```
000001                                  TITLE   CRMS3,'REV G',CARDREADER  / PUNCH TV (SAF)
      TITLE CRMS3, 'REV G', CARDREADER / PUNCH T&V (SAF)
DEVELOPMENT OF THIS PROGRAM HAS BEEN VIA MODULES WHICH ARE DEDICATED TO
PERFORM SPECIFIC TASKS. SOME OF THE MODULES ARE TESTS IN THEMSELVES
AND OTHERS ARE CALLED ON TO PERFORM A SPECIFIC FUNCTION AND THEN RETURN
TO THE CALLER.
THE FIRST TYPE OF MODULE, THE TESTS, HAVE THE FORM ZVKJLX AND ZVKJFS.
THE X SIGNIFIES THE LETTER OF THE TEST AND IS ALSO THE RESPONSE TO THE
QUESTION "NEXT ?:". THESE TESTS ARE ZVKJLB, ZVKJLC, ZVKJLF,
ZVKJLI, ZVKJLK, ZVKJLL, ZVKJLM, ZVKJLN, ZVKJLP, ZVKJLR AND ZVKJFS.
THE SECOND TYPE OF MODULE, IS THE SPECIFIC FUNCTION OR LIBRARY
MODULE. THESE HAVE THE FORM ZVKJYX AND ZVKXX, WHERE THE Y IS ANY
LETTER EXCEPT L.
ERROR LABELS
------------
ERROR LABELS IDENTIFY THE TEST IN WHICH AN ERROR OCCURS, THE PART
OF THE TEST AND THE SPECIFIC ERROR. THE MAJOR LABEL IDENTIFIES THE
TEST AND THESE ARE B, C, F, I, K, L, M, N, P, R AND FS. IF THE TEST
IS COMPOSED OF SEVERAL PARTS THEN THE NUMBER FOLLOWING THE TEST
LETTER WILL IDENTIFY THE PART.
THE MINOR LABEL, FLAGS THE SPECIFIC ERROR AND IS TWO CHARACTERS LONG.
THESE IN GENERAL ARE SET BY THE FUNCTION MODULES, WHILE THE MAJOR
LABELS ARE SET BY THE TEST MODULES.
IN GENERAL MAJOR LABELS THAT BEGIN WITH B, I OR L INDICATE PROBLEMS
WITH THE MDC OR ADAPTER, WHILE THOSE THAT BEGIN WITH C, F, K, M, N, P
OR R POINT TO THE DEVICE. SEE THE LISTING FOR SPECIFIC LABELS AND
THERE MEANINGS.
                    DESCRIPTION OF MODULES
                    ----------------------
NAME     CALLED BY                      DESCRIPTION
----     ------ --                      -----------
CARD     **                             CONTAINS MOST OF THE DECLERERTIONS
                                        AND IS THE TEST WHICH ASKS THE QUESTION
                                        "NEXT ?:".
ZVKJ1B   **                             LIBRARY OF MACROS AND EXTERNALS
ZVKGB    GET BREAK                      CHECKS FOR UNSOLICITAED INPUT
                                        FROM THE CONSOLE.
ZVKGC    GET CHAR                       INPUT 2 CHARACTERS VIA THE CONSOLE
                                        ONE VIA THE CONSOLE
ZVKJCA   COMPARE ASCII                  ROUTINE TO OUTPUT ASCII WAS
                                        AND SHOULD BE VALUES AND COLUMN
                                        NUMBER. CALLED BY TEST N.
ZVKJCC   COMPARE CHECKSUMS              CALLED BY VERIFY TO CHECK
                                        STORED CHECKSUMS AGAINST READ
                                        CHECKSUMS AND TO OUTPUT ANY
                                        DIFFERENCES.
ZVKJCS   CHECK STATE                    CHECKS STATE OF DEVICE AND PROMPTS
                                        USER TO START OR STOP DEVICE.
ZVKJET   CALCULATE ELAPSED TIME         WILL PRINT THE NUMBER OF CARDS READ
                                        OR PUNCHED AND THE RATE.
ZVKJFB   FILL BUFFER                    INITIALIZES A BUFFER TO A VALUE
ZVKJGC   GENERATE CHECKSUMS             CONVERTS A READ OR PUNCHED CARD
                                        INTO 8 CHECCKSUM WORDS
ZVKJIL   IOLD                           DATA TRANSFER PROCEDURE
ZVKJIO   INOUT                          READ OR WRITE PROCEDURE
ZVKJMC   MICRO                          ENTERS TEST MODE AND ISSUES MICRO OPS
ZVKJPD   PUNCH A DECK                   GENERATES AND PUNCHES SPECIFIC PATTERNS
                                        CALLED BY LETTER P.
ZVKJPN   PUT NUMBER                     OUTPUTS A SPACE SUPRESSED DECIMAL
                                        NUMBER.
ZVKJRE   REPORT ERROR                   ISSUES TE ERROR REPORT
ZVKJSE   CHECK STATUS FOR ERRORS        DETERMINES AND DISPLAYS RANGE
                                        AND STATUS ERRORS.
ZVKJTC   GENERATE TEST CHECKSUMS        USED WHEN "HAVE A 100 CARD TEST DECK ?:"
                                        IS ANSWERED YES, TO GENERATE THE
                                        CHECKSUMS FOR THAT DECK
ZVKJTD   TIME DELAY                     CALLED  BY VERIFY, TO DELAY
                                        BETWEEN CARD PICKS
ZVKJWR   WAIT FOR RETURN                WAITS FOR A C/R, BEFORE RETURNING
ZVKJLB   B                              ELECTRONICS TEST, CALLS TEST I AND L
ZVKJLC   C                              DEVICE CONTROLS
ZVKJLF   F                              PUNCH AT 400 CPM
ZVKJLI   I                              INTERRUPT TEST
ZVKJLK   K                              READ AND PUNCH TEST
ZVKJLM   M                              PUNCH ASCII DECK
ZVKJLM   N                              READ ASCII DECK
ZVKJLR   R OR V                         READ OR VERIFY ANY DECK
ZVKJFS   S                              RESTART OR START OF PROGRAM
ZV$BK    **                             CONTAINS THE CTRL LINKS AND IDEFS
                    CROSS REFERENCE FOR CRMS3
                    ------------------------
                                  CALLING ROUTINE
                  Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z
                  Z V V V V V V V V V V V V V V V V V V V V V V V V Z
                C K K K K K K K K K K K K K K K K K K K K K K K K V
                A J J J J J J J J J J J J J J J J J J J J J J J J $
                R C C C E G I I M P S T W L L L L L L L L P L S F B
                D A C S T C L O R C D E C R B C F I K L M N P R S K
                ---------------------------------------------------
     CARD       |   X X     X X X         X   X   X   X     X X X X
   ZVKGB        |X X X X         X             X X   X       X X   X
   ZVKGC        |X                          X   X X   X     X X X X
  ZVKJCA        |                                       X       X   X
  ZVKJCC        |                                               X   X
  ZVKJCS        |         X         X       X X X X X X X X X X X X
  ZVKJET        |                             X   X       X       X
  ZVKJFB        |               X   X         X   X   X X X       X
  ZVKJFC        |                                           X X X X
  ZVKJGC        |               X   X         X   X           X   X
C ZVKJIL        |         X     X             X   X   X X X       X
A ZVKJIO        |X   X X   X X     X   X     X X X X X X X X X X X
L ZVKJIR        |                                 X               X
L ZVKJMC        |                                     X           X
E ZVKJPD        |   X                                         X   X
D ZVKJPN        |   X X   X           X         X       X       X X
  ZVKJRE        |     X         X               X   X X X X X   X X
M ZVKJSE        |               X             X X X X X X X X   X
O ZVKJTC        |                                 X               X
D ZVKJTD        |     X                                           X
U ZVKJWR        |               X         X X   X   X X X X     X
L ZVKJLB        |X                                                X
E ZVKJLC        |X                                                X
  ZVKJLF        |X                                                X
```

```
         ZVKJLI |X                                                   X
         ZVKJLK |X                                                   X
         ZVKJLL |X                                                   X
         ZVKJLM |X                                                   X
         ZVKJLL |X                                                   X
         ZVKJLN |X                                                   X
         ZVKJLP |X                                                   X
         ZVKJLR |X                       X   X         X X           X
         ZVKJFS |X                                                   X
&&L
```

```
E    CARD.LIST                    05/19/78  1422.4R W 05/19/78  1422.4     11444220000
000001                                 TITLE    CARD,'REV 00'
000002                          *
000003                          * DESCRIPTION
000004                          * -----------
000005                          *
000006                          *   THIS PROGRAM TESTS AND VERIFIES PROPER OPERATION OF ALL LEVEL-6 CARD
000007                          *   EQUIPMENT. THE PROGRAM WILL RUN ON THE FOLLOWING SUBSYSTEMS:
000008                          *
000009                          *        CRM9101 DEVICE-PAC FOR CARD READER
000010                          *        CRU9101 300 CPM PUNCHED CARD READER
000011                          *        CRU9102 300 CPM PUNCHED AND MARKED CARD READER
000012                          *        CRU9103 500 CPM PUNCHED CARD READER
000013                          *        CRU9104 500 CPM PUNCHED AND MARKED CARD READER
000014                          *        CRU9108 300 CPM READER
000015                          *        CRU9109 300 CPM READER WITH IBM MS
000016                          *        CRU9110 300 CPM READER WITH HIS MS
000017                          *        CRU9111 500 CPM READER
000018                          *        CRU9112 500 CPM READER WITH IBM MS
000019                          *        CRU9113 500CPM READER WITH HIS MS
000020                          *        CRM9103 DEVICE PACC FOR READER/PUNCH OR PUNCH
000021                          *        PCU9101 Z14-1A CARD PUNCH
000022                          *        CCU9101 Z14-2A CARD READER/PUNCH
000023                          *
000024                          *        CRU9101 51 COLUMN CARD OPTION (NOT AVAILABLE THIS REL)
000025                          *
000026                          * REVISION HISTORY
000027                          * ----------------
000028                          *
000029                          *        A   NOV 1975   ORIGINAL RELEASE (READER ONLY)
000030                          *        B   JAN 1976
000031                          *        C   APR 1976
000032                          *        D   DEC 1976   CRM51
000033                          *        E   APR 1977   CRMX1 (SAF AND LAP)
000034                          *        F   JUL 1977   CRMX2 (SAF AND LAP)
000035                          *        G   APR 1978   CRM53 READER PUNCH
000036                          *
000037                          *
000038                          *
000039                          *              NAME      CRM53 - SAF
000040                          *              DOCUMENT  60129821-007
000041                          *
000042                          *
000043                          *
000044                          *   THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS CONFIDENTAL
000045                          *   AND PROPRIETARY TO AND THE EXCLUSIVE PROPERTY OF HONEYWELL INFORMA-
000046                          *   TION SYSTEMS INC. IT IS MADE AVAILABLE ONLY TO HONEYWELL AUTHORIZED
000047                          *   RECIPIENTS FOR THEIR USE SOLELY IN THE MAINTENANCE AND OPERATION OF
000048                          *   HONEYWELL PRODUCTS. THIS DOCUMENT AND INFORMATION MUST BE MAINTAINED
000049                          *   IN THE STRICTEST CONFIDENCE; IT MUST NOT BE REPRODUCE IN WHOLE OR
000050                          *   IN PART; AND IT SHALL NOT BE DISCLOSED TO ANY OTHER PARTY WITHOUT
000051                          *   THE PRIOR WRITTEN CONSENT OF HONEYWELL.
000052                          *
000053                          * DISTRIBUTION
000054                          * ------------
000055                          *
000056                          *   THIS PROGRAM WAS PREPARED USING THE NATIVE SYSTEM AND THE ZV$LIB
000057                          *   LIBRARY. INPUT TO THE PROGRAM DISTRIBUTION FACILITY CONSISTED OF;
000058                          *
000059                          *        1 - LINKED IMAGE ON DISKETTE
000060                          *        2 - HARD COPY OF THE LISTING FILE FROM THE
000061                          *            ASSEMBLER AND LINKER
000062                          *
000063                          *   THE EXECUTABLE IMAGE MAY BE REPRODUCED FOR SHIPMENT AS A CARD DECK,
000064                          *   A PAPER TAPE, OR A DISKETTE FILE MEMBER. WITHIN THE FILE IT HAS THE
000065                          *   MEMBER NAME "CRM53".
000066                          *   THE LISTING MAY BE REPRODUCED AS A MICROFICHE. A FORMAL SET OF DIS-
000067                          *   TRIBUTION RULES ARE FOUND IN THE DOCUMENT 60129314 ENTITLED
000068                          *   "T & D INDEX".
```

```
000069                          /
000070                          *
000071                          * RESTRICTIONS
000072                          * ------------
000073                          *
000074                          *   IT REQUIRES A MINIMUM OF 16K MEMORY TO RUN AND IS NON-RELOCATABLE.
000075                          *
000076                          * OPERATION
000077                          * ---------
000078                          *
000079                          *   LOAD THE PROGRAM AND START (OR RESTART) AT LOCATION 0100 HEX.
