144                                                          50213$:
2653  022144  000137  021376          JMP     2$              ;GO BACK AND TRY IT AGAIN
2654  022150                        4$:     LET OUTBUF := #14
 (4)  022150  012737  000014  002742                                    MOV     #14,OUTBUF
2655  022156                                OUTPUT #OUTBUF, #1
2656  022212                        EXIT TST
 (3)  022212  104432                                                    TRAP    C$EXIT
 (3)  022214  000142                                                    .WORD   L10020-.
2657
2658  022216  000000                INSTR:  .WORD 0
2659  022220  040504  043126  020125  VFUCHL: .ASCIZ /DAVFU CHANNEL SELECTION PAPER CONTROL/ <12><12><12>
      022226  044103  047101  042516
      022234  020114  042523  042514
      022242  052103  047511  020116
      022250  040520  042520  020122
      022256  047503  052116  047522
      022264  005114  005012     000
2660          022272                  .EVEN
2661  022272     040                CHARSP: .BYTE 40
2662  022273     130                CHARX:  .BYTE 130
2663
2664  022274     356     001     000  VFUDAT: .BYTE 356, 1, 0, 2, 0, 4, 0
      022277     002     000     004
      022302     000
2665  022303     010     000     020          .BYTE 10, 0, 20, 0, 40, 0, 0, 1
      022306     000     040     000
      022311     000     001
2666  022313     000     002     000          .BYTE 0, 2, 0, 4, 0, 10, 0, 20
      022316     004     000     010
      022321     000     020
2667  022323     000     040     000          .BYTE 0, 40, 0, 40, 0, 20, 0, 10
      022326     040     000     020
      022331     000     010
2668  022333     000     004     000          .BYTE 0, 4, 0, 2, 0, 1, 40, 0
      022336     002     000     001
      022341     040     000
2669  022343     020     000     010          .BYTE 20, 0, 10, 0, 4, 0, 2, 0
      022346     000     004     000
      022351     002     000
2670  022353     001     000     357          .BYTE 1, 0, 357
2671                                  .EVEN
2672  022356                          ENDTST
 (3)  022356                                                          L10020:
 (3)  022356  104401                                                    TRAP    C$ETST
2673
2674  022360                          ENDMOD
```

```
                                .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.
                                ;--

022360                          BGNTST  8.
(3) 022360                                                          T8::
                                ;PRINT TEST IDENTIFICATION
022360                          OUTPUT #DATPTH, #22.
                                ;PRINT ALTERNATING STRINGS OF CHARACTERS
022414                          INCR PATTERN FROM #1 TO #2 BY #1
(4) 022414  012737  000001  022732                                      MOV     #1,PATTERN
(5) 022422  000402                                                      BR      50216$
(4) 022424                                                          50217$:
(7) 022424  005237  022732                                              INC     PATTERN
(5) 022430                                                          50216$:
(5) 022430  023727  022732  000002                                      CMP     PATTERN,#2
(7) 022436  003076                                                      BGT     50220$
022440                              IF PATTERN EQ #1 THEN
(6) 022440  023727  022732  000001                                      CMP     PATTERN,#1
(9) 022446  001004                                                      BNE     50221$
022450                                  LET CHAR :B= #'U
(4) 022450  112737  000125  022702                                      MOVB    #'U,CHAR
022456                              ELSE
(4) 022456  000403                                                      BR      50222$
(3) 022460                                                          50221$:
022460                                  LET CHAR :B= #'?
(4) 022460  112737  000077  022702                                      MOVB    #'?,CHAR
022466                              ENDIF
(4) 022466                                                          50222$:
022466                              LET R4 := #OUTBUF
(4) 022466  012704  002742                                              MOV     #OUTBUF,R4
022472                              INCR CCNT FROM #1 TO #66. BY #1
(4) 022472  012737  000001  002270                                      MOV     #1,CCNT
(5) 022500  000402                                                      BR      50223$
(4) 022502                                                          50224$:
(7) 022502  005237  002270                                              INC     CCNT
(5) 022506                                                          50223$:
(5) 022506  023727  002270  000102                                      CMP     CCNT,#66.
(7) 022514  003011                                                      BGT     50225$
022516                                  LET (R4)+ :B= CHAR
(4) 022516  113724  022702                                              MOVB    CHAR,(R4)+
022522  105137  022702                  COMB        CHAR
022526                                  LET (R4)+ :B= CHAR
```

```
  (4)  022526  113724  022702                                                                    MOVB    CHAR,(R4)+
2708   022532  105137  022702                    COMB        CHAR
2709   022536                                ENDINC
  (4)  022536  000761                                                                            BR      50224$
  (3)  022540                                                                           50225$:
2710   022540                                LET (R4)+ :B= #15
  (4)  022540  112724  000015                                                                    MOVB    #15,(R4)+
2711   022544                                LET (R4) :B= #12
  (4)  022544  112714  000012                                                                    MOVB    #12,(R4)
2712   022550                                INCR LINCNT FROM #1 TO #16. BY #1
  (4)  022550  012737  000001  002264                                                            MOV     #1,LINCNT
  (5)  022556  000402                                                                           BR      50226$
  (4)  022560                                                                           50227$:
  (7)  022560  005237  002264                                                                   INC     LINCNT
  (5)  022564                                                                           50226$:
  (5)  022564  023727  002264  000020                                                           CMP     LINCNT,#16.
  (7)  022572  003017                                                                           BGT     50230$
2713   022574                                    OUTPUT #OUTBUF, #134.
2714   022630                                ENDINC
  (4)  022630  000753                                                                            BR      50227$
  (3)  022632                                                                           50230$:
2715   022632                            ENDINC
  (4)  022632  000674                                                                            BR      50217$
  (3)  022634                                                                           50220$:
2716   022634                            LET OUTBUF :B= #14
  (4)  022634  112737  000014  002742                                                            MOVB    #14,OUTBUF
2717   022642                                    OUTPUT  #OUTBUF, #1
2718   022676                            EXIT TST
  (3)  022676  104432                                                                           TRAP    C$EXIT
  (3)  022700  000034                                                                           .WORD   L10021-.
2719
2720   022702     000                    CHAR:   .BYTE 0
2721   022703     104  052101  020101   DATPTH: .ASCIZ /DATA TRANSFER PATHS/ <12><12><12>
       022710  051124  047101  043123
       022716  051105  050040  052101
       022724  051510  005012  000012
2722
2723                                     .EVEN
2724   022732  000000                    PATTERN:        .WORD 0
2725                                                     .EVEN
2726                                     .EVEN
2727
2728   022734                            ENDTST
  (3)  022734                                                                           L10021:
  (3)  022734  104401                                                                           TRAP    C$ETST
2729
2730   022736                            ENDMOD
```

```
 2732
 2733  022736                          .SBTTL PRINTABLE CHARACTERS
                                       BGNMOD
 2734
 2735                                  ;++
 2736                                  ;THIS TEST CHECKS FOR PROPER PRINTING
 2737                                  ;OF ALL PRINTABLE CHARACTERS. A ROW
 2738                                  ;OF EACH PRINTABLE CHARACTER WILL
 2739                                  ;APPEAR AS OUTPUT ON THE LINE
 2740                                  ;PRINTER.
 2741                                  ;--
 2742  022736                          BGNTST  9.
  (3)  022736                                                          T9::
 2743  022736                          OUTPUT #PRTCHR, #23.
 2744  022772                          IF CHRBND EQ #0 THEN
  (6)  022772  005737  002246                                                 TST     CHRBND
  (9)  022776  001004                                                         BNE     50231$
 2745  023000
  (4)  023000  012737  000137  002736        LET WORK := #137                 MOV     #137,WORK
 2746  023006                          ELSE
  (4)  023006  000403                                                         BR      50232$
  (3)  023010                                                          50231$:
 2747  023010
  (4)  023010  012737  000176  002736        LET WORK := #176                 MOV     #176,WORK
 2748  023016                          ENDIF
  (4)  023016                                                          50232$:
 2749  023016                          INCR CHAR FROM #40 TO WORK BY #1
  (4)  023016  012737  000040  022702                                         MOV     #40,CHAR
  (5)  023024  000402                                                         BR      50233$
  (4)  023026                                                          50234$:
  (7)  023026  005237  022702                                                 INC     CHAR
  (5)  023032                                                          50233$:
  (5)  023032  023737  022702  002736                                         CMP     CHAR,WORK
  (7)  023040  003042                                                         BGT     50235$
 2750  023042                              LET R4 := #OUTBUF
  (4)  023042  012704  002742                                                 MOV     #OUTBUF,R4
 2751  023046                              INCR CCNT FROM #1 TO #132. BY #1
  (4)  023046  012737  000001  002270                                         MOV     #1,CCNT
  (5)  023054  000402                                                         BR      50236$
  (4)  023056                                                          50237$:
  (7)  023056  005237  002270                                                 INC     CCNT
  (5)  023062                                                          50236$:
  (5)  023062  023727  002270  000204                                         CMP     CCNT,#132.
  (7)  023070  003003                                                         BGT     50240$
 2752  023072                                  LET (R4)+ :B= CHAR
  (4)  023072  113724  022702                                                 MOVB    CHAR,(R4)+
 2753  023076                              ENDINC
  (4)  023076  000767                                                         BR      50237$
  (3)  023100                                                          50240$:
 2754  023100                              LET (R4)+ :B= #15
  (4)  023100  112724  000015                                                 MOVB    #15,(R4)+
 2755  023104                              LET (R4)+ :B= #12
  (4)  023104  112724  000012                                                 MOVB    #12,(R4)+
 2756  023110                              OUTPUT #OUTBUF, #134.
 2757  023144                          ENDINC
  (4)  023144  000730                                                         BR      50234$
  (3)  023146                                                          50235$:
```

```
2758  023146                          LET OUTBUF :B= #14
 (4)  023146  112737  000014  002742                                  MOVB    #14,OUTBUF
2759  023154                          OUTPUT #OUTBUF, #1
2760  023210                          EXIT TST
 (3)  023210  104432                                                  TRAP    C$EXIT
 (3)  023212  000032                                                  .WORD   L10022-.
2761
2762  023214  051120  047111  040524  PRTCHR: .ASCIZ /PRINTABLE CHARACTERS/ <12><12><12>
      023222  046102  020105  044103
      023230  051101  041501  042524
      023236  051522  005012  000012
2763                                        .EVEN
2764
2765  023244                          ENDTST
 (3)  023244                                                L10022:
 (3)  023244  104401                                                  TRAP    C$ETST
2766
2767  023246                          ENDMOD
2768
```