000080                          *   IF A CONSOLE IS PRESENT, VERIFY CORRECT PROGRAM IDENTIFICATION
000081                          *   FROM THE LISTING OF THE I/O EQUIPMENT BY CHANNEL NUMBER AND
000082                          *   ID-CODE. THIS LISTING WILL BE FOLLOWED BY THE LOWEST AVAILABLE
000083                          *   AND HIGHEST AVAILABLE MEMORY LOCATION.
000084                          *
000085                          *   THE FOLLOWING IS A TYPICAL RESULT OF LOADING THE PROGRAM
000086                          *   AND STARTING TO RUN.
000087                          *
000088                          *        CRMS3, REV G, APR 20, 1978
000089                          *        ZV$LIB  REV. 7
000090                          *        ZV$AF = 1
000091                          *        WDT
000092                          *        CHAN  DEVC  ID
000093                          *        0400  DSKT  2010
000094                          *        0480  DSKT  2010
000095                          *        0500  CONS  2019
000096                          *        1300  CDR   2008
000097                          *        1380  LPT   2006
000098                          *        1400  PNCH  208A
000099                          *        MEMORY LOW  0000XXXX
000100                          *        MEMORY HIGH 00007FFF 32K
000101                          *
000102                          *
000103                          *   THE PROGRAM WILL ASK:
000104                          *
000105                          *        CHANNEL NUMBER ?; 1400 C/R
000106                          *
000107                          *   THE RESPONSE SHOULD BE THE 4 DIGIT HEX CHANNEL NUMBER ASSIGNED TO
000108                          *   THE CARD DEVICE TO BE TESTED. DEFAULT VALUE IS 1300
000109                          *   AND ON SUBSEQUENT RESTARTS A C/R ALONE WILL RETAIN THE
000110                          *   PREVIOUS VALUE.
000111                          *
000112                          *   THE FIRST EXECUTION OF THE PROGRAM WILL THEN ASK:
000113                          *
000114                          *        POWER FREQUENCY ?; 60 C/R
000115                          *
000116                          *   RESPOND WITH THE POWER FREQUENCY IN HERTZ, USUALLY 60 IN THE USA AND
000117                          *   50 ELSEWHERE. DEFAULT VALUE IS 60 HZ. THIS QUESTION IS NOT ASK ON
000118                          *   6/40 SYSTEMS.
000119                          *
000120                          *   IF THE DEVICE BEING TESTED IS A CARD READER THEN THE FOLLOWING
000121                          *   QUESTIONS WILL BE ASKED:
000122                          *
000123                          *        MARK SENSE ?; Y   (C/R)
000124                          *        40 COLUMNS ?; N   (C/R)
000125                          *
000126                          *   ANSWER WITH ONLY ONE CHARACTER, EITHER (Y) OR (N). RESPONDING WITH
000127                          *   N TO MARK SENSE ?; WILL NOT ASK THE 40 COLUMN QUESTION SINCE ALL
000128                          *   PUNCHED CARDS ARE 80 COLUMN. ANSWERING WITH A Y CAUSES THE 40 COLUMN
000129                          *   QUESTION TO BE ASKED.
000130                          *   ANSWERING Y TO THE 40 COLUMN QUESTION MEANS THE READER IS
000131                          *   CAPABLE OF READING IBM MARKED CARDS AND THAT THE READER
000132                          *   IS SET TO 40 COLUMNS. ANSWERING N REQUIRES THE IBM READER
000133                          *   TO BE SET TO 80 COLUMNS OR THAT THE READER IS A HIS READER, IN WHICH
000134                          *   CASE THE 40/80 SWITCH IS DISABLED AND IS PRESET TO 80 COLUMNS.
000135                          *
000136                          *   THE QUESTION "NEXT ?" WILL BE ASKED WHICH REQUIRES A LETTER AND A C/R
000137                          *   RESPONSE. LETTER A WILL RUN ALL ELECTRONICS AND DEVICE TESTS AND IS
000138                          *   THE RESPONSE RECOMMENDED. THE OPTIONS WILL BE EXPLAINED BELOW.
000139                          *   A C/R ALONE WILL RETAIN THE LAST RESPONSE TO "NEXT ?". DEFAULT
000140                          *   IS "A" FOR ALL.
000141                          *
000142                          * CONSOLE SELECTION
000143                          * -----------------
000144                          *
000145                          *   THE CONSOLE SEARCH RULES ARE: FIND THE CONSOLE WITH THE LOWEST
000146                          *   CHANNEL NUMBER CONNECTED THRU A MDC. IF THERE IS NO CONSOLE ON
000147                          *   AN MDC, THEN SEARCH FOR A TERMINAL WITH THE HIGHEST CHANNEL
000148                          *   NUMBER ASSIGNED TO AN ACLA ADAPTER ON A MLC CONTROLLER.
000149                          *   IF NO ASYNC ADAPTER IS FOUND, THEN GO TO THE FULL CONTROL PANEL.
000150                          *
000151                          *   THERE ARE THREE CONSOLE CHANNEL OPTIONS DETERMINED BY THE VALUE
000152                          *   OF LOCATION ZV$ITY.
000153                          *   IF ZV$TTY EQUALS (0000), SEARCH FOR A CONSOLE.
000154                          *   IF ZV$TTY EQUALS (FFFF), ASSUME THERE IS NO CONSOLE.
000155                          *   IF ZV$TTY EQUALS NEITHER (0000), NOR (FFFF), THEN IT IS THE
000156                          *   CONSOLE CHANNEL NUMBER. NOTE: DEFAULT IS TO SEARCH FOR A CONSOLE.
000157                          *
000158                          *   ALL CONSOLE I/O IS EVEN PARITY. IF CONSOLE IS ON MLC, IT MUST BE
000159                          *   ASYNC AND THE BAUD RATE SET AT 1200 TO MATCH THE PROGRAM SUPPLIED
000160                          *   RATE. IF IT IS NECESSARY TO CHANGE THE PROGRAM BAUD RATE, THEN THE
000161                          *   NEW BAUD RATE CODE SHOULD BE PUT INTO LOCATION "ZV$BUD" IN HEX.
000162                          *   THE TERMINAL BAUD RATE MUST BE SET TO MATCH THIS NEW BAUD RATE.
000163                          *   THE CORRECT HEX VALUE MAY BE OBTAINED FROM THE FOLLOWING TABLE.
000164                          *
000165                          *        ACLA I.D.      (2118) (2110)       (2108)
000166                          *        BAUD-RATE
000167                          *             50              0                1
000168                          *             75              1                2
000169                          *            110              2                3
000170                          *            134              3                4
000171                          *            150              4                5
000172                          *            200              5               ---
000173                          *            300              6                6
000174                          *            600              7                7
000175                          *            900             ---               8
000176                          *           1050              8               ---
000177                          *           1200              9                9
000178                          *           1800             10 (A)           10 (A)
000179                          *           2000             11 (B)          ----
000180                          *           2400             12 (C)           11 (B)
000181                          *           3600            ----              12 (C)
```

```
000182                          *          4800          13 (D)          13 (D)
000183                          *          7200          ----            14 (E)
000184                          *          9600          14 (E)          15 (F)
000185                          *         19200          15 (F)          ----
000186                          *
000187                          *    TO MAKE ANY OF THE ABOVE CHANGES, LOAD AND HALT THE PROGRAM
000188                          *    BEFORE EXECUTION. INSERT CHANGES THEN EXECUTE. MEMORY LOCATIONS
000189                          *    OF "ZV$TTY" AND "ZV$BUD" MAY BE FOUND IN THE MAP AT END OF LISTING.
000190                          *
000191                          *    CONSULT LEVEL-6 T&V MANJAL "AW94" FOR DETAILS ON HOW TO LOAD THE
000192                          *    TESTS.
```

```
000193                          /
000194                          *
000195                          * OPTIONS
000196                          * -------
000197                          *
000198                          *   RESPONSES TO "NEXT ?":
000199                          *
000200                          *        A - ALL TESTS, ELECTRONICS AND DEVICE  (DEVICE NEEDED)
000201                          *        B - ALL ELECTRONIC TESTS ONLY
000202                          *        C - DEVICE INTERLOCKS AND CONTROLS     (DEVICE NEEDED)
000203                          *        D - ALL DEVICE RELATED TESTS           (DEVICE NEEDED)
000204                          *        F - PUNCH AT 400 CPM                   (DEVICE NEEDED)
000205                          *        I - INTERRUPT TEST
000206                          *        K - READ/PUNCH SAME CARD               (DEVICE NEEDED)
000207                          *        L - TEST MODE + DATA LOOPBACK
000208                          *        M - PUNCH ASCII 3 CARD TEST DECK       (DEVICE NEEDED)
000209                          *        N - READ ASCII 3 CARD (4 MS) TEST DECK (DEVICE NEEDED)
000210                          *        P - PUNCH RANDOM OR INPUTTED DATA      (DEVICE NEEDED)
000211                          *        R - READ ANY DECK                      (DEVICE NEEDED)
000212                          *        S - START/RESTART PROGRAM
000213                          *        V - VERIFY LAST PUNCHED OR READ DECK   (DEVICE NEEDED)
000214                          *
000215                          *   A TEST SEQUENCE MAY BE INTERRUPTED TO TERMINATE DATA MISCOMPARES
000216                          *   AS IN VERIFY MODE OR WHENEVER THE STATEMENT "TYPE 'CARRIAGE RETURN',
000217                          *   WHEN READY" APPEARS, BY STRIKING THE CONSOLE BREAK KEY. THIS WILL
000218                          *   RETURN THE PROGRAM TO THE "NEXT ?:" QUESTION, AT WHICH POINT ANY
000219                          *   OPTION CAN BE SELECTED.
000220                          *
000221                          * ERROR REPORTING
000222                          *----------------
000223                          *
000224                          *   DETECTED ERRORS ARE REPORTED AS FOLLOWS:
000225                          *
000226                          *   ERR ABCD @ LLLL, MESSAGE ABOUT ERROR
000227                          *
000228                          * WHERE:
000229                          *   AB = MAJOR LABEL, REFERS TO SUB-TEST BEING PERFORMED
000230                          *        A = TEST TYPE
000231                          *        B = PART OF TEST
000232                          *   CD = MINOR LABEL, INDICATES SPECIFICALLY THE ERROR DETECTED
000233                          * LLLL = HEX LOCATION OF DETECTED ERROR
000234                          *
000235                          * IF THE ERROR DETECTED INVOLVES STATUS THE REPORT WILL BE AS FOLLOWS:
000236                          *
000237                          *   ERR ABSE @ LLLL, STATUS WAS XXXX, STATUS SHOULD BE YYYY
000238                          *   MESSAGE ABOUT ERROR
000239                          *
000240                          * WHERE:
000241                          *   AB = MAJOR LABEL, REFERS TO SUB-TEST BEING PERFORMED
000242                          *   SE = MINOR LABEL, STATUS ERROR
000243                          * LLLL = HEX LOCATION OF THE STATUS COMPARE
000244                          * XXXX = HEX STATUS WORD WAS
000245                          * YYYY = HEX STATUS WORD SHOULD BE
000246                          *
000247                          *
000248                          *   WHEN A STATUS ERROR HALT OCCURS ON A NO CONSOLE SYSTEM,
000249                          *   D4 WILL CONTAIN STATUS WAS AND D5, STATUS SHOULD BE
000250                          *
000251                          *   THE FOLLOWING IS A SUMMARY OF ALL MAJOR AND MINOR LABELS AND
000252                          *   THEIR ERROR INDICATION:
000253                          *
000254                          *
000255                          *   MAJOR LABEL (LABEL 1)
000256                          *
000257                          *      B1 -- INITIALIZATION AND CONFIGURATION OF 2008
000258                          *      B2 -- INITIALIZATION AND TASK SETTING OF 208A + 2088
000259                          *      C1 -- CAN'T PUT DEVICE ON-LINE
000260                          *      C2 -- STOP SWITCH DEFECTIVE
000261                          *      C3 -- OUTPUT STACKER SWITCH DEFECTIVE ON PUNCH
000262                          *      C4 -- LID SWITCH DEFECTIVE ON PUNCH
000263                          *      C5 -- READ DOOR INTERLOCK DETECTIVE ON PUNCH
000264                          *      F1 -- INITILIZATION OF PUNCH FOR 400 CPM PUNCH
000265                          *      F2 -- ACTUAL PUNCHING OF 400 CPM DECK
000266                          *      I1 -- INTERRUPT TEST
000267                          *      K1 -- INITIALIZATION OF READER/PUNCH FOR READ/PUNCH
000268                          *      K2 -- ACTUAL READING AND PUNCHING
000269                          *      L1 -- BINARY LOOPBACK FOR 2008
000270                          *      L2 -- ASCII LOOPBACK FOR 2008
000271                          *      L3 -- BINARY LOOPBACK FOR PUNCH
000272                          *      L4 -- TEST READER PROMS FOR PUNCH
000273                          *      L5 -- TEST PUNCH PROMS FOR PUNCH
000274                          *      L6 -- TEST JAM TIME OUT ONE-SHOT FOR PUNCH
000275                          *      M1 -- PUNCH ASCII CARD
000276                          *      M2 -- PUNCH ILLEGAL ASCII
000277                          *      M3 -- PUNCH BLANK CARD
000278                          *      N1 -- READ FIRST ASCII CARD
000279                          *      N2 -- READ 2ND MARK SENSE ASCII CARD
000280                          *      N3 -- READ ILLEGAL ASCII CARD
000281                          *      N4 -- READ BLANK CARD
000282                          *      P1 -- INITIALIZATION FOR RANDOM PUNCH
000283                          *      P2 -- COPY A CARD FOR PUNCHING A DECK
000284                          *      P3 -- COPY A CARD USING A DIFFERENT READER
000285                          *      P4 -- INPUT DATA FROM A KEYBOARD, FOR PUNCHING
000286                          *      P5 -- PUNCH THE SPECIFIED DECK
000287                          *      R1 -- INITIALIZE READER FOR READ OR VERIFY MODE
000288                          *      R2 -- READ DECK OF CARDS, GENERATE OR COMPARE CHECKSUMS
000289                          *      S1 -- FRESH START, FIRST I/O
000290                          *
000291                          *   MINOR LABEL (LABEL 2)
000292                          *
000293                          *      AM -- ASCII NOT SET ON INITIALIZE                          (2008)
000294                          *      BM -- BINARY NOT SET ON CONFIGURATION                      (2008)
000295                          *      CA -- ASCII MISCOMPARE, CONSOLELESS SYSTEMS ONLY   (2008 + 208A)
000296                          *      CC -- CHECKSUM MISCOMPARE, COLUMN ERROR            (2008 + 208A)
000297                          *      CD -- CHECKSUM MISCOMPARE, DROPPED BITS            (2008 + 208A)
000298                          *      CG -- CHECKSUM MISCOMPARE, PICKED BITS             (2008 + 208A)
000299                          *      CI -- CLOCK INTERRUPTED, CAN'T LEV BACK                    (ALL)
000300                          *      CL -- CHECK LIGHT ON                                       (ALL)
000301                          *      CP -- CP RUPT DEVICE                                       (ALL)
000302                          *      CR -- CHECKSUM MISCOMPARE, ROW ERROR               (2008 + 208A)
000303                          *      CS -- STATE OF DEVICE WRONG                                (ALL)
000304                          *      CW -- COUNTER WRONG IN TEST MODE                           (2008)
000305                          *      DL -- BINARY DATA LOOPBACK ERROR                   (208A + 2088)
```

```
000306                          *       DS -- DROPPED STROBE                                    (2008 + 208A)
000307                          *       FU -- FLAG NOT ON IN ZHIAFB                                (ALL)
000308                          *       IA -- INTERRUPT REG WRONG - OUTPUT AAAA                    (ALL)
000309                          *       IC -- WRONG INVALID CONVERSION                             (ALL)
000310                          *       ID -- WRONG DEVICE ON SPECIFIED CHANNEL                    (ALL)
000311                          *       IN -- IO INSTRUCTION NAK'D                                 (ALL)
000312                          *       IO -- IOLD NAK'D                                           (ALL)
000313                          *       IR -- CAN'T MOVE CP LEVEL FROM 0                           (ALL)
000314                          *       I5 -- INTERRUPT REG WRONG - OUTPUT 5555                    (ALL)
000315                          *       MA -- MEMORY ADDRESS WRONG AFTER IOLD                      (ALL)
000316                          *       MW -- XFERED TO MANY WORDS WITH AN IOLD                    (ALL)
000317                          *       NT -- JAM ONE SHOT DIDN'T TIME OUT IN 1/2 SEC              (ALL)
000318                          *       OS -- ENABLE EJECT ON ERROR FAILED                        (208A)
000319                          *       PC -- PROGRAM EJECT FAILED                                (208A)
000320                          *       RE -- RANGE REGISTER WRONG AFTER IOLD                      (ALL)
000321                          *       R1 -- DEVICE DIDN'T RUPT - SHOULD HAVE                     (ALL)
000322                          *       R2 -- DEVICE RUPTED - SHOULDN'T HAVE                       (ALL)
000323                          *       R3 -- PENDING RUPT WHEN CP = 63                            (ALL)
000324                          *       SE -- STATUS ERROR                                         (ALL)
000325                          *       SR -- ISA SREG WRONG                                       (ALL)
000326                          *       SW -- ERROR IN MARK, 40 OR 51 COLUMN STATUS BITS          (2008)
000327                          *       TC -- TASK NOT CLEARED ON INITIALIZE            (2088 + 208A)
000328                          *       TQ -- JAM ONE SHOT TIMED OUT LESS THEN 150 MS   (2088 + 208A)
000329                          *       TS -- TASK NOT CLEARED BY OUTPUT TASK           (2088 + 208A)
000330                          *       TW -- CANNOT SET TASK WORD                      (2088 + 208A)
000331                          *       WC -- WRONG VALID CONVERSION                               (ALL)
000332                          *       WF -- WRONG FLAG ON IN ZHIAFB                              (ALL)
000333                          *       WD -- WRONG DEVICE THAT RUPTED                             (ALL)
000334                          *
000335                          *   THIS PROGRAM CAN BE OPERATED WITHOUT A CONSOLE PRESENT.
000336                          *   UNDER THESE CIRCUMSTANCES ERROR DATA IS LIMITED TO THE MAJOR
000337                          *   AND MINOR ERROR LABELS DISPLAYED IN HARDWARE REGISTERS "R1"
000338                          *   AND "R2" RESPECTIVELY, AND "LLLL" DISPLAYED IN "B2". ALL
000339                          *   PARAMETER ENTRIES TO THE PROGRAM MUST BE MADE VIA REGISTER "R1".
000340                          *   THE PROCESS IS FURTHER EXPLAINED IN MANUAL "AW94" ENTITLED
000341                          *   "LEVEL-6 SYSTEM CHECKOUT AND T&V MANUAL".
```

```
000342                          /
000343                          *
000344                          * STATUS WORDS
000345                          * ------------
000346                          *
000347                          *
000348                          *    STATUS WORD FOR 2008 (READER)
000349                          *
000350                          *     ---- ---- ---- ----
000351                          *    I    I    I    I    I
000352                          *     ---- ---- ---- ----
000353                          *     !!!! !!!! !    !!!!-- UNCORRECTED MEMORY ERROR
000354                          *     !!!! !!!! !    !!!--- BUS PARITY ERROR
000355                          *     !!!! !!!! !    !!---- NONEXISTENT RESOURCE
000356                          *     !!!! !!!! !    !----- CORRECTED MEMORY ERROR
000357                          *     !!!! !!!! !---------- ILLEGAL ASCII CODE
000358                          *     !!!! !!!!------------ READ CHECK
000359                          *     !!!! !!!------------- EXTERNAL CLOCK TRACK
000360                          *     !!!! !!-------------- 51 COLUMN MODE
000361                          *     !!!! !--------------- 40 COLUMN MODE
000362                          *     !!!!------------------ MARK SENSE MODE
000363                          *     !!!------------------- DATA SERVICE RATE ERROR
000364                          *     !!-------------------- NON DATA SERVICE REQUEST
000365                          *     !--------------------- DEVICE READY
000366                          *
000367                          *
000368                          *    STATUS WORD FOR 2088 / 208A (PUNCH - READER/PUNCH)
000369                          *
000370                          *     ---- ---- ---- ----
000371                          *    I    I    I    I    I
000372                          *     ---- ---- ---- ----
000373                          *     !!!! !!!!      !!!!-- UNCORRECTED MEMORY ERROR
000374                          *     !!!! !!!!      !!!--- BUS PARITY ERROR
000375                          *     !!!! !!!!      !!---- NONEXISTENT RESOURCE
000376                          *     !!!! !!!!      !----- CORRECTED MEMORY ERROR
000377                          *     !!!! !!!!------------ HOPPER EMPTY
000378                          *     !!!! !!!------------- INTERVENTION ACQUIRED (JAM)
000379                          *     !!!! !!-------------- READ STATION OR PUNCH ECHO CHECK-DEVICE
000380                          *     !!!! !--------------- READ REGISTRATION OR PUNCH ECHO CHECK
000381                          *     !!!!------------------ INVALID HOLLERITH CARD CODE
000382                          *     !!!------------------- DATA SERVICE RATE ERROR
000383                          *     !!-------------------- NON DATA SERVICE REQUEST
000384                          *     !--------------------- DEVICE READY
000385                          *
000386                          *
000387                          *
000388                          *
000389                          * NOTE 1 - FOR ALL CARD EQUIPMENT
000390                          * ------
000391                          *   IT IS SUGGEST A MINIMUM DECK OF 100 CARDS BE TESTED WHENEVER THE
000392                          *   "TOTAL" OR "DESIRED TOTAL" QUESTION IS ASKED.
000393                          *   STRIKING THE BREAK KEY WHENEVER "TYPE 'CARRIAGE RETURN' WHEN READY"
000394                          *   IS ASKED, WILL RETURN TO THE "NEXT ?:" QUESTION.
000395                          *   AT THE BEGINNING OF EACH DISCRIPTION OF EACH TEST R, R/P OR P MAY
000396                          *   APPEAR. THESE ARE THE DEVICES FOR WHICH THE TEST APPLIES, R FOR
000397                          *   READER, R/P FOR READER AND PUNCH AND P FOR PUNCH.
000398                          *   COLUMNS REPORTED IN ERROR FOR MARK SENSE MODE ARE FOR HIS CARDS. IF
000399                          *   IBM CARDS ARE USED, THEN THE COLUMNS REPORTED ARE 1 GREATER .
000400                          *
000401                          *
000402                          * NOTE 2 - FOR READER/PUNCH OR PUNCH
000403                          * ------
000404                          *   UPON STATUS ERRORS WHILE PUNCHING, THE ERROR CARDS
000405                          *   WILL BE OFFSET STACKED AND SHOULD BE VERIFIED BY THE OPERATOR. THE
000406                          *   RATE IN CPM WILL ALSO BE SUPPRESSED.
000407                          *
000408                          *   DURING VERIFY MODE, CARDS WHICH CAUSE STATUS ERRORS WILL BE OFFSET
000409                          *   STACKED AND DATA MISCOMPARES WILL OFFSET THE CARD FOLLOWING
000410                          *   THE DATA ERROR CARD.
000411                          *
000412                          *   EXTRA CARDS SHOULD BE ADDED TO THE BACK OF ANY DECK BEING READ,
000413                          *   PUNCHED OR VERIFIED TO INSURE THE DEVICE IS ALWAYS READY.
000414                          *   AFTER A READ OR PUNCH OPERATION THE CARDS REMAINING IN THE TRANSPORT
000415                          *   SHOULD BE RUNOUT.
000416                          *
000417                          *   FOR TEST MODE TO OPERATE CORRECTLY ON THE 214 READER/
000418                          *   PUNCH OR PUNCH ONLY, ALL PANEL INDICATORS EXCEPT RECHECK
000419                          *   MUST BE OFF (I.E. TRANSPORT).
000420                          *
000421                          *
000422                          * NOTE 3 - FOR PUNCH ONLY
000423                          * ------
000424                          *   CARDS PUNCHED MAY BE VERIFIED ON A READER ON AN DIFFERENT CHANNEL
000425                          *   BY RESPONDING TO "READER CHANNEL NUMBER ?:" WITH THE APPROPRIATE
000426                          *   HEX NUMBER FOR THE THAT CHANNEL, WHILE IN VERIFY MODE.
000427                          *
000428                          * TEST A - ALL TESTS  (R, R/P, P)
000429                          * ------
000430                          *
000431                          *   RESPONDING TO "NEXT ?:" WITH THE LETTER "A" WILL RUN ALL TESTS THAT
000432                          *   ARE PERTINENT TO THE DEVICE BEING TESTED. THIS OPTION IS RECOMMENDED
000433                          *   AT LEAST ONCE FOR THE GIVEN READER, READER/PUNCH OR PUNCH AND
000434                          *   REQUIRES THE DEVICE TO BE PRESENT.
000435                          *
000436                          * TEST B - ELECTRONICS ONLY  (R, R/P, P)
000437                          * ------
000438                          *
000439                          *   ALL ELECTRONIC TESTS WILL BE RUN ON THE PARTICULAR ADAPTER SELECTED.
000440                          *   NO DEVICE IS NEEDED, BUT IF PRESENT IT MUST BE PUT OFF-LINE. THE
000441                          *   INTERRUPT TEST AND TEST MODE ARE ALSO RUN WHEN "B" IS SELECTED.