```
                                .SBTTL  NON-PRINTABLE CHARACTERS

023246                          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.
                                ;--

023246                          BGNTST  10.
(3)  023246                                                             T10::
                                ;INDICATE TEST CURRENTLY BEING DONE

                                        OUTPUT #NONCHR,#64.
023302                                  LET R4 := #NONBUF
(4)  023302  012704  024137                                                     MOV     #NONBUF,R4
                                ;
                                ; SETUP LINE COUNT ACORDING TO CHRBND SWITCH
                                ;
023306                                  IF CHRBND EQ #0 THEN
(6)  023306  005737  002246                                                     TST     CHRBND
(9)  023312  001004                                                             BNE     50241$
023314                                      LET WORK1 := #32.
(4)  023314  012737  000040  002740                                             MOV     #32.,WORK1
023322                                  ELSE
(4)  023322  000403                                                             BR      50242$
(3)  023324                                                             50241$:
023324                                      LET WORK1 := #27.
(4)  023324  012737  000033  002740                                             MOV     #27.,WORK1
023332                                  ENDIF
(4)  023332                                                             50242$:
                                ;
                                ; DO ONE LINE FOR EACH TABLE ENTRY
                                ;
023332                                  INCR LINCNT FROM #0 TO WORK1 BY #1
(4)  023332  005037  002264                                                     CLR     LINCNT
(5)  023336  000402                                                             BR      50243$
(4)  023340                                                             50244$:
(7)  023340  005237  002264                                                     INC     LINCNT
(5)  023344                                                             50243$:
(5)  023344  023737  002264  002740                                             CMP     LINCNT,WORK1
(7)  023352  003056                                                             BGT     50245$
023354                                      LET R3 := #OUTBUF
(4)  023354  012703  002742                                                     MOV     #OUTBUF,R3
                                ;
                                ; MOVE CODE AND MNEMONIC TO PRINT BUFFER
                                ;
023360                                  INCR WORK FROM #1 TO #8. BY #1
(4)  023360  012737  000001  002736                                             MOV     #1,WORK
(5)  023366  000402                                                             BR      50246$
(4)  023370                                                             50247$:
(7)  023370  005237  002736                                                     INC     WORK
(5)  023374                                                             50246$:
(5)  023374  023727  002736  000010                                             CMP     WORK,#8.
(7)  023402  003002                                                             BGT     50250$
023404                                      LET (R3)+ :B= (R4)+
```

```
  (4)  023404  112423                                                                  MOVB    (R4)+,(R3)+
2802   023406                                ENDINC
  (4)  023406  000770                                                                  BR      50247$
  (3)  023410                                                                50250$:
2803
2804                                ;
2805                                ;  PUT 120 BYTES OF CODE INTO PRINT BUFFER
2806                                ;
2807   023410                               INCR WORK FROM #1 TO #124. BY #1
  (4)  023410  012737  000001  002736                                                 MOV     #1,WORK
  (5)  023416  000402                                                                BR      50251$
  (4)  023420                                                                50252$:
  (7)  023420  005237  002736                                                        INC     WORK
  (5)  023424                                                                50251$:
  (5)  023424  023727  002736  000174                                                CMP     WORK,#124.
  (7)  023432  003002                                                                BGT     50253$
2808   023434                                   LET (R3)+ :B= (R4)
  (4)  023434  111423                                                                MOVB    (R4),(R3)+
2809   023436                               ENDINC
  (4)  023436  000770                                                                BR      50252$
  (3)  023440                                                                50253$:
2810
2811                                ;
2812                                ;  FOLLOWED BY CRLF
2813                                ;
2814   023440                               LET (R3)+ :B= #15
  (4)  023440  112723  000015                                                        MOVB    #15,(R3)+
2815   023444                               LET (R3)+ :B= #12
  (4)  023444  112723  000012                                                        MOVB    #12,(R3)+
2816
2817                                ;
2818                                ;  PRINT LINE OF OCTAL CODE, MNEMONIC, AND 120 BYTES(NONPRINTABLE CODE)
2819   023450                                ;
2820   023504                                   OUTPUT  #OUTBUF,#134.
  (6)  023504  005204                           LET R4 := R4 + #1                    INC     R4
2821   023506                               ENDINC
  (4)  023506  000714                                                                BR      50244$
  (3)  023510                                                                50245$:
2822
2823                                ;
2824                                ;  IF 64 CHAR BAND TEST FOR AUTO CONVERSION FROM LOWER CASE
2825                                ;  TO UPPER CASE CHARACTERS BY THE PRINTER.
2826   023510                                ;
2827   023510                                   IF CHRBND EQ #0 THEN
  (6)  023510  005737  002246                                                        TST     CHRBND
  (9)  023514  001073                                                                BNE     50254$
2828   023516                                      OUTPUT      #SKIP3,#4       ; SKIP 3 LINES
2829   023552                                      OUTPUT      #AUTCON,#46.    ; PRINT CONVERTED  MSG
2830   023606                                      LET R3 := #OUTBUF
  (4)  023606  012703  002742                                                        MOV     #OUTBUF,R3
2831   023612                                      INCR WORK FROM #141 TO #172 BY #1
  (4)  023612  012737  000141  002736                                                MOV     #141,WORK
  (5)  023620  000402                                                                BR      50255$
  (4)  023622                                                                50256$:
  (7)  023622  005237  002736                                                        INC     WORK
  (5)  023626                                                                50255$:
  (5)  023626  023727  002736  000172                                                CMP     WORK,#172
```

```
 (7)  023634  003003                                                                  BGT     50257$
2831  023636                          LET (R3)+ :B= WORK
 (4)  023636  113723  002736                                                          MOVB    WORK,(R3)+
2832  023642                      ENDINC
 (4)  023642  000767                                                                  BR      50256$
 (3)  023644                                                                50257$:
2833  023644                      LET (R3) :B= #12
 (4)  023644  112713  000012                                                          MOVB    #12,(R3)
2834  023650                      OUTPUT  #OUTBUF,#27.        ; LINE OF ALPHABET (LC)
2835  023704                  ENDIF
 (4)  023704                                                                50254$:
2836                        ;
2837                        ; DO A TOP OF FORMS
2838                        ;
2839  023704                      LET OUTBUF :B= #14
 (4)  023704  112737  000014  002742                                                  MOVB    #14,OUTBUF
2840  023712                      OUTPUT  #OUTBUF,#1
2841  023746                      EXIT    TST                     ;AND EXIT TEST
 (3)  023746  104432                                                                  TRAP    C$EXIT
 (3)  023750  000642                                                                  .WORD   L10023-.
2842                        ;
2843                        ;CHARACTER BUFFER AND TEST HEADER MESSAGE
2844                        ;
2845                        .NLIST BEX
2846  023752  047516  026516  051120  NONCHR: .ASCII  /NON-PRINTABLE CHARACTERS/<12>
2847  024003     101  043040  046125           .ASCIZ  /A FULL LINE OF EACH CODE WILL BE SENT/<12><12>
2848  024053     114  053517  051105  AUTCON: .ASCIZ  /LOWER CASE SHOULD BE CONVERTED TO UPPER CASE/<12><12>
2849  024132  005015  005012     000  SKIP3:  .ASCIZ  <15><12><12><12>
2850
2851  024137     040  030060  020060  NONBUF: .ASCII  / 000 NUL/<0>
2852  024150  030040  030460  051440          .ASCII  / 001 SOH/<1>
2853  024161     040  030060  020062          .ASCII  / 002 STX/<2>
2854  024172  030040  031460  042440          .ASCII  / 003 ETX/<3>
2855  024203     040  030060  020064          .ASCII  / 004 EOT/<4>
2856  024214  030040  032460  042440          .ASCII  / 005 ENQ/<5>
2857  024225     040  030060  020066          .ASCII  / 006 ACK/<6>
2858  024236  030040  033460  041040          .ASCII  / 007 BEL/<7>
2859  024247     040  030460  020060          .ASCII  / 010 BS /<10>
2860  024260  030040  030461  044040          .ASCII  / 011 HT /<11>
2861  024271     040  030460  020066          .ASCII  / 016 SO /<12>
2862  024302  030040  033461  051440          .ASCII  / 017 SI /<17>
2863  024313     040  031060  020060          .ASCII  / 020 DLE/<20>
2864  024324  030040  030462  054040          .ASCII  / 021 XON/<21>
2865  024335     040  031060  020062          .ASCII  / 022 DC2/<22>
2866  024346  030040  031462  054040          .ASCII  / 023 XOF/<23>
2867  024357     040  031060  020064          .ASCII  / 024 DC4/<24>
2868  024370  030040  032462  047040          .ASCII  / 025 NAK/<25>
2869  024401     040  031060  020066          .ASCII  / 026 SYN/<26>
2870  024412  030040  033462  042440          .ASCII  / 027 ETB/<27>
2871  024423     040  031460  020060          .ASCII  / 030 CAN/<30>
2872  024434  030040  030463  042440          .ASCII  / 031 EM /<31>
2873  024445     040  031460  020062          .ASCII  / 032 SUB/<32>
2874  024456  030040  031463  042440          .ASCII  / 033 ESC/<33>
2875  024467     040  031460  020064          .ASCII  / 034 FS /<34>
2876  024500  030040  032463  043440          .ASCII  / 035 GS /<35>
2877  024511     040  031460  020066          .ASCII  / 036 RS /<36>
```

```
 2878  024522  030040  033463  052440          .ASCII  / 037 US /<37>
 2879  024533     040  032061  020060          .ASCII  / 140    /
 2880  024543     140                          .BYTE   140
 2881  024544  030440  031467  020040          .ASCII  / 173    /
 2882  024554     173                          .BYTE   173
 2883  024555     040  033461  020064          .ASCII  / 174    /
 2884  024565     174                          .BYTE   174
 2885  024566  030440  032467  020040          .ASCII  / 175    /
 2886  024576     175                          .BYTE   175
 2887  024577     040  033461  020066          .ASCII  / 176    /
 2888  024607     176                          .BYTE   176
 2889  024610     000                          .BYTE   0
 2890          024612                          .EVEN
 2891
 2892                                  .LIST BEX
 2893  024612                          ENDTST
   (3) 024612                                                    L10023:
   (3) 024612  104401                                                  TRAP    C$ETST
 2894
 2895  024614                          ENDMOD
 2896
```