000442                          *
000443                          * TEST C - DEVICE CONTROLS  (R, R/P, P)
000444                          * ------
000445                          *
000446                          *   TESTS FOR PROPER OPERATION OF THE START AND STOP SWITCHES ON ALL
000447                          *   CARD DEVICES. RUNNING THIS TEST WILL CAUSE THE PROGRAM TO PRINT
000448                          *   "LOAD INPUT, PUT ON-LINE
000449                          *   TYPE 'CARRIAGE RETURN', WHEN READY". THE INPUT HOPPER SHOULD
000450                          *   CONTAIN AT LEAST TWO CARDS, THE OUTPUT STACKER MUST NOT BE FULL
000451                          *   AND THE DEVICE MUST BE PUT ON-LINE. A BREAK WILL RETURN TO "NEXT ?:"
000452                          *   WHILE A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO CONTINUE WITH
000453                          *   THIS TEST. THE TEST THEN REQUIRES THE DEVICE TO BE PUT OFF-LINE BY
000454                          *   "STOP DEVICE", WITHIN 10 SECONDS OR AN ERROR WILL BE REPORTED. THE
```

```
000455                          *   READER/PUNCH OR PUNCH ONLY MUST THEN BE PUT ON-LINE AND THE STACKER
000456                          *   IS TESTED, "SLIDE OUTPUT STACKER", BY SLIDING IT AWAY FROM THE
000457                          *   DEVICE. THE STACKER MUST THEN BE RELEASED, "RELEASE STACKER", AND
000458                          *   THE DEVICE PUT ON LINE. AT THIS POINT THE SAFETY INTERLOCKS FOR
000459                          *   THE LID AND REAR DOOR OF THE PUNCH MAY BE TESTED, BY RESPONDING TO
000460                          *   "TEST INTERLOCKS ?:" WITH "Y". THE LID SHOULD THEN BE OPENED AND
000461                          *   CLOSED, AND THE PUNCH SHOULD BE STARTED AND PUT ON-LINE. THE REAR
000462                          *   DOOR MUST THEN BE OPENED, CLOSED AND THE PUNCH STARTED AND PUT ON-
000463                          *   LINE.
000464                          *
000465                          * TEST D - DEVICE TESTS ONLY  (R, R/P, P)
000466                          * ------
000467                          *
000468                          *   RESPONDING TO "NEXT ?:" WITH "D", WILL RUN ONLY DEVICE INVOLVED
000469                          *   TESTS. THESE TESTS INCLUDE TEST C,M,N,P,V,F,K AND R.
000470                          *
000471                          * TEST F - PUNCH 400 CPM  (R/P, P)
000472                          * ------
000473                          *
000474                          *   THIS TEST WILL PUNCH COLUMNS 6-8 WITH THE DECIMAL VALUE OF THE CARD
000475                          *   POSITION WITHIN THE DECK, FOR A MAXIMUM OF 999 CARDS. FOR EXAMPLE,
000476                          *   CARD #1 WILL HAVE IN COLUMN 8 A 1 PUNCHED AND CARD 999 WILL HAVE IN
000477                          *   COLUMN 6,7 AND 8 A 9 PUNCHED. NO HEX CONVERSION IS NEEDED SINCE THE
000478                          *   CARD NUMBER CAN BE READ DIRECTLY FROM THE CARD. RESPONDING TO
000479                          *   "PUNCH 400 CPM - MAX 999 CARDS, TOTAL ?:" WITH A VALUE AND CARRIAGE
000480                          *   RETURN, WILL PRINT A READY PUNCH MESSAGE. A BREAK AT THIS POINT
000481                          *   RETURNS TO "NEXT ?:" AND A C/R (CARRIAGE RETURN) CONTINUES WITH
000482                          *   THE TEST, EXPECTING THE DEVICE TO BE READY WITH AN INPUT HOPPER
000483                          *   CONTAINING AT LEAST THE NUMBER OF CARDS TO BE PUNCHED + SEVERAL
000484                          *   EXTRA CARDS BEHIND THE DECK.
000485                          *   AFTER PUNCHING THE DECK, THE NUMBER PUNCHED AND IF NO STATUS ERRORS
000486                          *   HAVE OCCURED THEN THE RATE IN CPM, WILL BE REPORTED. UNDER OPTIMUM
000487                          *   CONDITIONS A RATE OF 400 CPM CAN BE ACHIEVED BUT A MORE REALISTIC
000488                          *   RATE OF ABOUT 375 CPM SHOULD BE EXPECTED.
000489                          *   IF ANY STATUS ERRORS HAVE OCCURED THEN THE NEXT CARD WILL BE OFFSET
000490                          *   STACKED.
000491                          *   "VERIFY ?:" WILL THEN BE ASKED. RESPONDING WITH A "Y" AND RELOADING
000492                          *   THE DECK WILL CAUSE THE DECK JUST PUNCHED TO BE VERIFIED FOR PROPER
000493                          *   PUNCHES (SEE TEST V). RESPONDING WITH A "N" TERMINATES THIS TEST.
000494                          *
000495                          * TEST I - INTERRUPT TEST (R, R/P, P)
000496                          * ------
000497                          *
000498                          *   RESPONDING TO "NEXT ?:" WITH "I" AND PUTTING THE DEVICE OFF-LINE
000499                          *   WILL TEST THE ABILITY OF THE SUBSYSTEM TO INTERRUPT A LOWER PRIORITY
000500                          *   AND BE INTERRUPTED BY A HIGHER PRIORITY. NO DEVICE IS NEEDED.
000501                          *
000502                          * TEST K - READ AND PUNCH  (R/P)
000503                          * ------
000504                          *
000505                          *   THIS TEST WILL READ DATA IN COLUMNS 1-8 AND REPEAT IT TO COLUMN 72
000506                          *   OR 8 TIMES. RESPONDING TO "USING 400 CPM DECK?:" WITH "Y", ASSUMES
000507                          *   THAT THE DECK PUNCHED BY TEST "F" IS BEING USED. RESPONDING WITH
000508                          *   "N" MEANS THAT ANY OTHER DECK MAY BE USED PROVIDING COLUMNS 9-80 ARE
000509                          *   BLANK.
000510                          *   (I.E. ASSUME COLUMN 1 HAS A 1 PUNCH, THEN COLUMNS 9,17,25,33,41,49,
000511                          *   57 AND 65 WILL BE PUNCHED WITH A 1 PUNCH)
000512                          *   THE PUNCHED DECK MAY BE VERIFIED.
000513                          *   STATUS ERRORS OFFSET THE NEXT CARD.
000514                          *
000515                          * TEST L - TEST MODE / DATA LOOPBACK  (R, R/P, P)
000516                          * ------
000517                          *
000518                          *   THE DEVICE MUST BE OFF-LINE OR ABSENT FOR THIS TEST. THE ABILITY OF
000519                          *   THE DATA REGISTER ON THE ADAPTER BOARD, OF THE READER (2008), TO
000520                          *   COUNT IN BINARY MODE AND THE ELECTRONICS WHICH DO THE 12 BIT TO 8
000521                          *   BIT CONVERSION IN ASCII MODE, WILL BE TESTED.
000522                          *   FOR THE READER/PUNCH + PUNCH ONLY, A CLOSED DATA PATH WILL BE ESTAB-
000523                          *   LISHED AND DATA COMBINATIONS WILL BE SENT TO THE ADAPTER AND READ
000524                          *   BACK TO CONFIRM THE BINARY PATH. THE ELECTRONICS WHICH DO THE 12 BIT
000525                          *   TO 8 BIT CONVERSION IN ASCII (READER PROMS) AND THE ELECTRONICS
000526                          *   WHICH DO THE 8 BIT TO 12 BIT CONVERSION FROM ASCII (PUNCH PROMS)
000527                          *   WILL BE TESTED. THE CARD JAM ONE SHOT WILL BE TESTED, TO MAKE SURE
000528                          *   THAT A TIME OUT OCCURS BETWEEN 150 MS AND 500 MS.
000529                          *
000530                          * TEST M - ASCII PUNCH  (R/P, P)
000531                          * ------
000532                          *
000533                          *   THIS TEST WILL PUNCH A STANDARD 3 CARD ASCII TEST DECK. A MINIMUM OF
000534                          *   FIVE (5) CARDS ARE TO BE LOADED INTO THE INPUT HOPPER, THE DEVICE
000535                          *   PUT ON-LINE AND A C/R GIVEN IN RESPONSE TO "TYPE 'CARRIAGE RETURN',
000536                          *   WHEN READY".
000537                          *   THE FIRST CARD PUNCHED WILL CONTAIN 64 LEGAL ASCII PUNCHES (ONLY 1
000538                          *   PUNCH IN ROWS 1-7). FOLLOWING THE PUNCHING, CONDITIONS CHECKED
000539                          *   INCLUDE INTERRUPT AT END OF CARD, CORRECT MEMORY ADDRESS AND CORRECT
000540                          *   RESIDUAL RANGE.
000541                          *   THE SECOND CARD WILL BE PUNCHED IN BINARY, WITH AN 11-8-6 PUNCH IN
000542                          *   COLUMN 1 AND A 1-2 PUNCH IN COLUMN 2 (ILLEGAL ASCII). THE THIRD CARD
000543                          *   PUNCH, WILL BE A CARD OF BINARY 0'S (BLANK CARD).
000544                          *   THIS DECK MAY BE VERIFIED, IF ONLY A PUNCH, BY VISUAL INSPECTION
000545                          *   AND/OR BY RESTARTING THE TEST AND RUNNING TEST "N" ON A READER.
000546                          *
000547                          * TEST N - ASCII READ  (R, R/P)
000548                          * ------
000549                          *
000550                          *   TYPING "N" WILL ALLOW THE 3 CARD ASCII DECK PUNCHED BY "M" OR THE
000551                          *   3 CARD ASCII DECK SUPPLIED WITH THE PROGRAM, TO BE VERIFIED. THIS
000552                          *   TEST IS A MUST TO DETERMINE FIXED READ ERRORS, ON A READER ONLY.
000553                          *   IN MARK SENSE MODE A 4 CARD TEST DECK IS NEEDED WITH HALF THE LEGAL
000554                          *   ASCII CHARACTERS ON CARD 1 AND THE REST ON CARD 2, 11-8-6 AND 1-2
000555                          *   MARKS ON CARD 3 AND A BLANK FORTH CARD. THE MARK SENSE DECK IS THE
000556                          *   SAME FOR 40/80 COLUMN IBM MODE AND 80 COLUMN HONEYWELL MODE.
000557                          *   THE LEGAL CHARACTERS FOR ASCII CARD 1 (CARDS 1-2, MARK SENSE) AND
000558                          *   THE PUNCHES (MARKS) ARE SHOWN IN THE ASCII TABLE.
000559                          *   UPON LOADING THE DECK, READYING THE DEVICE AND GIVING A C/R, THE
000560                          *   FIRST CARD WILL BE READ AND INTERRUPT AT END OF RANGE WILL BE
000561                          *   CHECKED, CORRECT MEMORY ADDRESS, CORRECT RESIDUAL RANGE AND CORRECT
000562                          *   DATA TRANSFER WILL ALSO BE CHECKED. DATA MISCOMPARES ARE REPORTED
000563                          *   IN HEX AS VALUE IS, SHOULD BE, IN COLUMN. STRIKING BREAK WILL RETURN
000564                          *   TO THE "NEXT ?:" QUESTION.
000565                          *   IN MARK SENSE MODE CARDS 1 + 2 ARE TREATED AS 1 CARD 160 COLUMNS
000566                          *   LONG (I.E. CARD 1 BEGINS WITH COLUMN 1 AND ENDS WITH COLUMN 80.
000567                          *   CARD 2 BEGINS WITH 81 AND ENDS WITH 160). SINCE MARKS MAY BE IN
```

```
000568                          *   EVEN OR ODD COLUMNS, DEPENDING ON THE PRINTING OF THE CARD, THE
000569                          *   COLUMN REPORTED IN ERROR MAY APPEAR TO BE BETWEEN MARKED COLUMNS ON
000570                          *   THE CARD. IF THIS IS THE CASE, THE COLUMN IN ERROR WILL BE 1 LESS
000571                          *   THEN THE COLUMN REPORTED. (PRINT OUT FOR HIS CARDS)
000572                          *   READING CARD 2 (MARK SENSE CARD 3) IN ASCII, WILL SET THE ILLEGAL
000573                          *   ASCII STATUS BIT, WHICH WILL BE CHECKED AND ON THE R/P THIS CARD
000574                          *   WILL BE OFF SET STACKED. CARD 3 (MARK SENSE CARD 4), THE BLANK CARD,
000575                          *   TESTS TO MAKE SURE THAT THE CORRECT NUMBER OF BYTES OF DATA ARE
000576                          *   TRANSFERED AND THAT THE RESIDUAL RANGE IS CORRECT. THE OPERATOR, ON
000577                          *   A R/P, WILL BE REQUESTED TO VERIFY THAT THE SECOND CARD IS OFFSET
000578                          *   STACKED.
000579                          *
000580                          *   NOTES : ON HONEYWELL MARK SENSE READERS ONLY 80 COLUMN MODE IS
000581                          *           PERMITTED.
000582                          *           MARK SENSE CARD 1 HAS 40 COLUMNS OF MARKS
000583                          *
000584                          * TEST P - PUNCH ANY DECK  (R/P, P)
000585                          * ------
000586                          *
000587                          *   A FIXED OR RANDOM PATTERN MAY BE PUNCHED UNTIL A SPECIFIED MAXIMUM
000588                          *   NUMBER IS REACHED OR UNTIL BREAK IS STRUCK. THE FIRST QUESTION
000589                          *   ASKED WILL BE "PUNCH 100 CARD TEST DECK ?:". RESPONDING WITH "Y"
000590                          *   WILL PUNCH 100 CARDS, WITH THE FIRST CARD HAVING EVERY ROW WITH
000591                          *   EVERY COLUMN PUNCHED, CARDS 2-81 WILL BE PUNCHED IN THE FOLLOWING
000592                          *   SEQUENCE: CARD 2 WILL HAVE A 1 PUNCH (BINARY 1) IN COLUMN 1, CARD 3
000593                          *   WILL HAVE A 2 PUNCH (BINARY 2 / ROW 8) IN COLUMN 2 AND SO ON WITH
000594                          *   CARD 81 HAVING AN 80 PUNCH (BINARY 50 / ROW 3 + 5). THE LAST 19
000595                          *   CARDS WILL BE PUNCHED WITH RANDOM DATA.
000596                          *   RESPONDING WITH "N" WILL CAUSE "RANDOM PUNCHES?:" TO BE ASKED.
000597                          *   TYPING "Y" AND SPECIFING THE NUMBER OF CARDS TO BE PUNCHED, WILL
000598                          *   PUNCH A DECK OF CARDS WITH THE FIRST 100 CARDS BEING THE SAME AS
000599                          *   THE 100 CARD TEST DECK AND THE REST OF THE DECK WILL BE RANDOM
000600                          *   PUNCHES.
000601                          *   RESPONDING TO "RANDOM PUNCHES ?:" WITH "N" WILL ASK "COPY A CARD ?:".
000602                          *   A "Y" RESPONSE, WILL ALLOW A CARD TO BE READ ON THE READER/PUNCH OR
000603                          *   ON A DIFFERENT CHANNEL (READER), IF A PUNCH ONLY IS BEING TESTED.
000604                          *   AFTER THE CARD IS READ, THE R/P, MUST BE STOPPED AND RUNOUT. SHOULD
000605                          *   THERE BE NO OTHER READER AVAILABLE AND THIS ROUTINE IS ENTERED,
000606                          *   STRIKING BREAK DURING THE PRINTOUT OF "READER CHANNEL NUMBER ?:"
000607                          *   WILL RETURN CONTROL TO "NEXT ?:".
000608                          *   RESPONDING TO "COPY A CARD ?:" WITH "N", ALLOWS DATA TO BE INPUTTED
000609                          *   FROM THE KEYBOARD, 3 HEX VALUES PER COLUMN, UNTIL JUST A RETURN IS
000610                          *   RECEIVED. FOR EXAMPLE TYPING: 101 C/R
000611                          *                                   1 C/R
000612                          *                                   0 C/R
000613                          *                                  23 C/R
000614                          *                                  C/R
000615                          *   WILL RESULT IN COLUMN 1 HAVING A 1-9 PUNCH, COLUMN 2 HAVING A 9
000616                          *   PUNCH, COLUMN 3 BEING BLANK, COLUMN 4 HAVING A 4-8-9 PUNCH AND THE
000617                          *   REST OF THE CARD BLANK.
000618                          *   RESPONDING TO "100 CARD TEST DECK" WITH "N", WILL ASK FOR THE
000619                          *   NUMBER OF CARDS TO BE PUNCHED (DECIMAL). THE PUNCHING OPERATION
000620                          *   MAY BE TERMINATE BEFORE THE TOTAL IS REACHED BY STRIKING BREAK.
000621                          *   AFTER THE DECK IS PUNCHED, THE NUMBER PUNCHED AND THE RATE IN CPM,
000622                          *   PROVIDING NO STATUS ERRORS HAVE OCCURED, WILL BE REPORTED. SHOULD
000623                          *   STATUS ERRORS OCCUR, THEN THE NEXT CARD AFTER THE ERROR CARD WILL
000624                          *   THE DECK MAY BE VERIFIED.
000625                          *
000626                          * TEST R - READ ANY DECK  (R R/P)
000627                          * ------
000628                          *
000629                          *   THIS TEST WILL READ ANY DECK AND WILL GENERATE CHECKSUMS (PARITY),
000630                          *   WHICH ARE STORED IN MEMORY. SELECTING "R" WILL PRINT THE MAXIMUM
000631                          *   NUMBER OF CARDS AND WILL ASK FOR THE "DESIRED TOTAL ?:" ENTERING A
000632                          *   DECIMAL VALUE AND A C/R WILL CAUSE THE CARDS TO BE READ. SHOULD
000633                          *   THE DEVICE GO OFF-LINE DURING THE READ, THE NUMBER READ AND THE
000634                          *   NUMBER SPECIFIED  WILL BE REPORTED AND CONTROL RETURNS TO "NEXT ?:".
000635                          *   SUCCESSFUL READING OF THE DECK RESULTS IN THE RATE IN CPM BEING
000636                          *   REPORTED.
000637                          *
000638                          * TEST S - FRESH START  (R, R/P, P)
000639                          * ------
000640                          *
000641                          *   THIS IS NOT A TEST IN ITSELF, BUT IS A METHOD TO RESTART EXECUTION
000642                          *   OF THE PROGRAM. IT IS EQUIVALENT TO CLEARING DO AND EXECUTING FROM
000643                          *   LOCATION 100.
000644                          *   THE CHANNEL NUMBER IS ASK FOR AND THE DEVICE IS IDENTIFIED AS A
000645                          *   READER, READER/PUNCH OR PUNCH. IF IT IS NOT ONE OF THESE THREE, THEN
000646                          *   THE QUESTION IS REPEATED. IF THE ID SHOWS A READER IS TO BE TESTED,
000647                          *   THEN MARK SENSE QUESTIONS WILL BE ASKED
000648                          *   RESPONDING TO "CHANNEL NUMBER ?:" WITH A C/R RETAINS THE OLD VALUE.
000649                          *   SEE OPERATION FOR A FURTHER EXPLAINATION.
000650                          *
000651                          * TEST V - VERIFY ANY DECK  (R, R/P)
000652                          * ------
000653                          *
000654                          *   IF THE DEVICE IS A PUNCH ONLY, THE OPERATOR MAY SELECT A READER
000655                          *   ON ANOTHER CHANNEL BY ANSWERING "READER CHANNEL NUMBER ?:" WITH THE
000656                          *   CHANNEL OF THE READER. SHOULD THERE NOT BE A READER AVAILABLE, THE
000657                          *   OPERATOR MAY STRIKE THE BREAK KEY DURING THE ABOVE PRINT OUT AND
000658                          *   CONTROL RETURNS TO "NEXT ?:"
000659                          *   THIS TEST CAN BE CALLED FROM TESTS F,K,P AND R, AND WILL VERIFY
000660                          *   THE PREVIOUS READ OR PUNCH OPERATION.
000661                          *   DURING THE READING OF CARDS, THE STOP/START SWITCHES MAY BE TESTED
000662                          *   WITH NO ILL EFFECTS TO THE COMPARISIONS.
000663                          *   ANY CARD IN ERROR, WILL BE REPORTED ALONG WITH ROWS AND COLUMNS IN
000664                          *   ERROR AND BIT DIFFERENCE. DROPPED OR MISSING STROBES WILL RESULT IN
000665                          *   ONLY THE CARD AND FIRST COLUMN IN ERROR BEING REPORTED, ALONG WITH
000666                          *   "DROPPED STROBE".
000667                          *   THE OPERATOR CAN SELECT A RANDOM OR FIXED DELAY BETWEEN CARD PICKS.
000668                          *   THE VALUE HE/SHE SELECTS WILL BE ROUNDED TO THE NEAREST CLOCK TICK.
000669                          *   IF ANY ERRORS ARE DETECTED AFTER THE THIRD CARD, THEN AT THE END OF
000670                          *   THIS TEST, THE "AVERAGE DELAY IN MSEC" WILL BE REPORTED (ROUNDED TO
000671                          *   THE NEAREST TICK).
000672                          *   SHOULD THE OPERATOR TYPE "V" WITHOUT ANY PREVIOUS READ OR PUNCH
000673                          *   OPERATION, THE PROGRAM WILL ASK "HAVE A TEST DECK ?:". IF THE 100
000674                          *   CARD TEST DECK IS AVAILABLE, HE SHOULD TYPE "Y". AT THIS POINT
000675                          *   SEVERAL SECONDS WILL BE SPENT TO GENERATE THE CHECKSUMS, BEFORE
000676                          *   HE IS INSTRUCTED TO READY THE DECK.
000677                          *   IF THE ERROR PRINT OUT IS TO BE TERMINATE, THEN STRIKING BREAK WILL
000678                          *   DO SO.
000679                          *   MARK SENSE COLUMN VALUES APPLY TO HIS CARDS, IBM CARDS ARE 1 LESS
000680                          *   THEN THE COLUMN NUMBER REPORTED.
```

```
000681                                  /
000682                                  *
000683                                  * GLOBAL CONSTANTS AND VARIABLES
000684                                  *
000685  0000                            XDEF    (ID,D)
000686  0001                            XDEF    (STATWA,D+1)
000687  0002                            XDEF    (STATUS,D+2)
000688  0003                            XDEF    (HERTZ,D+3)
000689  0007                            XDEF    (CHANNE,D+7)
000690  0004                            XDEF    (PUESDO,D+4)
000691  0005                            XDEF    (TESTTY,D+5)
000692  0006                            XDEF    (TYPEOF,D+6)
000693  0008                            XDEF    (SETTOH,D+8)
000694  0009                            XDEF    (SETTOA,D+9)
000695  000A                            XDEF    (ASEJCT,D+10)
000696  000B                            XDEF    (BINARY,D+11)
000697  0014                            XDEF    (READST,D+20)
000698  000F                            XDEF    (NUMBER,D+15)
000699  000D                            XDEF    (LABEL1,D+13)
000700  000E                            XDEF    (LABEL2,D+14)
000701  0011                            XDEF    (MAXIMU,D+17)
000702  0012                            XDEF    (CARDSS,D+18)
000703  0013                            XDEF    (CARDSR,D+19)
000704  0015                            XDEF    (BITCOU,D+21)
000705  0016                            XDEF    (LOWADD,D+22)
000706  0017                            XDEF    (RORVMO,D+23)
000707  0019                            XDEF    (HAVEGE,D+25)
000708  0010                            XDEF    (SPEED,D+16)
000709  000C                            XDEF    (ERRORF,D+12)
000710  0022                            XDEF    (CKSEM,D+34)
000711  0020                            XDEF    (L21SEM,D+32)
000712  0023                            XDEF    (DLVL,D+35)
000713  0021                            XDEF    (DVSEM,D+33)
000714  0024                            XDEF    (CPLEVE,D+36)
000715  0025                            XDEF    (LEVTYP,D+37)
000716  001B                            XDEF    (TEMPCH,D+27)
000717  0018                            XDEF    (SREG,D+24)
000718  001C                            XDEF    (SCR,D+28)
000719  001D                            XDEF    (IOTM,D+29)
000720  001E                            XDEF    (IOX1,D+30)
000721  001F                            XDEF    (IOX2,D+31)
000722  001A                            XDEF    (READOR,D+26)
000723  019E                            XDEF    (CHECKS,A+181)
000724  0146                            XDEF    (BINCOU,A+93)
000725  0153                            XDEF    (ASCCNT,A+106)
000726  015D                            XDEF    (BINOB,A+116)
000727  0171                            XDEF    (PPROMS,A+136)
000728  0183                            XDEF    (RPROMS,A+154)
000729  0195                            XDEF    (ONESHO,A+172)
000730  0274                            XDEF    (WORDCO,A+395)
000731  01A5                            XDEF    (BUFF1,A+188)
000732  01F9                            XDEF    (BUFF2,A+272)
000733  0249                            XDEF    (CHAR,A+352)
000734  0271                            XDEF    (ILLEGA,A+392)
000735  00E9                            XDEF    (ISAC,A)
000736  0108                            XDEF    (ISAD,A+31)
000737  0127                            XDEF    (ISAP,A+62)
000738  02F5                            XDEF    (IOWORD,A+524)
000739  004F                            XDEF    FRESHS
000740  00CA                            XDEF    LETR
000741  00CE                            XDEF    LETV
000742  0051                            XDEF    ASKNEX
000743  004C                            XDEF    CARD
000744  0008                    ZKCOM   COMM    8
000745  0000  0000              D       RESV    41,0
000746  0028                    S1      EQU     $-1
000747  0029  008C              S2      DC      <L2
000748  002A  008E              S3      DC      <L3
000749  002B  00A2              S4      DC      <L5
000750  002C  00A2              S5      DC      <L5
000751  002D  0083              S6      DC      <L32
000752  002E  00D5              S7      DC      <L39
000753  002F  0083              S8      DC      <L32
000754  0030  0083              S9      DC      <L32
000755  0031  0090              S10     DC      <L8
000756  0032  0083              S11     DC      <L32
000757  0033  00DF              S12     DC      <L41
000758  0034  009A              S13     DC      <L34
000759  0035  00AC              S14     DC      <L36
000760  0036  00B6              S15     DC      <L37
000761  0037  0083              S16     DC      <L32
000762  0038  00C0              S17     DC      <L38
000763  0039  0083              S18     DC      <L32
000764  003A  00CA              S19     DC      <LETR
000765  003B  004F              S20     DC      <FRESHS
000766  003C  0083              S21     DC      <L32
000767  003D  0083              S22     DC      <L32
000768  003E  00CE              S23     DC      <LETV
000769  003F  0083              S24     DC      <L32
000770  0040  0083              S25     DC      <L32
000771  0041  0083              S26     DC      <L32
000772  0042  00E3              S27     DC      <L17
000773  0043  0005              T29     DC      5
000774  0044  4E45 5854 2400            TEXT    'NEXT$'
000775  0047  0008              T33     DC      8
000776  0048  494E 5641 4C49            TEXT    'INVALID$'
        004B  4424
000777  004C                    CARD    EQU     $
000778  004C  8F00 0001      K         SAVE    <ZKCOM+1,Z'0011'
        004E  0011
000779                                  *       'INTEGER' ID,STATUS WAS,STATUS SHOULD BE,HERTZ;
000780                                  *       'INTEGER' PUESDO CARD STATUS,TEST TYPE;
000781                                  *       'INTEGER' TYPE OF PUNCHES;
000782                                  *       'INTEGER' CHANNEL := 'HEX' (1300);