```
2898                                    .SBTTL  BAND PATTERN
2899  024614                            BGNMOD
2900                                    ;++
2901                                    ;BAND PATTERN TEST
2902                                    ;
2903                                    ;THIS TEST PRODUCES AN IMAGE OF THE ENTIRE BAND PATTERN. THE PRINT-OUT
2904                                    ;IS ORGANIZED TO LOCATE THE FOUR  QUADRANTS OF THE BAND IN THE FOLLOWING
2905                                    ;FORMAT:
2906                                    ;
2907                                    ;       QUADRANT NO.1           QUADRANT NO. 2
2908                                    ;       QUADRANT NO.3           QUADRANT NO.4
2909                                    ;
2910                                    ;       QUADRANT NO.1           ETC.
2911                                    ;
2912                                    ;THE REASON FOR THIS ARRANGEMENT IS TO FACILITATE VISUAL INSPECTION
2913                                    ;OF THE PRINTOUT AS WELL AS TO ACCOMODATE THE 208 CHARACTERS OF THE BAND
2914                                    ;IN 132 COLUMNS.
2915                                    ;
2916                                    ; SWITCHES TESTED ARE:  CHRBND = 1 FOR 96 CHAR BAND
2917                                    ;                              = 0 FOR 64 CHAR BAND
2918                                    ;                       USA    =1 FOR AMERICAN PRINT SET
2919                                    ;                              =0 FOR BRITISH  PRINT SET
2920                                    ;
2921                                    ;
2922                                    ;
2923                                    ;--
2924  024614                            BGNTST  11.
  (3) 024614                                                                T11::
2925                                    ;
2926                                    ;PRINT OUT THE TEST HEADER ACCORDING TO SWITCH CHRBND
2927                                    ;
2928  024614                                    IF CHRBND EQ #0 THEN
  (6) 024614  005737  002246                                                TST     CHRBND
  (9) 024620  001017                                                        BNE     50260$
2929  024622                                        OUTPUT #BP64ID,#23.
2930  024656                                    ELSE
  (4) 024656  000416                                                        BR      50261$
  (3) 024660                                                        50260$:
2931  024660                                        OUTPUT #BP96ID,#23.
2932  024714                                    ENDIF
  (4) 024714                                                        50261$:
2933                                    ;
2934                                    ; FIX THE BAND PATTERN ACORDING TO SWITCH 'USA'
2935                                    ;
2936  024714                                    IF USA EQ #1 THEN
  (6) 024714  023727  002256  000001                                        CMP     USA,#1
  (9) 024722  001032                                                        BNE     50262$
2937  024724                                        LET BP64A :B= #0
  (4) 024724  105037  025466                                                CLRB    BP64A
2938  024730                                        LET BP64B :B= #043
  (4) 024730  112737  000043  025472                                        MOVB    #043,BP64B
2939  024736                                        LET BP64C :B= #0
  (4) 024736  105037  025606                                                CLRB    BP64C
2940  024742                                        LET BP64D :B= #043
  (4) 024742  112737  000043  025613                                        MOVB    #043,BP64D
2941  024750                                        LET BP64E :B= #0
```

```
  (4)  024750  105037  025724                                                          CLRB    BP64E
2942   024754
  (4)  024754  112737  000043  025730                                                  MOVB    #043,BP64F
2943   024762
  (4)  024762  105037  026051                                                          CLRB    BP96A
2944   024766
  (4)  024766  112737  000043  026056                                                  MOVB    #043,BP96B
2945   024774
  (4)  024774  105037  026244                                                          CLRB    BP96C
2946   025000
  (4)  025000  112737  000043  026251                                                  MOVB    #043,BP96D
2947   025006                                ELSE
  (4)  025006  000431                                                                  BR      50263$
  (3)  025010                                                                50262$:
2948
2949   025010                             ; SETUP FOR POUND STERLING SIGN
                                                  LET BP64A :B= #043
  (4)  025010  112737  000043  025466                                                  MOVB    #043,BP64A
2950   025016
                                                  LET BP64B :B= #0
  (4)  025016  105037  025472                                                          CLRB    BP64B
2951   025022
                                                  LET BP64C :B= #043
  (4)  025022  112737  000043  025606                                                  MOVB    #043,BP64C
2952   025030
                                                  LET BP64D :B= #0
  (4)  025030  105037  025613                                                          CLRB    BP64D
2953   025034
                                                  LET BP64E :B= #043
  (4)  025034  112737  000043  025724                                                  MOVB    #043,BP64E
2954   025042
                                                  LET BP64F :B= #0
  (4)  025042  105037  025730                                                          CLRB    BP64F
2955   025046
                                                  LET BP96A :B= #043
  (4)  025046  112737  000043  026051                                                  MOVB    #043,BP96A
2956   025054
                                                  LET BP96B :B= #0
  (4)  025054  105037  026056                                                          CLRB    BP96B
2957   025060
                                                  LET BP96C :B= #043
  (4)  025060  112737  000043  026244                                                  MOVB    #043,BP96C
2958   025066
                                                  LET BP96D :B= #0
  (4)  025066  105037  026251                                                          CLRB    BP96D
2959   025072                                ENDIF
  (4)  025072                                                                50263$:
2960
2961   025072                             ;
INCR LINCNT FROM #1 TO #18. BY #1                                  ;1 PAGE OF THE PATTERN
  (4)  025072  012737  000001  002264                                                  MOV     #1,LINCNT
  (5)  025100  000402                                                                  BR      50264$
  (4)  025102                                                                50265$:
  (7)  025102  005237  002264                                                          INC     LINCNT
  (5)  025106                                                                50264$:
  (5)  025106  023727  002264  000022                                                  CMP     LINCNT,#18.
  (7)  025114  003041                                                                  BGT     50266$
2962   025116                                IF CHRBND EQ #0 THEN
  (6)  025116  005737  002246                                                          TST     CHRBND
  (9)  025122  001017                                                                  BNE     50267$
2963   025124                                      OUTPUT #BP64,#BNDCNT
2964   025160                                ELSE
  (4)  025160  000416                                                                  BR      50270$
  (3)  025162                                                                50267$:
2965   025162                                      OUTPUT #BP96,#BNDCT2
2966   025216                                ENDIF
  (4)  025216                                                                50270$:
```

```
 2967  025216                                 ENDINC
  (4)  025216  000731                                                          BR      50265$
  (3)  025220                                                          50266$:
 2968
 2969  025220                                 LET OUTBUF := #14
  (4)  025220  012737  000014  002742                                         MOV     #14,OUTBUF
 2970  025226                                 OUTPUT #OUTBUF,#1
 2971  025262                                 EXIT TST
  (3)  025262  104432                                                         TRAP    C$EXIT
  (3)  025264  001042                                                         .WORD   L10024-.
 2972                                        .NLIST BEX
 2973  025266  033071  041440  040510        BP96ID: .ASCIZ  /96 CHAR BAND PATTERN/<12><12><12>
 2974  025316  032066  041440  040510        BP64ID: .ASCIZ  /64 CHAR BAND PATTERN/<12><12><12>
 2975                                        ;
 2976                                        ;BAND PATTERN TABLE : 64 CHARACTER BAND
 2977                                        ;
 2978  025346  020040  020040  020040        BP64:   .ASCII  /          ABECDTFGOHIJ1KL2MN3OP4QRS5UV6WX7YZ8_+%9&)*(:,/
 2979  025435     057  026444  037073                .ASCII  '/$-;><.        ?@E][T\'0^'
 2980  025466  036443  020461                BP64A:  .ASCII  <043>'=1!'
 2981  025472  031043  040447  041063        BP64B:  .ASCII  <043>/2'A3BC4DFG5HI6JK7LM8NOP9QR*SU,VW-XYZ./<12>
 2982  025541     040  020040  020040                .ASCII  '          +E%&T)(0:/$1;>2<?3@]4[\'5^'
 2983  025606  033043  020475     067        BP64C:  .ASCII  <043>'6=!7'
 2984  025613     043  034047  041101        BP64D:  .ASCII  <043>/'8ABC9DF*GH,IJ-KLM.        NOEPQTRSOUVW1XY2Z_3+%4&)(5/
 2985  025701     072  033057  035444                .ASCII  ':/6$;7><8?@]9[\*'^,'
 2986  025724  036443  020455                BP64E:  .ASCII  <043>'=-!'
 2987  025730  023443  005056     012        BP64F:  .ASCII  <043>/'./<12><12>
 2988  025735     000                        ENDP64: .BYTE   0
 2989                                        .EVEN
 2990          000367                        BNDCNT= ENDP64-BP64
 2991
 2992                                        ;
 2993                                        ; BAND PATTERN TABLE : 96 CHAR BAND
 2994                                        ;
 2995  025736  020040  020040  020040        BP96:   .ASCII  '          ABCDOEFG1HIJ2KLMN3OPQ4RST5UVWX6YZ'
 2996  026011     137  022067  023045                .ASCII  ' 7$%&8)(:/9+;>'
 2997  026027     145                                .BYTE   145
 2998  026030  037474     100                        .ASCII  '<?@'
 2999  026033     164                                .BYTE   164
 3000  026034  020040  020040  020040                .ASCII  '       ][\'',
 3001  026051     043     154                BP96A:  .BYTE   43,154
 3002  026053     075  020452                        .ASCII  '=*!'
 3003  026056     043     047     055        BP96B:  .BYTE   043,47,55,175,173,176,136,56,177,140,141,60,142
 3004  026073     143     144     061                .BYTE   143,144,61,146,147,150,151,62,152,153,154,63
 3005  026107     155     156     157                .BYTE   155,156,157,64,160,161,162,163,65,165,166,167,66
 3006  026124     170     171     172                .BYTE   170,171,172,67,12
 3007  026131     040  020040  020040                .ASCII  '          ABCD8EFG9HIJ'
 3008  026157     145                                .BYTE   145
 3009  026160  046113  047115                        .ASCII  'KLMN'
 3010  026164     164                                .BYTE   164
 3011  026165     117  050520  051054                .ASCII  'OPQ,RST'
 3012  026174  052452  053526  026530                .ASCII  '*UVWX-YZ .$%&0)(:/1+;>2<?@3        ][\''4'
 3013  026244  030443  032475     041        BP96C:  .ASCII  <043>'1=5!'
 3014  026251     043     047     066        BP96D:  .BYTE   043,47,66,175,173,176,136,67,177,140,141,70
 3015  026265     142     143     144                .BYTE   142,143,144,71,146,147,150,151,145,152,153,154
 3016  026301     164     155     156                .BYTE   164,155,156,157,54,160,161,162,163,52,165,166
 3017  026315     167     055     170                .BYTE   167,55,170,171,172,56,12,12
```