000783                                  *       'INTEGER' SET TO HOLLERITH := 'HEX' (4400);
000784                                  *       'INTEGER' SET TO ASCII := 0;
000785                                  *       'INTEGER' ASCII ERROR EJECT := 'HEX' (4000);
000786                                  *       'INTEGER' BINARY MODE := 'HEX' (8000);
000787                                  *       'INTEGER' ERROR FLAG; ( FLAG FOR NUMBER OF MESSAGES PRINTED)
000788                                  *       'INTEGER' LABEL1;
000789                                  *       'INTEGER' LABEL2;
000790                                  *       'INTEGER' NUMBER OF COLUMNS;
000791                                  *       'INTEGER' SPEED := 'HEX' (18);
```

```
000792                          *       'INTEGER' MAXIMUM CARDS;
000793                          *       'INTEGER' CARDS SPECIFIED := 100;
000794                          *       'INTEGER' CARDS READ;
000795                          *       'INTEGER' READ STATUS;
000796                          *       'INTEGER' BIT COUNT;
000797                          *       'INTEGER' LOW ADDRESS;
000798                          *       'INTEGER' R OR V MODE;
000799                          *       'INTEGER' SREG;
000800                          *       'INTEGER' HAVE GENERATED CHECKSUMS;
000801                          *       'INTEGER' READ OR PUNCH;
000802                          *       'INTEGER' TEMP CHARACTER;
000803                          **;
000804                          ** MICRO CONSTANTS AND VARIABLES;
000805                          **;
000806                          *       'INTEGER' SET CHANNEL READY      := 'HEX' (4018);
000807                          *       'INTEGER' SET TEST MODE          := 'HEX' (2000);
000808                          *       'INTEGER' WAIT LOOP ADDRESS      := 'HEX' (0000);
000809                          *       'INTEGER' RETURN FROM TEST MODE  := 'HEX' (0500);
000810                          **;
000811                          **   INTERRUPT SAVE AREA AND VARIABLES;
000812                          **;
000813                          *       'TABLE' ISA FOR CLOCK            @31,1@
000814                          *            @ISACD                      'INTEGER' 2;
000815                          *             ISACP                      'INTEGER' 5@;
000816                          *       'TABLE' ISA FOR DEVICE           @31,1@
000817                          *            @ISARD                      'INTEGER' 2;
000818                          *             ISARP                      'INTEGER' 5@;
000819                          *       'TABLE' ISA FOR PROGRAM          @31,1@
000820                          *            @ISATD                      'INTEGER' 2;
000821                          *             ISATP                      'INTEGER' 5@;
000822                          *       'INTEGER' L2ISEMAPHORE,
000823                          *                 DEVICESEMAPHORE,
000824                          *                 CLOCKSEMAPHORE,
000825                          *                 DEVICELEVEL,
000826                          *                 CPLEVEL,
000827                          *                 LEVTYPE;
```

```
000828                          /;
000829                          **;
000830                          **  MICRO CODE ARRAYS;
000831                          **;
000832                          **       2008  READER;
000833                          **       MICROS TO MAKE 2008 COUNT IN BINARY;
000834                          *       'INTEGER' 'ARRAY' BINCOUNT @0:12@ :=
000835                          *            'HEX' (0009), 'HEX'(A0C0), 'HEX' (9E40), 'HEX' (9801), 'HEX'(2100),
000836                          **     INST COUNT,SET INDEX MODE,DATA SELECTOR,CLEAR NDSR,DATA BY
000837                          *       TE TAKE;
000838                          *            'HEX' (88EA), 'HEX'(B200), 'HEX' (9900), 'HEX' (88EB), 'HEX'(B200);
000839                          **     SSPA LOC 3E ,MWT LOW BYTE,LCN AAD3 40 PICK,SSPA 3F, MWT HI
000840                          *       GH BYTE;
000841                          **;
000842                          **       MICROS TO MAKE 2008 COUNT IN ASCII;
000843                          *       'INTEGER' 'ARRAY' ASCCNT @0:9@ :=
000844                          *            'HEX' (0006), 'HEX' (A0C0), 'HEX' (9C40), 'HEX' (9801),
000845                          **     INSTR COUNT, SET INDEX MODE,DATA SELECTOR,CLEAR NDSR;
000846                          *            'HEX' (88EA), 'HEX' (B200), 'HEX' (2100);
000847                          **     SSPA LOC 3E ,MWT AAD1 LOW BYTE, DATA BYTE TAKEN;
000848                          **       208A / 2088;
000849                          **;
000850                          **       MICROS FOR BINARY LOOPBACK;
000851                          *       'INTEGER' 'ARRAY' BINOB @0:19@  :=
000852                          *            'HEX' (0010), 'HEX' (9848), 'HEX' (8100),'HEX' (2034),'HEX' (88EB),
000853                          **     INSTR COUNT,TEST & BINARY MODE, SET PUNCHX+00       , SSPA
000854                          *       LOC 3F;
000855                          *            'HEX' (716A), 'HEX' (88EA), 'HEX' (716A), 'HEX' (0000),'HEX'(8180),
000856                          **     XFB AAD1 BSPM SRIA,SSPA 3E, XFB AAD1     ,CLOCK STROBE, SET
000857                          *       ;
000858                          *            'HEX' (2034), 'HEX' (88EB), 'HEX' (B200), 'HEX' (2100),'HEX'(88EA),
000859                          **     READX+00,SSPA LOC 3F ,MWT  HIGH BYTE ,DATA BYTE TAKEN,SSPA
000860                          *       LOC3E;
000861                          *            'HEX' (B200), 'HEX' (0008);
000862                          **     MWT  LOW BYTE, CLEAR ADAPTER;
000863                          **;
000864                          **       MICROS TO TEST PUNCH PROMS;
000865                          *       'INTEGER' 'ARRAY' PPROMS @0:17@ :=
000866                          *            'HEX' (000F), 'HEX' (9840), 'HEX' (8100), 'HEX' (2034),'HEX'(88EA),
000867                          **     INST COUNT, TEST &ASCII MODE, SET PUNCHX+00       , SSPA L
000868                          *       OC 3E;
000869                          *            'HEX' (716A), 'HEX' (0000),
000870                          **     XFB AAD1 BSPM SRIA, STROBE ADAPT;
000871                          *            'HEX' (9848), 'HEX' (88EA), 'HEX' (B200),'HEX'(2100),
000872                          **     SET BINARY,SSPA LOC 3E, MWT HIGH BYTE,DATA BYTE;
000873                          *            'HEX' (8180), 'HEX' (2034), 'HEX' (88EB), 'HEX' (B200),'HEX'(0008);
000874                          **     SET   READX+00              ,SSPA LOC 3F  ,MWT LOW BYTE , CL
000875                          *       EAR DAP;
000876                          **;
000877                          **       MICROS TO TEST READ PROMS;
000878                          *       'INTEGER' 'ARRAY' RPROMS @0:17@ :=
000879                          *            'HEX' (000E), 'HEX' (9848), 'HEX' (8100), 'HEX' (2034),'HEX'(88EB),
000880                          **     INST COUNT,TEST & BIN MODE, SET PUNCHX+00           ,SSPA
000881                          *       LOC 3F;
000882                          *            'HEX' (716A), 'HEX' (88EA), 'HEX' (716A), 'HEX' (0000),'HEX'(9840),
000883                          **     XFB AAD1 BSPM SRIA,SSPA 3E,XFB AAD1      ,CLOCK STROBE ,SET
000884                          *       ASCII;
000885                          *            'HEX' (8180), 'HEX' (2034), 'HEX' (88EA), 'HEX' (B200),'HEX'(0008);
000886                          **     SET  READX+00              ,SSPA LOC 3E  ,MWT ASCII    ,CLE
000887                          *       AR ADAP;
000888                          **;
000889                          **       MICROS TO TIME CARD JAM ONE SHOT;
000890                          *       'INTEGER''ARRAY' ONE SHOT @0:8@ :=
000891                          *            'HEX' (0005), 'HEX' (0008), 'HEX' (8180), 'HEX' (2034),
000892                          **     INSTR COUNT, INITIALIZE ADAPTER, DON'T RESET ONE SHOT;
000893                          *            'HEX' (8180), 'HEX' (2034);
000894                          **     STROBE ONE SHOT;
000895                          **;
000896                          ** GLOBAL R/W BUFFERS;
000897                          **;
000898                          *       'INTEGER' 'ARRAY' CHECKSUM @0:6@;
000899                          *       'INTEGER' 'ARRAY' BUFFER @0:83@;
000900                          *       'INTEGER' 'ARRAY' BUFFER2 @0:79@;
000901                          **;
000902                          ** ASCII CHARACTERS FOR PUNCHED CARD 1 AND MARKED CARDS 1 + 2;
000903                          **;
000904                          *       'INTEGER' 'ARRAY' CHAR@0:39@ :=
000905                          **;
000906                          *       'COMMENT' HEX VALUE          ACTUAL PUNCHES  SYMBOL ;
000907                          *       'COMMENT'      L R               L R          L R  ;
000908                          *            'HEX' (3031), 'COMMENT'      0/1          0/1  ;
000909                          *            'HEX' (3233), 'COMMENT'      2/3          2/3  ;
000910                          *            'HEX' (3435), 'COMMENT'      4/5          4/5  ;
000911                          *            'HEX' (3637), 'COMMENT'      6/7          6/7  ;
000912                          *            'HEX' (3839), 'COMMENT'      8/9          8/9  ;
000913                          *            'HEX' (4142), 'COMMENT'  12-1/12-2        A/B  ;
000914                          *            'HEX' (4344), 'COMMENT'  12-3/12-4        C/D  ;
000915                          *            'HEX' (4546), 'COMMENT'  12-5/12-6        E/F  ;
000916                          *            'HEX' (4748), 'COMMENT'  12-7/12-8        G/H  ;
000917                          *            'HEX' (494A), 'COMMENT'  12-9/11-1        I/J  ;
000918                          *            'HEX' (4B4C), 'COMMENT'  11-2/11-3        K/L  ;
000919                          *            'HEX' (4D4E), 'COMMENT'  11-4/11-5        M/N  ;
000920                          *            'HEX' (4F50), 'COMMENT'  11-6/11-7        O/P  ;
000921                          *            'HEX' (5152), 'COMMENT'  11-8/11-9        Q/R  ;
000922                          *            'HEX' (5354), 'COMMENT'   0-2/0-3         S/T  ;
000923                          *            'HEX' (5556), 'COMMENT'   0-4/0-5         U/V  ;
000924                          *            'HEX' (5758), 'COMMENT'   0-6/0-7         W/X  ;
000925                          *            'HEX' (595A), 'COMMENT'   0-8/0-9         Y/Z  ;
000926                          *            'HEX' (3A23), 'COMMENT'   8-2/8-3     COLON/NUMBER SIGN;
000927                          *            'HEX' (4027), 'COMMENT'   5-4/8-5        AT/SGL QUOTE  ;
000928                          ** SECOND MARK SENSE CARD;
000929                          *            'HEX' (3D22), 'COMMENT'   8-6/8-7     EQUAL/DBL QUOTE  ;
000930                          *            'HEX' (202F), 'COMMENT' BLANK/0-1     SPACE/FOR SLANT  ;
000931                          *            'HEX' (5C2C), 'COMMENT' 0-8-2/0-8-3 B SLAN/COMMA       ;
000932                          *            'HEX' (255F), 'COMMENT' 0-8-4/0-8-5     PCT/LFT ARROW  ;
000933                          *            'HEX' (3E3F), 'COMMENT' 0-8-6/0-8-7     >/?            ;
000934                          *            'HEX' (2D5D), 'COMMENT'     11/11-8-2  DASH/@          ;
000935                          *            'HEX' (242A), 'COMMENT'11-8-3/11-8-4     $/*            ;
000936                          *            'HEX' (295E), 'COMMENT'11-8-5/11-8-7 RTPAR/)            ;
000937                          *            'HEX' (265B), 'COMMENT'     12/12-8-2    AND/[          ;
000938                          *            'HEX' (2E3C), 'COMMENT'12-8-3/12-8-4PERIOD/ <           ;
000939                          *            'HEX' (282B), 'COMMENT'12-8-5/12-8-6LFT PA/PLUS         ;
000940                          *            'HEX' (217D), 'COMMENT'12-8-7/11-0     X PT/RT ENC      ;
```

```
000941                          *           'HEX' (7B20), 'COMMENT'  12-0/BLANK LFTENC/SPACE      ;
000942                          *           'HEX' (2020), 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000943                          *           'HEX' (2020), 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000944                          *           'HEX' (2020), 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000945                          *           'HEX' (2020), 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000946                          *           'HEX' (2020), 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000947                          *           'HEX' (2020), 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000948                          *           'HEX' (2020); 'COMMENT' BLANK/BLANK  SPACE/SPACE      ;
000949                          **;