```
 3018  026325     000                 ENDP96: .BYTE   0
 3019                                 .EVEN
 3020           000367                BNDCT2=ENDP96-BP96
 3021                                 .EVEN
 3022                                 .LIST BEX
 3023  026326                         ENDTST
  (3)  026326                                                  L10024:
  (3)  026326  104401                                                 TRAP    C$ETST
 3024  026330                         ENDMOD
```

```
                                .SBTTL SPURIOUS HAMMER FIRING

026330                          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.
                                ;--

026330                          BGNTST  12.
026330                                                                  T12::

                                ;PRINT THE TEST HEADER

026330                          OUTPUT #HAMFIR,#25.

                                ;OUTPUT THE ACTUAL WEDGE AT THIS POINT

026364                          INCR WORK FROM #1 TO #132. BY #1                ;NUMBER OF LINES TO OUTPUT
026364  012737  000001  002736                                                  MOV     #1,WORK
026372  000402                                                                  BR      50271$
026374                                                                  50272$:
026374  005237  002736                                                          INC     WORK
026400                                                                  50271$:
026400  023727  002736  000204                                                  CMP     WORK,#132.
026406  003077                                                                  BGT     50273$
                                                                ;ALSO NUMBER OF PRINTING CHARACTERS

026410                              LET R4 := #OUTBUF           ;OUTPUT BUFFER POINTER
026410  012704  002742                                                        MOV     #OUTBUF,R4
026414                              LET SPCCNT := #132. - WORK  ;NUMBER OF SPACES TO FILL IN
026414  012737  000204  026654                                                MOV     #132.,SPCCNT
026422  163737  002736  026654                                                SUB     WORK,SPCCNT

                                ;FILL THE OUTPUT BUFFER WITH THE REQUIRED NUMBER OF SPACES

026430                              WHILE SPCCNT NE #0 DO
026430                                                                  50274$:
026430  005737  026654                                                        TST     SPCCNT
026434  001405                                                              BEQ     50275$
026436                                  LET (R4)+ :B= #40       ;SPACE FILL
026436  112724  000040                                                      MOVB    #40,(R4)+
026442                                  LET SPCCNT := SPCCNT - #1       ;UPDATE FILLER COUNTER
026442  005337  026654                                                      DEC     SPCCNT
026446                              ENDDO
026446  000770                                                              BR      50274$
026450                                                                  50275$:
026450                              LET CCNT := #0
026450  005037  002270                                                      CLR     CCNT
026454                              LET CHRGEN := #77       ;FIRST CHARACTER A "?"
026454  012737  000077  002274                                              MOV     #77,CHRGEN
026462                              LET STRCNT := #33.      ;# OF CHARACTERS IN GROUP
026462  012737  000041  002272                                              MOV     #33.,STRCNT
```

```
3061  026470                          WHILE CCNT LT WORK DO                ;NOW FILL IN REST OF BUFFER
 (4)  026470                                                                                50276$:
 (6)  026470  023737  002270  002736                                                          CMP     CCNT,WORK
 (9)  026476  002022                                                                        BGE     50277$
3062  026500                              IF STRCNT EQ #0 THEN
 (6)  026500  005737  002272                                                               TST     STRCNT
 (9)  026504  001006                                                                       BNE     50300$
3063
3064                                ;RESET  GROUP POINTERS AND COUNTERS
3065
3066  026506                                LET STRCNT := #33.
 (4)  026506  012737  000041  002272                                                       MOV     #33.,STRCNT
3067  026514                                LET CHRGEN := #77
 (4)  026514  012737  000077  002274                                                       MOV     #77,CHRGEN
3068  026522                              ENDIF
 (4)  026522                                                                            50300$:
3069  026522                              LET (R4)+ :B= CHRGEN
 (4)  026522  113724  002274                                                               MOVB    CHRGEN,(R4)+
3070  026526                              LET CHRGEN := CHRGEN + #1
 (6)  026526  005237  002274                                                               INC     CHRGEN
3071  026532                              LET CCNT := CCNT + #1
 (6)  026532  005237  002270                                                               INC     CCNT
3072  026536                              LET STRCNT := STRCNT - #1            ;UPDATE POINTERS AND COUNTERS
 (6)  026536  005337  002272                                                               DEC     STRCNT
3073  026542                          ENDDO
 (4)  026542  000752                                                                       BR      50276$
 (3)  026544                                                                            50277$:
3074
3075                                ;NOW SET UP LINE TERMINATOR AND OUTPUT THE LINE.
3076
3077  026544                              LET (R4)+ :B= #12
 (4)  026544  112724  000012                                                               MOVB    #12,(R4)+
3078                                ;OUTPUT THE  LINE
3079
3080  026550                              OUTPUT #OUTBUF,#133.
3081
3082  026604                          ENDINC
 (4)  026604  000673                                                                       BR      50272$
 (3)  026606                                                                            50273$:
3083
3084  026606                          LET OUTBUF := #14
 (4)  026606  012737  000014  002742                                                       MOV     #14,OUTBUF
3085  026614                          OUTPUT #OUTBUF,#1                    ;END OF TEST FORMFEED
3086  026650                          EXIT TST
 (3)  026650  104432                                                                       TRAP    C$EXIT
 (3)  026652  000036                                                                       .WORD   L10025-.
3087
3088                                ;COUNTERS, POINTERS, TEXT BUFFER,  AND HEADER FOR TEST PRINTOUT
3089
3090  026654  000000                SPCCNT: .WORD 0
3091
3092                                ;TEST HEADER  MESSAGE
3093
3094  026656  050123  051125  047511  HAMFIR: .ASCIZ  /SPURIOUS HAMMER FIRING/<12><12><12>
      026664  051525  044040  046501
      026672  042515  020122  044506
```

```
        026700  044522  043516  005012
        026706  000012
  3095
  3096                                  ;
  3097
  3098                                  .EVEN
  3099
  3100  026710                          ENDTST
   (3)  026710                                                  L10025:
   (3)  026710  104401                                                  TRAP    C$ETST
  3101
  3102  026712                          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. EACH LINE OF THE OUTPUT CONSISTS OF THE SEQUENCE OF
                                ;NUMBERS ''123456789'' FOLLOWED BY A SERIES OF UNDERSCORES AND SHOULD END IN THE
                                ;SEQUENCE '012'' IN COLUMNS 130 THRU 132.
                                ;--
026712                          BGNTST  13.
  (3) 026712                                                                    T13::

                                ;PRINT TEST IDENTIFICATION
026712                          OUTPUT #PRTCTL, #16.
                                ;PRINT 32 LINES, CHARACTER STRING ON EACH LINE SHOULD END IN 123
026746                          DECR LINCNT FROM #31. TO #0 BY #1
  (4) 026746  012737  000037  002264                                            MOV     #31.,LINCNT
  (5) 026754  000402                                                            BR      50301$
  (4) 026756                                                                50302$:
  (7) 026756  005337  002264                                                    DEC     LINCNT
  (5) 026762                                                                50301$:
  (5) 026762  005737  002264                                                    TST     LINCNT
  (7) 026766  002417                                                            BLT     50303$

                                ;
                                ; LOAD THE PRINT BUFFER WITH 139 CHARACTERS AND A LINEFEED
                                ; ONLY THE FIRST 132 CHARACTERS SHOULD PRINT.
                                ;
                                        OUTPUT #CTLBUF, #140.
027024                          ENDDEC
  (4) 027024  000754                                                            BR      50302$
  (3) 027026                                                                50303$:
                                LET OUTBUF := #14
  (4) 027026  012737  000014  002742                                            MOV     #14,OUTBUF
                                OUTPUT #OUTBUF, #1
                                EXIT TST
  (3) 027070  104432                                                            TRAP    C$EXIT
  (3) 027072  000240                                                            .WORD   L10026-.

027074  051120  047111  020124  PRTCTL: .ASCIZ /PRINT CONTROL/ <12><12><12>
027102  047503  052116  047522
027110  005114  005012     000
027115     061  031462  032464  CTLBUF: .ASCII  /1234567890_____________________________/
027122  033466  034470  057460
027130  057537  057537  057537
027136  057537  057537  057537
027144  057537  057537  057537
027152  057537  057537  057537
027160  057537  057537  057537          .ASCII  /_____________________________________/
027166  057537  057537  057537
027174  057537  057537  057537
027202  057537  057537  057537
027210  057537  057537  057537
027216  057537  057537     137
027223     137  057537  057537          .ASCII  /_____________________________________/
027230  057537  057537  057537
027236  057537  057537  057537
```

```
        027244  057537  057537  057537
        027252  057537  057537  057537
        027260  057537  057537  057537
 3133   027266  057537  057537  057537          .ASCIZ  /_____________________0123456789/<12>
        027274  057537  057537  057537
        027302  057537  057537  057537
        027310  057537  057537  057537
        027316  030460  031462  032464
        027324  033466  034470  000012
 3134
 3135                                           .EVEN
 3136
 3137   027332                          ENDTST
  (3)   027332                                                          L10026:
  (3)   027332  104401                                                          TRAP    C$ETST
 3138
 3139   027334                          ENDMOD
 3140
```

```
                                .SBTTL  CRITICAL PATH
                                BGNMOD
                                STARS
                                ;;*****************************************************************
                                ;++
                                ;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
                                ;;*****************************************************************
                                ;
                                ;--
                                BGNTST  14.
                                                                                        T14::
                                ;
                                        OUTPUT #CRTPTH,#16.
                                        IF CHRBND EQ #0 THEN            ;FIRST TEST FOR 64 CHARACTER BAND
                                                                                        TST     CHRBND
                                                                                        BNE     50304$
                                            LET PTHPNT := #TAB64        ;SET UP PATTERN TABLE
                                                                                        MOV     #TAB64,PTHPNT
                                        ELSE
                                                                                        BR      50305$
                                                                                50304$:
                                            LET PTHPNT := #TAB96        ;ELSE SET UP FOR 96 CHARACTER BAND
                                                                                        MOV     #TAB96,PTHPNT
                                        ENDIF
                                                                                50305$:

                                ;
                                ;OUTPUT 32 LINES OF WORST CASE PATTERN
                                ;
                                        INCR LINCNT FROM #1 TO #32. BY #1
                                                                                        MOV     #1,LINCNT
                                                                                        BR      50306$
                                                                                50307$:
                                                                                        INC     LINCNT
                                                                                50306$:
                                                                                        CMP     LINCNT,#32.
                                                                                        BGT     50310$
                                        IF CHRBND EQ #0 THEN
                                                                                        TST     CHRBND
                                                                                        BNE     50311$
                                            OUTPUT PTHPNT,#110.
                                        ELSE
                                                                                        BR      50312$
                                                                                50311$:
                                            OUTPUT PTHPNT,#124.
                                        ENDIF
                                                                                50312$:
                                        ENDINC
                                                                                        BR      50307$
                                                                                50310$:
```

```
3174  027542                          LET OUTBUF := #14
 (4)  027542  012737  000014  002742                                  MOV     #14,OUTBUF
3175  027550                          OUTPUT #OUTBUF,#1
3176                                  ;
3177  027604                          EXIT TST
 (3)  027604  104432                                                  TRAP    C$EXIT
 (3)  027606  000402                                                  .WORD   L10027-.
3178                                  ;
3179  027610  000000                 PTHPNT: .WORD   0
3180                                  ;
3181  027612  051103  052111  041511 CRTPTH: .ASCIZ  /CRITICAL PATH/<12><12><12>
      027620  046101  050040  052101
      027626  005110  005012     000
3182        027634                    .EVEN
3183                                  ;64 CHARACTER BAND PATTERN
3184                                  ;
3185                                  ;       CRITICAL HAMMER FIRE PATTERN 64 CHARACTER BAND
3186  027634     101     064     105 TAB64:  .BYTE   101,064,105,122,104,065,106,126,060,127,111
      027637     122     104     065
      027642     106     126     060
      027645     127     111
3187  027647     067     061     132         .BYTE   067,061,132,114,137,115,045,060,127,111,067
      027652     114     137     115
      027655     045     060     127
      027660     111     067
3188  027662     061     132     114         .BYTE   061,132,114,137,115,045,063,046,120,052,121
      027665     137     115     045
      027670     063     046     120
      027673     052     121
3189  027675     072     123     057         .BYTE   072,123,057,115,045,063,046,120,052,121,072
      027700     115     045     063
      027703     046     120     052
      027706     121     072
3190  027710     123     057     125         .BYTE   123,057,125,055,066,076,130,056,131,100,123
      027713     055     066     076
      027716     130     056     131
      027721     100     123
3191  027723     057     125     055         .BYTE   057,125,055,066,076,130,056,131,100,070,042
      027726     066     076     130
      027731     056     131     100
      027734     070     042
3192  027736     053     124     071         .BYTE   053,124,071,042,051,136,131,100,070,135,053
      027741     042     051     136
      027744     131     100     070
      027747     135     053
3193  027751     124     071     042         .BYTE   124,071,042,051,136,050,075,054,041,044,062
      027754     051     136     050
      027757     075     054     041
      027762     044     062
3194  027764     073     101     051         .BYTE   073,101,051,136,050,075,054,041,044,062,073
      027767     136     050     075
      027772     054     041     044
      027775     062     073
3195  027777     101     074     102         .BYTE   101,074,102,077,064,105,106,133,065,015,012,000
      030002     077     064     105
      030005     106     133     065
```

```
                030010      015      012      000
3196
3197                        030014                        .EVEN
3198
3199                                                      ;
3200                                                      ;96 CHARACTER TABLE
3201                                                      ;
3202                                                      ;       CRITICAL HAMMER FIRE PATTERN 96 CHARACTER BAND
3203            030014      101      064      103      TAB96:  .BYTE   101,064,103,123,060,065,106,126,061,130,111
                030017      123      060      065
                030022      106      126      061
                030025      130      111
3204            030027      131      062      137              .BYTE   131,062,137,114,044,116,046,061,130,111,131
                030032      114      044      116
                030035      046      061      130
                030040      111      131
3205            030042      062      137      114              .BYTE   062,137,114,044,116,046,117,051,121,072,122
                030045      044      116      046
                030050      117      051      121
                030053      072      122
3206            030055      071      124      073              .BYTE   071,124,073,116,046,117,051,121,072,122,071
                030060      116      046      117
                030063      051      121      072
                030066      122      071
3207            030070      124      073      125              .BYTE   124,073,125,145,127,077,066,136,132,133,124
                030073      145      127      077
                030076      066      136      132
                030101      133      124
3208            030103      073      125      145              .BYTE   073,125,145,127,077,066,136,132,133,067,042
                030106      127      077      066
                030111      136      132      133
                030114      067      042
3209            030116      045      040      070              .BYTE   045,040,070,075,050,041,132,133,067,042,045
                030121      075      050      041
                030124      132      133      067
                030127      042      045
3210            030131      040      070      075              .BYTE   040,070,075,050,041,057,047,053,175,076,176
                030134      050      041      057
                030137      047      053      175
                030142      076      176
3211            030144      074      056      050              .BYTE   074,056,050,041,057,047,053,175,076,176,074
                030147      041      057      047
                030152      053      175      076
                030155      176      074
3212            030157      056      100      140              .BYTE   056,100,140,135,060,134,143,054,061,040,040
                030162      135      060      134
                030165      143      054      061
                030170      040      040
3213            030172      040      040      040              .BYTE   040,040,040,060,040,040,040,040,040,040,040
                030175      060      040      040
                030200      040      040      040
                030203      040      040
3214            030205      150      015      012              .BYTE   150,015,012
3215
3216                                                      .EVEN
3217
```

```
 3218  030210                          ENDTST
  (3)  030210                                                  L10027:
  (3)  030210  104401                                                 TRAP    C$ETST
 3219  030212                          ENDMOD
 3220
```

```
                                .SBTTL  MULTIPLE LINE ADVANCE

030212                          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.
                                ;--


030212                          BGNTST  15.
(3) 030212                                                              T15::

                                ;PRINT TEST IDENTIFICATION

030212                          OUTPUT #MULINE,#78.

030246                          LET STACHR := #TABSTR           ;OUTPUT CHARACTERS
(4) 030246  012737  030500  030476                                      MOV     #TABSTR,STACHR

030254                          REPEAT
(3) 030254                                                          50313$:
030254                              LET LINCNT :B= @STACHR              ;GET A CHARACTER TO OUTPUT
(4) 030254  117737  000216  002264                                      MOVB    @STACHR,LINCNT
030262                              LET LINCNT := LINCNT AND #7 ;MAKE THE ASCII TO OCTAL
(6) 030262  013746  002264                                              MOV     LINCNT,-(SP)
(6) 030266  042716  000007                                              BIC     #7,(SP)
(6) 030272  042637  002264                                              BIC     (SP)+,LINCNT
030276                              LET R3 := #OUTBUF           ;SET UP OUTPUT BUFFER
(4) 030276  012703  002742                                              MOV     #OUTBUF,R3
030302                              INCR CCNT FROM #1 TO #132. BY #1
(4) 030302  012737  000001  002270                                      MOV     #1,CCNT
(5) 030310  000402                                                      BR      50314$
(4) 030312                                                          50315$:
(7) 030312  005237  002270                                              INC     CCNT
(5) 030316                                                          50314$:
(5) 030316  023727  002270  000204                                      CMP     CCNT,#132.
(7) 030324  003003                                                      BGT     50316$
030326                                  LET (R3)+ :B= @STACHR       ;PUT CHARACTER  IN OUTPUT BUFFER
(4) 030326  117723  000144                                              MOVB    @STACHR,(R3)+
030332                              ENDINC
(4) 030332  000767                                                      BR      50315$
(3) 030334                                                          50316$:
030334                              LET R4 := #0
(4) 030334  005004                                                      CLR     R4
030336                              WHILE R4 NE LINCNT DO
(4) 030336                                                          50317$:
(6) 030336  020437  002264                                              CMP     R4,LINCNT
(9) 030342  001404                                                      BEQ     50320$
030344                                  LET (R3)+ :B= #12           ;FILL WITH LINE FEEDS
(4) 030344  112723  000012                                              MOVB    #12,(R3)+
030350                                  LET R4 := R4 + #1
(6) 030350  005204                                                      INC     R4
030352                              ENDDO
```

```
  (4)  030352  000771                                                                  BR      50317$
  (3)  030354                                                                50320$:
3254
3255                                ;NOW OUTPUT THE ACTUAL LINE
3256
3257  030354                            LET R4 := LINCNT + #132.          ;NUMBER OF CHARACTERS TO OUTPUT
  (4)  030354  013704  002264                                                          MOV     LINCNT,R4
  (6)  030360  062704  000204                                                          ADD     #132.,R4
3258  030364                            LET STACHR := STACHR + #1         ; UPDATE CHARACTER COUNT
  (6)  030364  005237  030476                                                          INC     STACHR
3259  030370                            OUTPUT #OUTBUF,R4           ;OUTPUT THE LINE
3260
3261  030422                            UNTIL LINCNT EQ #0
  (3)  030422  005737  002264                                                          TST     LINCNT
  (6)  030426  001312                                                                  BNE     50313$
3262  030430                            LET OUTBUF := #14
  (4)  030430  012737  000014  002742                                                  MOV     #14,OUTBUF
3263  030436                            OUTPUT #OUTBUF,#1
3264
3265  030472                            EXIT TST
  (3)  030472  104432                                                                  TRAP    C$EXIT
  (3)  030474  000140                                                                  .WORD   L10030-.
3266
3267
3268  030476  000000                    STACHR: .WORD 0
3269
3270  030500  033462  033062  033463    TABSTR: .ASCIZ  /272637463540/
       030506  033064  032463  030064
       030514     000
3271  030515     115  046125  044524    MULINE: .ASCII  /MULTIPLE LINE ADVANCE/<12>
       030522  046120  020105  044514
       030530  042516  040440  053104
       030536  047101  042503     012
3272  030543     116  046525  042502            .ASCIZ  /NUMBERS PRINTED REPRESENT # LINES TO NEXT LINE PRINTED/<12><12>
       030550  051522  050040  044522
       030556  052116  042105  051040
       030564  050105  042522  042523
       030572  052116  021440  046040
       030600  047111  051505  052040
       030606  020117  042516  052130
       030614  046040  047111  020105
       030622  051120  047111  042524
       030630  005104  000012
3273
3274
3275
3276                                .EVEN
3277
3278  030634                            ENDTST
  (3)  030634                                                                L10030:
  (3)  030634  104401                                                                  TRAP    C$ETST
3279  030636                            ENDMOD
3280
```