000950                          ** ASCII CHARACTERS FOR PUNCHED CARD 2 AND MARKED CARD 3;
000951                          **;
000952                          *       'INTEGER' 'ARRAY' ILLEGAL ASCII @0:2@ := 'HEX' (020A),'HEX' (0180);
000953                          *       'COMMENT'   COL    PUNCH      GRAPHIC      ;
000954                          *       'COMMENT'     1    11-8-6     SEMI-COLON   ;
000955                          *       'COMMENT'     2    1-2        NONE         ;
000956                          *       'COMMENT'  3-80    BLANK      SPACE        ;
000957                          **;
000958                          ** BINARY TO ASCII LOOK UP TABLES - FOR TEST MODE;
000959                          **;
000960                          *       'INTEGER' 'ARRAY' WORD CONVERSION TABLE @0:128@ :=
000961                          *           'HEX' (2039),    'COMMENT' ' NO PUNCH/9  SPACE/9;
000962                          *           'HEX' (3898),    'COMMENT'  8/9-8  8/NA ;
000963                          *           'HEX' (3704),    'COMMENT'  7/9-7  7/EOT ;
000964                          *           'HEX' (221A),    'COMMENT'  8-7/9-8-7  "/S ;
000965                          *           'HEX' (3696),    'COMMENT'  6/9-6  6/NA ;
000966                          *           'HEX' (3D9E),    'COMMENT'  8-6/9-8-6  =/NA ;
000967                          *           'HEX' (3595),    'COMMENT'  5/9-5  5/NA ;
000968                          *           'HEX' (2715),    'COMMENT'  8-5/9-8-5  '/NAK ;
000969                          *           'HEX' (3494),    'COMMENT'  4/9-4  4/NA ;
000970                          *           'HEX' (4014),    'COMMENT'  8-4/9-8-4  AT/DC4 ;
000971                          *           'HEX' (3393),    'COMMENT'  3/9-3  3/NA ;
000972                          *           'HEX' (239B),    'COMMENT'  8-3/9-8-3 POUND/NA ;
000973                          *           'HEX' (3216),    'COMMENT'  2/9-2  2/SYN ;
000974                          *           'HEX' (3A9A),    'COMMENT'  8-2/9-8-2  :/NA ;
000975                          *           'HEX' (3191),    'COMMENT'  1/9-1  1/NA ;
000976                          *           'HEX' (6099),    'COMMENT'  8-1/9-8-1 B.S.(L.C.)/NA ;
000977                          *           'HEX' (305A),    'COMMENT'  0/0-9  0/Z ;
000978                          *           'HEX' (5988),    'COMMENT'  0-8/0-9-8  Y/NA ;
000979                          *           'HEX' (581B),    'COMMENT'  0-7/0-9-7  X/ESC ;
000980                          *           'HEX' (3F07),    'COMMENT'  0-8-7/0-9-8-7  ?/BEL ;
000981                          *           'HEX' (5717),    'COMMENT'  0-6/0-9-6  W/ETB ;
000982                          *           'HEX' (3E06),    'COMMENT'  0-8-6/0-9-8-6  >/ACK ;
000983                          *           'HEX' (560A),    'COMMENT'  0-5/0-9-5  V/LF ;
000984                          *           'HEX' (5F05),    'COMMENT'  0-8-5/0-9-8-5  LPA/ENQ ;
000985                          *           'HEX' (5584),    'COMMENT'  0-4/0-9-4  U/NA ;
000986                          *           'HEX' (258C),    'COMMENT'  0-8-4/0-9-8-4  PERCENT/NA ;
000987                          *           'HEX' (5483),    'COMMENT'  0-3/0-9-3  T/NA ;
000988                          *           'HEX' (2C8B),    'COMMENT'  0-8-3/0-9-8-3 ,/NA ;
000989                          *           'HEX' (5382),    'COMMENT'  0-2/0-9-2  S/NA ;
000990                          *           'HEX' (5C8A),    'COMMENT'  0-8-2/0-9-8-2  B.S./NA ;
000991                          *           'HEX' (2F81),    'COMMENT'  0-1/0-9-1  / /  NA ;
000992                          *           'HEX' (B989),    'COMMENT'  0-8-1/0-9-8-1  NA/NA ;
000993                          *           'HEX' (2D52),    'COMMENT'  11/11-9  -/R ;
000994                          *           'HEX' (5118),    'COMMENT'  11-8/11-9-8  Q/CAN ;
000995                          *           'HEX' (5087),    'COMMENT'  11-7/11-9-7  P/NA ;
000996                          *           'HEX' (5E1F),    'COMMENT'  11-8-7/11-9-8-7  APU/US ;
000997                          *           'HEX' (4F08),    'COMMENT'  11-6/11-9-6  O/BS ;
000998                          *           'HEX' (3B1E),    'COMMENT'  11-8-6/11-9-8-6  S.C./RS ;
000999                          *           'HEX' (4E85),    'COMMENT'  11-5/11-9-5  N/NA ;
001000                          *           'HEX' (291D),    'COMMENT'  11-8-5/11-9-8-5  )/GS ;
001001                          *           'HEX' (4D9D),    'COMMENT'  11-4/11-9-4  M/NA ;
001002                          *           'HEX' (2A1C),    'COMMENT'  11-8-4/11-9-8-4  */FS ;
001003                          *           'HEX' (4C13),    'COMMENT'  11-3/11-9-3  L/DC3 ;
001004                          *           'HEX' (248F),    'COMMENT'  11-2/11-9-2  K/DC2 ;
001005                          *           'HEX' (4B12),    'COMMENT'  11-2/11-9-2  K/DC2 ;
001006                          *           'HEX' (5D92),    'COMMENT'  11-8-2/11-9-8-2  ./NA ;
001007                          *           'HEX' (4A11),    'COMMENT'  11-1/11-9-1  J/DC1 ;
001008                          *           'HEX' (B119),    'COMMENT'  11-8-1/11-9-8-1  NA/EM ;
001009                          *           'HEX' (7D7A),    'COMMENT'  11-0/11-0-9  .(L.C.)/Z(L.C.) ;
001010                          *           'HEX' (79B8),    'COMMENT'  11-0-8/11-0-9-8  Y(L.C.)/NA ;
001011                          *           'HEX' (78B7),    'COMMENT'  11-0-7/11-0-9-7  X(L.C.)/NA ;
001012                          *           'HEX' (D7F9),    'COMMENT'  11-0-8-7/11-0-9-8-7  NA/NA ;
001013                          *           'HEX' (77B6),    'COMMENT'  11-0-6/11-0-9-6  W(L.C.)/NA ;
001014                          *           'HEX' (D6F8),    'COMMENT'  11-0-8-6/11-0-9-8-6  NA/NA ;
001015                          *           'HEX' (76B5),    'COMMENT'  11-0-5/11-0-9-5  V(L.C.)/NA ;
001016                          *           'HEX' (D5F7),    'COMMENT'  11-0-8-5/11-0-9-8-5  NA/NA ;
001017                          *           'HEX' (75B4),    'COMMENT'  11-0-4/11-0-9-4  U(L.C.)/NA ;
001018                          *           'HEX' (D4F6),    'COMMENT'  11-0-8-4/11-0-9-8-4  NA/NA ;
001019                          *           'HEX' (74B3),    'COMMENT'  11-0-3/11-0-9-3  T(L.C.)/NA ;
001020                          *           'HEX' (D3F5),    'COMMENT'  11-0-8-3/11-0-9-8-3  NA/NA ;
001021                          *           'HEX' (73B2),    'COMMENT'  11-0-2/11-0-9-2  S(L.C.)/NA ;
001022                          *           'HEX' (D2F4),    'COMMENT'  11-0-8-2/11-0-9-8-2  NA/NA ;
001023                          *           'HEX' (7E9F),    'COMMENT'  11-0-1/11-0-9-1  @/NA ;
001024                          *           'HEX' (D180),    'COMMENT'  11-0-8-1/11-0-9-8-1  NA/NA ;
001025                          *           'HEX' (2649),    'COMMENT'  12/12-9  I/I ;
001026                          *           'HEX' (4897),    'COMMENT'  12-8/12-9-8  H/NA ;
001027                          *           'HEX' (477F),    'COMMENT'  12-7/12-9-7  G/DEL ;
001028                          *           'HEX' (210F),    'COMMENT'  12-8-7/12-9-8-7  @/SI ;
001029                          *           'HEX' (4686),    'COMMENT'  12-6/12-9-6  F/NA ;
001030                          *           'HEX' (2B0E),    'COMMENT'  12-8-6/12-9-8-6  +/SO ;
001031                          *           'HEX' (4509),    'COMMENT'  12-5/12-9-5  E/HT ;
001032                          *           'HEX' (280D),    'COMMENT'  12-8-5/12-9-8-5  (/CR ;
001033                          *           'HEX' (449C),    'COMMENT'  12-4/12-9-4  D/NA ;
001034                          *           'HEX' (3C0C),    'COMMENT'  12-8-4/12-9-8-4  </FF ;
001035                          *           'HEX' (4303),    'COMMENT'  12-3/12-9-3  C/ETX ;
001036                          *           'HEX' (2E0B),    'COMMENT'  12-8-3/12-9-8-3  ./VT ;
001037                          *           'HEX' (4202),    'COMMENT'  12-2/12-9-2  B/STX ;
001038                          *           'HEX' (5B8E),    'COMMENT'  12-8-2/12-9-8-2  @/NA ;
001039                          *           'HEX' (4101),    'COMMENT'  12-1/12-9-1  A/SOH ;
001040                          *           'HEX' (A88D),    'COMMENT'  12-8-1/12-9-8-1  NA/NA ;
001041                          *           'HEX' (7B69),    'COMMENT'  12-0/12-0-9  @(L.C.)/1(L.C.) ;
001042                          *           'HEX' (68A7),    'COMMENT'  12-0-8/12-0-9-8  H(L.C.)/NA ;
001043                          *           'HEX' (67A6),    'COMMENT'  12-0-7/12-0-9-7  G(L.C.)/NA ;
001044                          *           'HEX' (C9ED),    'COMMENT'  12-0-8-7/12-0-9-8-7  NA/NA ;
001045                          *           'HEX' (66A5),    'COMMENT'  12-0-6/12-0-9-6  F(L.C.)/NA ;
001046                          *           'HEX' (C8EC),    'COMMENT'  12-0-8-6/12-0-9-8-6  NA/NA ;
001047                          *           'HEX' (65A4),    'COMMENT'  12-0-5/12-0-9-5  E(L.C.)/NA ;
001048                          *           'HEX' (C7EB),    'COMMENT'  12-0-8-5/12-0-9-8-5  NA/NA ;
001049                          *           'HEX' (64A3),    'COMMENT'  12-0-4/12-0-9-4  D(L.C.) NA ;
001050                          *           'HEX' (C6EA),    'COMMENT'  12-0-8-4/12-0-9-8-4  NA/NA ;
001051                          *           'HEX' (63A2),    'COMMENT'  12-0-3/12-0-9-3  C(L.C.)/NA ;
001052                          *           'HEX' (C5E9),    'COMMENT'  12-0-8-3/12-0-9-8-3  NA/NA ;
001053                          *           'HEX' (62A1),    'COMMENT'  12-0-2/12-0-9-2  B(L.C.)/NA ;
```

```
001054                          *           'HEX' (C4E8),    'COMMENT'  12-0-8-2/12-0-9-8-2  NA/NA ;
001055                          *           'HEX' (61A0),    'COMMENT'  12-0-1/12-0-9-1  A(L.C.)/NA ;
001056                          *           'HEX' (C300),    'COMMENT'  12-0-8-1/12-0-9-8-1  NA/NA ;
001057                          *           'HEX' (7C72),    'COMMENT'  12-11/12-11-9  B5(L.C.)/R(L.C.);
001058                          *           'HEX' (71B0),    'COMMENT'  12-11-8/12-11-9-8  Q(L.C.)/NA ;
001059                          *           'HEX' (70AF),    'COMMENT'  12-11-7/12-11-9-7  P(L.C.)/NA ;
001060                          *           'HEX' (D0F3),    'COMMENT'  12-11-8-7/12-11-9-8-7  NA/NA ;
001061                          *           'HEX' (6FAE),    'COMMENT'  12-11-6/12-11-9-6  O(L.C.)/NA ;
001062                          *           'HEX' (CFF2),    'COMMENT'  12-11-8-6/12-11-9-8-6  NA/NA ;
001063                          *           'HEX' (6EAD),    'COMMENT'  12-11-5/12-11-9-5  N(L.C.)/NA ;
001064                          *           'HEX' (CEF1),    'COMMENT'  12-11-8-5/12-11-9-8-5  NA/NA ;
001065                          *           'HEX' (6DAC),    'COMMENT'  12-11-4/12-11-9-4  M(L.C.)/NA ;
001066                          *           'HEX' (CDF0),    'COMMENT'  12-11-8-4/12-11-9-8-4  NA/NA ;
001067                          *           'HEX' (6CAB),    'COMMENT'  12-11-3/12-11-9-3  L(L.C.)/NA ;
001068                          *           'HEX' (CCEF),    'COMMENT'  12-11-8-3/12-11-9-8-3  NA/NA ;
001069                          *           'HEX' (6BAA),    'COMMENT'  12-11-2/12-11-9-2  K(L.C.)/NA ;
001070                          *           'HEX' (CBEE),    'COMMENT'  12-11-8-2/12-11-9-8-2  NA/NA ;
001071                          *           'HEX' (6AA9),    'COMMENT'  12-11-1/12-11-9-1  J(L.C.)/NA ;
001072                          *           'HEX' (CA10),    'COMMENT'  12-11-8-1/12-11-9-8-1  NA/DLE ;
001073                          *           'HEX' (BAE1),    'COMMENT'  12-11-0/12-11-0-9  NA/NA ;
001074                          *           'HEX' (E0C2),    'COMMENT'  12-11-0-8/12-11-0-9-8  NA/NA ;
001075                          *           'HEX' (DFC1),    'COMMENT'  12-11-0-7/12-11-0-9-7  NA/NA ;
001076                          *           'HEX' (E7FF),    'COMMENT'  12-11-0-8-7/12-11-0-9-8-7  NA/NA ;
001077                          *           'HEX' (DEC0),    'COMMENT'  12-11-0-6/12-11-0-9-6  NA/NA ;
001078                          *           'HEX' (E6FE),    'COMMENT'  12-11-0-8-6/12-11-0-9-8-6  NA/NA ;
001079                          *           'HEX' (DDBF),    'COMMENT'  12-11-0-5/12-11-0-9-5  NA/NA ;
001080                          *           'HEX' (E5FD),    'COMMENT'  12-11-0-8-5/12-11-0-9-8-5  NA/NA ;
001081                          *           'HEX' (DCBE),    'COMMENT'  12-11-0-4/12-11-0-9-4  NA/NA ;
001082                          *           'HEX' (E4FC),    'COMMENT'  12-11-0-8-4/12-11-0-9-8-4  NA/NA ;
001083                          *           'HEX' (DBBD),    'COMMENT'  12-11-0-3/12-11-0-9-3  NA/NA ;
001084                          *           'HEX' (E3FB),    'COMMENT'  12-11-0-8-3/12-11-0-9-8-3  NA/NA ;
001085                          *           'HEX' (DABC),    'COMMENT'  12-11-0-2/12-11-0-9-2  NA/NA;
001086                          *           'HEX' (E2FA),    'COMMENT'  12-11-0-8-2/12-11-0-9-8-2  NA/NA ;
001087                          *           'HEX' (D9BB),    'COMMENT'  12-11-0-1/12-11-0-9-1  NA/NA ;
001088                          *           'HEX' (D890);    'COMMENT'  12-11-0-8-1/12-11-0-9-8-1  NA/NA  ;