```
                                .SBTTL  CHARACTER ALIGNMENT
030636                          BGNMOD
                                ;++
                                ;THIS TEST CHECKS CHARACTER ALIGNMENT BY OVERPRINTING LINES OF ALTERNATING
                                ;E'S AND SPACES. THE STARTING CHARACTER OF EACH LINE IS ALSO ALTERNATED TO
                                ;PRODUCE A CHECKERBOARD PATTERN.
                                ;--
030636                          BGNTST  16.
(3) 030636                                                                      T16::
                                ;PRINT TEST IDENTIFICATION
030636                          OUTPUT #CHRALN,#22.
                                ;PRINT 48 LINES OF ALTERNATING 'E'"S AND ''SPACE'"S
030672                          1$:     LET LINCNT := #24.
(4) 030672  012737  000030  002264                                           MOV     #24.,LINCNT
030700                          2$:     IF LINCNT LE #0 THEN
(6) 030700  005737  002264                                                  TST     LINCNT
(9) 030704  003002                                                          BGT     50321$
030706                                      INLINE <JMP 3$>
(2) 030706  000137  031312                                                  JMP     3$
030712                                  ENDIF
(4) 030712                                                                  50321$:
                                ;LOAD BUFFER WITH ALTERNATING STRING OF 'E'"S AND ''SPACE'"S
030712                                  OUTPUT  #ESPSTR,#132.
                                ;PRINT LINE
030746                                  LET OUTBUF := #15
(4) 030746  012737  000015  002742                                          MOV     #15,OUTBUF
030754                                  OUTPUT #OUTBUF,#1
                                ;OVERPRINT LINE
031010                                  OUTPUT  #ESPSTR,#132.
031044                                  LET OUTBUF := #12
(4) 031044  012737  000012  002742                                          MOV     #12,OUTBUF
031052                                  OUTPUT #OUTBUF,#1
                                ;NOW THE ALTERNATE PATTERN
                                ;LOAD BUFFER WITH ALTERNATING STRING OF 'E'"S AND ''SPACE'"S
031106                                  OUTPUT  #ESPSTR+1,#132.
                                ;PRINT LINE
031142                                  LET OUTBUF := #15
(4) 031142  012737  000015  002742                                          MOV     #15,OUTBUF
031150                                  OUTPUT #OUTBUF,#1
                                ;OVERPRINT LINE
031204                                  OUTPUT  #ESPSTR+1,#132.
031240                                  LET OUTBUF := #12
(4) 031240  012737  000012  002742                                          MOV     #12,OUTBUF
031246                                  OUTPUT #OUTBUF,#1
031302                                  LET LINCNT := LINCNT - #1
(6) 031302  005337  002264                                                  DEC     LINCNT
031306                                  INLINE <JMP     2$>
(2) 031306  000137  030700                                                  JMP     2$
031312                          3$:
031312                          LET OUTBUF := #14
(4) 031312  012737  000014  002742                                          MOV     #14,OUTBUF
031320                          OUTPUT #OUTBUF,#1
031354                          EXIT TST
(3) 031354  104432                                                          TRAP    C$EXIT
(3) 031356  000236                                                          .WORD   L10031-.
031360  044103  051101  041501  CHRALN: .ASCIZ /CHARACTER ALIGNMENT/<12><12><12>
```

```
                031366  042524  020122  046101
                031374  043511  046516  047105
                031402  005124  005012     000
 3323   031407      105     040     105  ESPSTR: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031412      040     105     040
                031415      105     040     105
                031420      040     105     040
                031423      105     040     105
                031426      040     105     040
 3324   031431      105     040     105  ESPST1: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031434      040     105     040
                031437      105     040     105
                031442      040     105     040
                031445      105     040     105
                031450      040     105     040
 3325   031453      105     040     105  ESPST2: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031456      040     105     040
                031461      105     040     105
                031464      040     105     040
                031467      105     040     105
                031472      040     105     040
 3326   031475      105     040     105  ESPST3: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031500      040     105     040
                031503      105     040     105
                031506      040     105     040
                031511      105     040     105
                031514      040     105     040
 3327   031517      105     040     105  ESPST4: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031522      040     105     040
                031525      105     040     105
                031530      040     105     040
                031533      105     040     105
                031536      040     105     040
 3328   031541      105     040     105  ESPST5: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031544      040     105     040
                031547      105     040     105
                031552      040     105     040
                031555      105     040     105
                031560      040     105     040
 3329   031563      105     040     105  ESPST6: .BYTE 105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40,105,40
                031566      040     105     040
                031571      105     040     105
                031574      040     105     040
                031577      105     040     105
                031602      040     105     040
 3330   031605      105     040     105  ESPST7: .BYTE 105,40,105,40,105,40,105
                031610      040     105     040
                031613      105
 3331                                     .EVEN
 3332   031614                           ENDTST
  (3)   031614                                                          L10031:
  (3)   031614  104401                                                          TRAP    C$ETST
 3333   031616                           ENDMOD
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINES
031616                          BGNSRV
                                ;
                                ;++
                                ;INTERRUPT VECTORS ARE ESTABLISHED DURING INITIALIZATION
                                ;POINTING TO THE BASIC ROUTINES WHICH
                                ;SET UP THE UNIT NUMBER CAUSING THE INTERRUPTS.
                                ;LINE NUMBER IS RETURNED IN R2
                                ;
                                ;--
000000                          X=0
031616  000020                  INT00:  .REPT 16.
                                        SETPRI #PRI04
                                        PUSH    R2
                                        LET R2 := #X
                                        INLINE <JMP     IODRV>
                                X=X+2
                                        .ENDR
(4)  031616  012700  000200                                             MOV     #PRI04,R0
(4)  031622  104441                                                     TRAP    C$SPRI
(3)  031624  010246                                                     MOV     R2,-(SP)
(5)  031626  012702  000000                                             MOV     #X,R2
(3)  031632  000137  004166                                             JMP     IODRV
(4)  031636  012700  000200                                             MOV     #PRI04,R0
(4)  031642  104441                                                     TRAP    C$SPRI
(3)  031644  010246                                                     MOV     R2,-(SP)
(5)  031646  012702  000002                                             MOV     #X,R2
(3)  031652  000137  004166                                             JMP     IODRV
(4)  031656  012700  000200                                             MOV     #PRI04,R0
(4)  031662  104441                                                     TRAP    C$SPRI
(3)  031664  010246                                                     MOV     R2,-(SP)
(5)  031666  012702  000004                                             MOV     #X,R2
(3)  031672  000137  004166                                             JMP     IODRV
(4)  031676  012700  000200                                             MOV     #PRI04,R0
(4)  031702  104441                                                     TRAP    C$SPRI
(3)  031704  010246                                                     MOV     R2,-(SP)
(5)  031706  012702  000006                                             MOV     #X,R2
(3)  031712  000137  004166                                             JMP     IODRV
(4)  031716  012700  000200                                             MOV     #PRI04,R0
(4)  031722  104441                                                     TRAP    C$SPRI
(3)  031724  010246                                                     MOV     R2,-(SP)
(5)  031726  012702  000010                                             MOV     #X,R2
(3)  031732  000137  004166                                             JMP     IODRV
(4)  031736  012700  000200                                             MOV     #PRI04,R0
(4)  031742  104441                                                     TRAP    C$SPRI
(3)  031744  010246                                                     MOV     R2,-(SP)
(5)  031746  012702  000012                                             MOV     #X,R2
(3)  031752  000137  004166                                             JMP     IODRV
(4)  031756  012700  000200                                             MOV     #PRI04,R0
(4)  031762  104441                                                     TRAP    C$SPRI
(3)  031764  010246                                                     MOV     R2,-(SP)
(5)  031766  012702  000014                                             MOV     #X,R2
(3)  031772  000137  004166                                             JMP     IODRV
(4)  031776  012700  000200                                             MOV     #PRI04,R0
(4)  032002  104441                                                     TRAP    C$SPRI
(3)  032004  010246                                                     MOV     R2,-(SP)
```

```
   (5)  032006  012702  000016          MOV     #X,R2
   (3)  032012  000137  004166          JMP     IODRV
   (4)  032016  012700  000200          MOV     #PRI04,R0
   (4)  032022  104441                  TRAP    C$SPRI
   (3)  032024  010246                  MOV     R2,-(SP)
   (5)  032026  012702  000020          MOV     #X,R2
   (3)  032032  000137  004166          JMP     IODRV
   (4)  032036  012700  000200          MOV     #PRI04,R0
   (4)  032042  104441                  TRAP    C$SPRI
   (3)  032044  010246                  MOV     R2,-(SP)
   (5)  032046  012702  000022          MOV     #X,R2
   (3)  032052  000137  004166          JMP     IODRV
   (4)  032056  012700  000200          MOV     #PRI04,R0
   (4)  032062  104441                  TRAP    C$SPRI
   (3)  032064  010246                  MOV     R2,-(SP)
   (5)  032066  012702  000024          MOV     #X,R2
   (3)  032072  000137  004166          JMP     IODRV
   (4)  032076  012700  000200          MOV     #PRI04,R0
   (4)  032102  104441                  TRAP    C$SPRI
   (3)  032104  010246                  MOV     R2,-(SP)
   (5)  032106  012702  000026          MOV     #X,R2
   (3)  032112  000137  004166          JMP     IODRV
   (4)  032116  012700  000200          MOV     #PRI04,R0
   (4)  032122  104441                  TRAP    C$SPRI
   (3)  032124  010246                  MOV     R2,-(SP)
   (5)  032126  012702  000030          MOV     #X,R2
   (3)  032132  000137  004166          JMP     IODRV
   (4)  032136  012700  000200          MOV     #PRI04,R0
   (4)  032142  104441                  TRAP    C$SPRI
   (3)  032144  010246                  MOV     R2,-(SP)
   (5)  032146  012702  000032          MOV     #X,R2
   (3)  032152  000137  004166          JMP     IODRV
   (4)  032156  012700  000200          MOV     #PRI04,R0
   (4)  032162  104441                  TRAP    C$SPRI
   (3)  032164  010246                  MOV     R2,-(SP)
   (5)  032166  012702  000034          MOV     #X,R2
   (3)  032172  000137  004166          JMP     IODRV
   (4)  032176  012700  000200          MOV     #PRI04,R0
   (4)  032202  104441                  TRAP    C$SPRI
   (3)  032204  010246                  MOV     R2,-(SP)
   (5)  032206  012702  000036          MOV     #X,R2
   (3)  032212  000137  004166          JMP     IODRV
3353
```

```
                                .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.
                                ;--

032216                          BGNSRV
032216                          CLKTCK: SETPRI #PRIO6
  (3) 032216  012700  000300                                                    MOV     #PRIO6,R0
  (3) 032222  104441                                                            TRAP    C$SPRI
                                        IF TICK EQ #0 THEN
  (6) 032224  005737  032272                                                    TST     TICK
  (9) 032230  001005                                                            BNE     50322$
                                            LET TICK := #60.            ;60 TICKS PER SECOND
  (4) 032232  012737  000074  032272                                            MOV     #60.,TICK
                                            LET TIME := TIME + #1
  (6) 032240  005237  032270                                                    INC     TIME
                                        ENDIF
  (4) 032244                                                            50322$:
                                        LET TICK := TICK - #1           ;BACK UP SECOND TIMER
  (6) 032244  005337  032272                                                    DEC     TICK
                                        IF CLKTYP EQ #2 THEN
  (6) 032250  023727  002306  000002                                            CMP     CLKTYP,#2
  (9) 032256  001003                                                            BNE     50323$
                                            LET @CLKCSR := #100
  (4) 032260  012777  000100  150024                                            MOV     #100,@CLKCSR
                                        ENDIF
  (4) 032266                                                            50323$:

032266                          ENDSRV                          ;AND EXIT
  (3) 032266                                                            L10033:
  (2) 032266  000002                                                            RTI
                                ;
032270  000000                  TIME:   .WORD   0
032272  000000                  TICK:   .WORD   0
```

```
                                .SBTTL  HARDWARE PARAMETER SECTION
032274                          BGNMOD

                                ;++
                                ;THIS SECTION INCLUDES THE QUESTIONS WHICH REQUEST THE OPERATOR TO
                                ;FURNISH THE HARDWARE INFORMATION NECESSARY TO BUILD THE HARDWARE
                                ;P-TABLES.
                                ;
                                ;--
032274                          BGNHRD
(3) 032274  000010                                                      .WORD L10034-L$HARD/2
(3) 032276                                                          L$HARD::

032276                                  GPRMA   GETADR,0,0,160000,177516,YES
(4) 032276  000031                                                      .WORD   T$CODE
(4) 032300  032316                                                      .WORD   GETADR
(4) 032302  160000                                                      .WORD   T$LOLIM
(4) 032304  177516                                                      .WORD   T$HILIM

032306                                  GPRMA   GETVEC,2,0,110,770,YES
(4) 032306  001031                                                      .WORD   T$CODE
(4) 032310  032333                                                      .WORD   GETVEC
(4) 032312  000110                                                      .WORD   T$LOLIM
(4) 032314  000770                                                      .WORD   T$HILIM

032316                          ENDHRD
(2)                                                                     .EVEN
(3) 032316                                                          L10034:

032316  050114  030461  040440  GETADR: .ASCIZ  /LP11 ADDRESS/
032324  042104  042522  051523
032332     000
032333     111  052116  051105  GETVEC: .ASCIZ  /INTERRUPT VECTOR/
032340  052522  052120  053040
032346  041505  047524  000122

                                .EVEN
```

```
3399                                    .SBTTL  SOFTWARE PARAMETER SECTION
3400                                    ;
3401                                    ;++
3402                                    ;THIS SECTION INCLUDES THE QUESTIONS WHICH REQUEST THE OPERATOR TO FURNISH
3403                                    ;THE SOFTWARE INFORMATION NECESSARY TO BUILD THE SOFTWARE P-TABLES.
3404                                    ;--
3405                                    ;
3406  032354                            BGNSFT
 (3)  032354  000031                                                        .WORD L10035-L$SOFT/2
 (3)  032356                                                              L$SOFT::
3407  032356                                    GPRML   MGTINT,0,1,YES
 (4)  032356  000130                                                        .WORD   T$CODE
 (4)  032360  032440                                                        .WORD   MGTINT
 (4)  032362  000001                                                        .WORD   1
3408
3409  032364                                    GPRML   GETBND,2,1,YES
 (4)  032364  001130                                                        .WORD   T$CODE
 (4)  032366  032476                                                        .WORD   GETBND
 (4)  032370  000001                                                        .WORD   1
3410
3411  032372                                    GPRML   GETDAV,4,1,YES
 (4)  032372  002130                                                        .WORD   T$CODE
 (4)  032374  032520                                                        .WORD   GETDAV
 (4)  032376  000001                                                        .WORD   1
3412  032400                                    GPRML   GETMAN,6,1,YES
 (4)  032400  003130                                                        .WORD   T$CODE
 (4)  032402  032547                                                        .WORD   GETMAN
 (4)  032404  000001                                                        .WORD   1
3413  032406                                    GPRMD   GETTIM,10,D,377,4,60.,YES
 (4)  032406  004052                                                        .WORD   T$CODE
 (4)  032410  032621                                                        .WORD   GETTIM
 (4)  032412  000377                                                        .WORD   377
 (4)  032414  000004                                                        .WORD   T$LOLIM
 (4)  032416  000074                                                        .WORD   T$HILIM
3414  032420                                    GPRML   GETPLA,12,1,YES
 (4)  032420  005130                                                        .WORD   T$CODE
 (4)  032422  032706                                                        .WORD   GETPLA
 (4)  032424  000001                                                        .WORD   1
3415  032426                                    GPRMD   GETMAX,14,D,377,1,255.,YES
 (4)  032426  006052                                                        .WORD   T$CODE
 (4)  032430  032727                                                        .WORD   GETMAX
 (4)  032432  000377                                                        .WORD   377
 (4)  032434  000001                                                        .WORD   T$LOLIM
 (4)  032436  000377                                                        .WORD   T$HILIM
3416
3417  032440                            ENDSFT
 (2)                                                                        .EVEN
 (3)  032440                                                              L10035:
3418
3419                                    .NLIST BEX
3420  032440  052522  020116  040515    MGTINT: .ASCIZ   /RUN MANUAL INTERVENTION TESTS/
3421  032476  033071  041440  040510    GETBND: .ASCIZ  /96 CHARACTER BAND/
3422  032520  040504  043126  020125    GETDAV: .ASCIZ  /DAVFU OPTION INSTALLED/
3423  032547     120  051105  047506    GETMAN: .ASCIZ  /PERFORM MANUAL PRINTING SPEED MEASUREMENT/
3424  032621     104  051505  051111    GETTIM: .ASCIZ  /DESIRED TIME INTERVAL FOR PRINTING SPEED CALCULATION/
3425  032706  042524  052123  047111    GETPLA: .ASCIZ  /TESTING IN U.S.A/
```

```
 3426  032727     101  052125  042117  GETMAX: .ASCIZ  /AUTODROP ERROR COUNT/
 3427                                  .LIST BEX
 3428                                  .EVEN
 3429                                  ;
 3430
 3431  032754  000020                  PATCH:  .BLKW   20
 3432  033014                          LASTAD
   (2)                                                                .EVEN
   (4)  033014  000000                                                .WORD   0
   (4)  033016  000000                                                .WORD   0
   (3)  033020                        L$LAST::
 3433  033020                          ENDMOD
 3434          000001                  .END
```

```
ACTIVE= 100000
ADR   = 000020 G
ASSEMB= 000010
AUTCON  024053
BANDSW  011212
BIT0  = 000001 G
BIT00 = 000001 G
BIT01 = 000002 G
BIT02 = 000004 G
BIT03 = 000010 G
BIT04 = 000020 G
BIT05 = 000040 G
BIT06 = 000100 G
BIT07 = 000200 G
BIT08 = 000400 G
BIT09 = 001000 G
BIT1  = 000002 G
BIT10 = 002000 G
BIT11 = 004000 G
BIT12 = 010000 G
BIT13 = 020000 G
BIT14 = 040000 G
BIT15 = 100000 G
BIT2  = 000004 G
BIT3  = 000010 G
BIT4  = 000020 G
BIT5  = 000040 G
BIT6  = 000100 G
BIT7  = 000200 G
BIT8  = 000400 G
BIT9  = 001000 G
BNDCNT= 000367
BNDCT2= 000367
BNDPAT  015152
BNDRDY  011347
BNDSWI  003340
BNDSW1  011300
BNKSWI  003273
BOE   = 000400 G
BP64    025346
BP64A   025466
BP64B   025472
BP64C   025606
BP64D   025613
BP64E   025724
BP64F   025730
BP64ID  025316
BP96    025736
BP96A   026051
BP96B   026056
BP96C   026244
BP96D   026251
BP96ID  025266

BUFADD  002334
BUFCNT  002336
CCNT    002270
CHAR    022702
CHARSP  022272
CHARX   022273
CHRALN  031360
CHRBND  002246
CHRGEN  002274
CK1     005426
CLEAN   006270
CLKCSR  002312
CLKENA  002320
CLKSET  002314
CLKTCK  032216
CLKTYP  002306
CLKVEC  002316
CLOCKP  002310
CRTPTH  027612
CSRERR  003170
CTLBUF  027115
CTLEND  004650
CTLLOP  004370
CURADD  002534
CURCNT  002574
C$AU  = 000052
C$AUTO= 00006i
C$BRK = 000022
C$BSEG= 000004
C$BSUB= 000002
C$CEFG= 000045
C$CLCK= 000062
C$CLEA= 000012
C$CLOS= 000035
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000051
C$DRPT= 000024
C$DU  = 000053
C$EDIT= 000003
C$ERDF= 000055
C$ERHR= 000056
C$ERRO= 000060
C$ERSF= 000054
C$ERSO= 000057
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 00000!
C$EXIT= 000032
C$GETB= 000026
C$GETW= 000027

C$GMAN= 000043
C$GPHR= 000042
C$GPLO= 000030
C$GPRI= 000040
C$INIT= 000011
C$INLP= 000020
C$MANI= 000050
C$MEM = 000031
C$MSG = 000023
C$OPEN= 000034
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$QIO = 000377
C$RDBU= 000007
C$REFG= 000047
C$RESE= 000033
C$REVI= 000003
C$RFLA= 000021
C$RPT = 000025
C$SEFG= 000046
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
DATPTH  022703
DFPTBL  002232 G
DIAGMC= 000000
DROPED= 000200
DROPIT  004656
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
ENDP64  025735
ENDP96  026325
END2    007100
END4    014454
ERRCOD  002322
ERRFLG  002324
ERROR = 002000
ERRSVC  002674
ERRTBL  002676 G
ERR06   017502
ERR07   017553
ERR11   007172
ERR12   007246
ERR13   007331
ESPSTR  031407
ESPST1  031431
ESPST2  031453
ESPST3  031475

ESPST4  031517
ESPST5  031541
ESPST6  031563
ESPST7  031605
EVL   = 000004 G
E$END = 002100
E$LOAD= 000035
FAKE    004764
FFSET   013175
FLAG    002262
FLSMSG  013251
FLSMS1  013343
FLSSEL  012504
FRMLTH  012306
FSTMSG  020613
F$AU  = 000015
F$AUTO= 000020
F$BGN = 000040
F$CLEA= 000007
F$DU  = 000016
F$END = 000041
F$HARD= 000004
F$HW  = 000013
F$INIT= 000006
F$JMP = 000050
F$MOD = 000000
F$MSG = 000011
F$PROT= 000021
F$PWR = 000017
F$RPT = 000012
F$SEG = 000003
F$SOFT= 000005
F$SRV = 000010
F$SUB = 000002
F$SW  = 000014
F$TEST= 000001
GETADR  032316
GETBND  032476
GETDAV  032520
GETFLG  017214
GETMAN  032547
GETMAX  032727
GETPLA  032706
GETTIM  032621
GETVEC  032333
G$CNTO= 000200
G$DELM= 000372
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO  = 000000
G$OFFS= 000400

G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
G$RADB= 000000
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES = 000010
HAMFIR  026656
HAMRDY  011114
HAMRSW  010753
HAMSW1  011037
HOE   = 100000 G
IBE   = 010000 G
IDU   = 000040 G
IER   = 020000 G
INCDAT  017263
INCTBL  017617
INDEX   002330
INHINT  002244
INSTR   022216
INTERR  007134
INTER1  003410
INTHDL  007164
INTOO   031616
IOCTRL  004330
IODRV   004166
ISR   = 000100 G
iXE   = 004000 G
I$AU  = 000041
I$AUTO= 000041
I$CLN = 000041
I$DU  = 000041
I$HRD = 000041
I$INIT= 000041
I$MOD = 000041
I$MSG = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR = 000041
I$RPT = 000041
I$SEG = 000041
I$SETU= 000041
I$SFT = 000041
I$SRV = 000041
I$SUB = 000041
I$TST = 000041
J$JMP = 000167
LCTMSG  021016
LCTiBL  020532
```

```
LINCNT  002264          L$PRIO  002042 G        NONBUF  024137          REFLI2  013067          T$$AUT= 010002
LINPER  015154          L$PROT  002122 G        NONCHR  023752          REPLUP  014514          T$$CLE= 010005
LINSWI  012340          L$PRT   002112 G        NOSTOP  017334          RESET   017377          T$$HAR= 010034
LINSW1  012424          L$REPP  002062 G        NOTIM   006107          RESVEC  006204 G        T$$HW = 010001
LOE    = 040000 G       L$REV   002010 G        NRGT16  005612          SECMSG  020650          T$$INI= 010004
LOT    = 000010 G       L$SOFT  032356 G        NRGT17  005675          SFPTBL  002244 G        T$$PRO= 010000
LPBUF   002434          L$SPC   002056 G        NSTTBL  017624          SKIP3   024132          T$$SOF= 010035
LPCSR   002340          L$SPCP  002020 G        OFFLIN  015221          SPCCNT  026654          T$$SRV= 010033
LPDROP  003732          L$SPTP  002024 G        ONEFIL= 000001          STACHR  030476          T$$SUB= 010016
LPERR   004000          L$STA   002030 G        ONLIN1  015261          STATER= 000001          T$$SW = 010003
LPERR2  015142          L$SW    002244 G        ONLIN2  015362          STATUS  002474          T$$TES= 010031
LPINTR  002634          L$TEST  002114 G        ONLIN3  015460          STRCNT  002272          T1      006406 G
LPM64   015547          L$TIML  002014 G        OUTBUF  002742          SVCGBL= 000000          T1A     006422
LPM96   015630          L$UNIT  002012 G        OUTTIM  003470          SVCINS= 000001          T1B     007116
LPVEC   002400          L10001  002236          OVFLO = 000100          SVCSUB= 000001          T1C     006426
LSTCNT  002266          L10002  002240          O$APTS= 000000          SVCTAG= 000001          T10     023246 G
LUNIT   002300          L10003  002262          O$AU  = 000000          SVCTST= 000001          T11     024614 G
L$ACP   002110 G        L10004  006202          O$BGNR= 000000          S$LSYM= 010000          T12     026330 G
L$APT   002036 G        L10005  006404          O$BGNS= 000001          TABA64  015712          T13     026712 G
L$AUT   002070 G        L10006  007402          O$DU  = 000000          TABA96  016120          T14     027334 G
L$AUTO  002236 G        L10007  007162          O$ERRT= 000000          TABSTR  030500          T15     030212 G
L$CCP   002106 G        L10010  007170          O$GNSW= 000001          TAB64   027634          T16     030636 G
L$CLEA  006266 G        L10011  011450          O$POIN= 000001          TAB96   030014          T2      007404 G
L$CO    002032 G        L10012  013644          O$SETU= 000000          TICK    032272          T3      011452 G
L$DEPO  002011 G        L10013  016326          PAPCHK  012603          TIME    032270          T3MOV   012304
L$DESC  002172 G        L10014  017632          PAPRDY  010670          TIMOUT= 000002          T3SET   012302
L$DESP  002076 G        L10015  016644          PAPRSW  010542          TXERR   003443          T4      013646 G
L$DEVP  002060 G        L10016  017212          PAPSWI  003230          TXNOIN  003515          T5      016330 G
L$DISP  002132 G        L10017  021224          PAPSW1  010612          T$ARGC= 000001          T5.1    016420
L$DLY   002116 G        L10020  022356          PATCH   032754          T$CODE= 006052          T5.2    016646
L$DTP   002040 G        L10021  022734          PATTER  022732          T$ERRN= 000014          T6      017634 G
L$DTYP  002034 G        L10022  023244          PERIOD  002254          T$EXCP= 000000          T7      021226 G
L$DUT   002072 G        L10023  024612          PLOC    006200          T$FLAG= 000040          T8      022360 G
L$DVTY  002222 G        L10024  026326          PNT    = 001000 G       T$GMAN= 000000          T9      022736 G
L$EF    002052 G        L10025  026710          PRI    = 002000 G       T$HILI= 000377          UAM    = 000200 G
L$ENVI  002044 G        L10026  027332          PRINTR  002304          T$LAST= 000001          UNIT    002276
L$ETP   002102 G        L10027  030210          PRI00  = 000000 G       T$LOLI= 000001          USA     002256
L$EXP1  002046 G        L10030  030634          PRI01  = 000040 G       T$LSYM= 010000          UUT     002326
L$EXP4  002064 G        L10031  031614          PRI02  = 000100 G       T$LTNO= 000020          UUTEQ0  003546
L$EXP5  002066 G        L10033  032266          PRI03  = 000140 G       T$NEST= 000000          VFUCHL  022220
L$HARD  032276 G        L10034  032316          PRI04  = 000200 G       T$NS0  = 000010         VFUCMD  002332
L$HIME  002120 G        L10035  032440          PRI05  = 000240 G       T$NS1  = 000000         VFUDAT  022274
L$HPCP  002016 G        MANSPD  002252          PRI06  = 000300 G       T$NS2  = 000005         VFUERR  017232
L$HPTP  002022 G        MAXERR  002260          PRI07  = 000340 G       T$PTNU= 000000          VFULCT  020470
L$HW    002232 G        MGTINT  032440          PRTCHR  023214          T$SAVL= 177777          VFUOPT  002250
L$ICP   002104 G        MOVMSG  013440          PRTCTL  027074          T$SEGL= 177777          VFUSEL  003615
L$INIT  005020 G        MOVMS1  013535          PRTSPD  015511          T$SUBN= 000000          VFUSE1  003703
L$LADP  002026 G        MOVMS2  013633          PTABAD  002302          T$TAGL= 177777          VFUTBL  020571
L$LAST  033020 G        MRESET  005731          PTHPNT  027610          T$TAGN= 010036          WORK    002736 G
L$LOAD  002100 G        MULINE  030515          RDYERR  003206          T$TEMP= 000000          WORK1   002740
L$LUN   002074 G        NMLFLS  013103          READY   006010          T$TEST= 000020          X      = 000040
L$MREV  002050 G        NOLLCK  006045          REFLIN  012675          T$TSTM= 177777          X$ALWA= 000000
L$NAME  002000 G        NOINTR= 000003          REFLI1  012772          T$TSTS= 000001          X$FALS= 000040
```

CZLPLA.P11 01-OCT-79 09:25 SYMBOL TABLE

```
X$OFFS= 000400          $F$NO  = 000403          $LOCTA= 177777          $TSK1  = 050317          $$LOCN= 000000
X$TRUE= 000020          $F$OR  = 000320          $LSTIN= 000001          $TSK2  = 050320          $$REG = 177777
$BGNLE= 177777          $F$RTI= 000350          $LSTTA= 000001          $TSK3  = 050277          $$RETU= 000000
$ERFLG= 000400          $F$RTN= 000300          $NESTL= 177777          $TSK4  = 050300          $$RTN1= 000000
$F$AND= 000310          $F$SEL= 000140          $NSK0  = 000110         $TSK5  = 050161          $$RTN2= 000000
$F$BAD= 000401          $F$THE= 000330          $NSK1  = 000120         $TSK6  = 050162          $$SRC  = 000027
$F$BLA= 000170          $F$TRU= 000404          $NSK2  = 000110         $TSK7  = 050152          $$TGSV= 050127
$F$CAS= 00015C          $F$UNT= 000130          $NSK3  = 000120         $$ARGC= 000000           $$TGS1= 000003
$F$DEC= 000220          $F$WHI= 000120          $NSK4  = 000110         $$BYTE= 000403           $$TGS2= 000000
$F$DO  = 000340         $F$YES= 000402          $NSK5  = 000110         $$CASE= 000404           $$TO   = 000000
$F$FAL= 000405          $IFLEV= 177777          $SAVLE= 177777          $$DST  = 000037          $$$TAG= 050000
$F$GOO= 000400          $ISK0  = 000001         $SSK0  = 050320         $$ELOC= 000402           .      = 033020
$F$IF  = 000110         $ISK1  = 000001         $TAGLE= 177777          $$ERFL= 000000
$F$INC= 000210          $ISK2  = 000001         $TAGNU= 050324          $$FLAG= 000001
$F$LOO= 000200          $ISK3  = 000001         $TEMP  = 050323         $$FROM= 000000
$F$NAM= 000160          $ISK4  = 000001         $TSK0  = 050323         $$LOC  = 032256

. ABS.  033020    000


 ERRORS DETECTED:  0

 CZLPLA,CZLPLA.SEQ/DOC=SVC/ML,SPMAC/ML,CZLPLA
 RUN-TIME: 319 295 2 SECONDS
 RUN-TIME RATIO: 1091/617=1.7
 CORE USED:  24K  (47 PAGES)

 DOCUMENT PAGES:  122
```