```

```
001089                                  /;
001090                                  **;
001091                                  ** TABLE OF FUNCTION CODES;
001092                                  **;
001093                                  *       'TABLE' IO WORD  @17,1@
001094                                  *            @OUTPUT CONTROL          'INTEGER' 0 ;
001095                                  *             INPUT INTERRUPT         'INTEGER' 1 ;
001096                                  *             OUTPUT INTERRUPT        'INTEGER' 2 ;
001097                                  *             INPUT FIRMWARE REV      'INTEGER' 3 ;
001098                                  *             INPUT TASK              'INTEGER' 4 ;
001099                                  *             OUTPUT TASK             'INTEGER' 5 ;
001100                                  *             INPUT MEMORY BYTE       'INTEGER' 6 ;
001101                                  *             READ                    'INTEGER' 7 ;
001102                                  *             INPUT MODULE ADDRESS    'INTEGER' 8 ;
001103                                  *             INPUT RANGE             'INTEGER' 9 ;
001104                                  *             INPUT CONFIGURATION     'INTEGER' 10;
001105                                  *             OUTPUT CONFIGURATION    'INTEGER' 11;
001106                                  *             INPUT STATUS WORD       'INTEGER' 12;
001107                                  *             INPUT DEVICE ID         'INTEGER' 13;
001108                                  *             GET WAIT ADDRESS        'INTEGER' 14;
001109                                  *             OUTPUT DATA             'INTEGER' 15;
001110                                  *             PUNCH                   'INTEGER' 16@;=
001111                                  *             1,2,3,4,6,7,8,9,10,12,16,17,24,38,62,63,73;
001112                                  ************************************************************
001113                                  *       *******;
001114                                  ** SWITCH TO INTERPERT RESPONSE TO "NEXT";
001115                                  **;
001116                                  *       'SWITCH' LETTER SWITCH :=
001117                                  *            LETTER A,          'COMMENT' ALL TESTS;
001118                                  *            LETTER B,          'COMMENT' ELECTRONICS ONLY;
001119                                  *            LETTER C,          'COMMENT' DEVICE CONTROLS;
001120                                  *            LETTER D,          'COMMENT' DEVICE TEST ONLY;
001121                                  *            INVALID INPUT,     'COMMENT' E;
001122                                  *            LETTER F,          'COMMENT' PUNCH AT 400 CPM;
001123                                  *            INVALID INPUT,     'COMMENT' G;
001124                                  *            INVALID INPUT,     'COMMENT' H;
001125                                  *            LETTER I,          'COMMENT' INTERRUPT TEST;
001126                                  *            INVALID INPUT,     'COMMENT' J;
001127                                  *            LETTER K,          'COMMENT' READ/PUNCH;
001128                                  *            LETTER L,          'COMMENT' TEST MODE + DATA LOOPBACK;
001129                                  *            LETTER M,          'COMMENT' PUNCH ASCII;
001130                                  *            LETTER N,          'COMMENT' READ ASCII;
001131                                  *            INVALID INPUT,     'COMMENT' O;
001132                                  *            LETTER P,          'COMMENT' PUNCH RANDOM TEST DECK;
001133                                  *            INVALID INPUT,     'COMMENT' Q;
001134                                  *            LETTER R,          'COMMENT' READ ANY DECK;
001135                                  *            FRESH START,       'COMMENT' START/RESTART PROGRAM;
001136                                  *            INVALID INPUT,     'COMMENT' T;
001137                                  *            INVALID INPUT,     'COMMENT' U;
001138                                  *            LETTER V,          'COMMENT' VERIFY ANY DECK;
001139                                  *            INVALID INPUT,     'COMMENT' W;
001140                                  *            INVALID INPUT,     'COMMENT' X;
001141                                  *            INVALID INPUT,     'COMMENT' Y;
001142                                  *            LETTER Z ;         'COMMENT' PATCH ROUTINE;
001143                                  *       FRESH START;
001144  004F  D3C0 0000      P     FRESHS    LNJ         5,KJFS
001145                                  *            FRESH START P;
```

```
001146                                  /;
001147                                  ** ASK "NEXT ?;" OF OPERATOR AND EXPECT A RESPONSE OF THE FORM;
001148                                  **;
001149                                  **       C/R ONLY              USE THE MOST RECENT SELECTION AGAI
001150                                  *      N;
001151                                  **       LETTER AND C/R       MAKE NEW SELECTION;
001152                                  **;
001153                                  *      ASK NEXT;
001154          0051                    ASKNEX   EQU      $
001155  0051    0005                             DC       5
001156                                  *           CLOCK OFF; (WHO KNOWS WHAT WAS BEING DONE BEFORE)
001157                                  *            GET BREAK;
001158  0052    D3C0 0000       P                LNJ       5,VKGB
001159  0054    FBC0 FFD1                        LAB       7,D+38
001160  0056    9BC0 FFB4                        LAB       1,D+11
001161  0058    9F87                             STB       1,$B7
001162                                  *           INOUT (INITIALIZE, OUTPUT CONTROL @0@);
001163  0059    ABC0 029B                        LAB       2,A+524
001164  005B    AFC0 FFCB                        STB       2,D+39
001165  005D    D3C0 0000       P                LNJ       5,JIO
001166                                  *           ASK NEW QUESTION ("NEXT$");
001167  005F    FBC0 FFC6                        LAB       7,D+38
001168  0061    9BC0 FFE1                        LAB       1,T29
001169  0063    9F87                             STB       1,$B7
001170  0064    D3C0 0000       P                LNJ       5,VKANQ
001171                                  *           TEMP CHARACTER := TEST TYPE * 256;
001172  0066    E840 FF9E                        LDR       6,D+5
001173  0068    6008                             SOL       6,8
001174  0069    EF40 FFB1                        STR       6,D+27
001175  006B    FBC0 FFBA                        LAB       7,D+38
001176  006D    9BC0 FFAD                        LAB       1,D+27
001177  006F    9F87                             STB       1,$B7
001178  0070    D3C0 0000       P                LNJ       5,VKGC
001179                                  *           GET CHAR (TEMP CHARACTER );
001180                                  *           'IF' TEMP CHARACTER 'EQ' 'HEX' (0D)
001181                                  *           'THEN'
001182                                  *                'GO TO' KEEP OLD SELECTION;
001183  0072    E840 FFA8                        LDR       6,D+27
001184  0074    6D0D                             CMV       6,13
001185  0075    0907                             BE        >L31
001186                                  *           'IF' TEMP CHARACTER 'GE' 'LITERAL' (A) 'AND'
001187  0076    6D41                             CMV       6,65
001188  0077    080C                             BAL       >L32
001189                                  *              TEMP CHARACTER 'LE' 'LITERAL' (Z)
001190                                  *           'THEN'
001191                                  *           'BEGIN'
001192  0078    6D5A                             CMV       6,90
001193  0079    0A0A                             BAG       >L32
001194                                  *              TEST TYPE := TEMP CHARACTER;
001195                                  *      KEEP OLD SELECTION;
001196  007A    EF40 FF8A                        STR       6,D+5
001197          007C                    L31      EQU       $
001198                                  *                  'GO TO' LETTER SWITCH @('BITS' @5,0@ TEST TYPE)@;
001199  007C    9840 FF88                        LDR       1,D+5
001200  007E    9570 001F                        AND       1,=Z'001F'
001201  0080    9C90 0028                        LDB       1,<S1.$R1
001202  0082    8381                             JMP       $B1
001203                                  *           'END';
001204                                  *      INVALID INPUT;
001205          0083                    L32      EQU       $
001206                                  *           PUT NEW TEXT ("INVALID$");
001207  0083    FBC0 FFA2                        LAB       7,D+38
001208  0085    9BC0 FFC1                        LAB       1,T33
001209  0087    9F87                             STB       1,$B7
001210  0088    D3C0 0000       P                LNJ       5,VKPNT
001211                                  *           'GO TO' ASK NEXT;
001212                                  **;
001213                                  ** LETTER A - ALL TESTS;
001214                                  **;
001215                                  *      LETTER A;
001216  008A    83C0 FFC6                        JMP       ASKNEX
001217          008C                    L2       EQU       $
001218                                  *           HAVE GENERATED CHECKSUMS := FALSE;
001219                                  **;
001220                                  ** LETTER B - ELECTRONICS ALONE;
001221                                  **;
001222                                  *      LETTER B;
001223  008C    8740 FF8C                        CL        D+25
001224                                  *           LETTER BP;
001225                                  **;
001226                                  **  LETTER I - INTERRUPT TEST;
001227                                  **;
001228                                  *      LETTER I;
001229  008E    D3C0 0000       P       L3       LNJ       5,KJLB
001230                                  *           LETTER IP;
001231  0090    D3C0 0000       P       L8       LNJ       5,KJLI
001232                                  *           'IF' TEST TYPE 'EQ' 'LITERAL' (A)
001233  0092    E840 FF72                        LDR       6,D+5
001234  0094    6D41                             CMV       6,65
001235  0095    0905                             BE        >L34
001236                                  *           'OR' TEST TYPE 'EQ' 'LITERAL' (B)
001237                                  *           'THEN'
001238                                  *                'GO TO' LETTER L
001239                                  *           'ELSE'
001240  0096    6D42                             CMV       6,66
001241  0097    0903                             BE        >L34
001242                                  *                'GO TO' ASK NEXT;
001243                                  **;
001244                                  ** LETTER L - TEST MODE + DATA LOOPBACK;
001245                                  **;
001246                                  *      LETTER L;
001247  0098    83C0 FFB8                        JMP       ASKNEX
001248                                  *           LETTER LP;
001249  009A    D3C0 0000       P       L34      LNJ       5,KJLL
001250                                  *           'IF' TEST TYPE 'EQ' 'LITERAL' (A)
001251                                  *           'THEN'
001252                                  *                'GO TO' LETTER C
001253                                  *           'ELSE'
001254  009C    E840 FF68                        LDR       6,D+5
001255  009E    6D41                             CMV       6,65
001256  009F    0903                             BE        >L5
001257                                  *                'GO TO' ASK NEXT;
001258                                  **;
```

```
001259                                  ** LETTER D - DEVICE TESTS ONLY;
001260                                  **;
001261                                  *       LETTER D;
001262  00A0  83C0 FFB0                    JMP     ASKNEX
001263                                  **;
001264                                  ** LETTER C - SAFETY  FEATURES;
001265                                  **;
001266                                  *       LETTER C;
001267                                  *           LETTER CP;
001268  00A2  D3C0 0000       P     L5      LNJ     5,KJLC
001269  00A4  E840 FF60                     LDR     6,D+5
001270  00A6  6D41                         CMV     6,65
001271  00A7  0905                         BE      >L36
001272                                  *           'IF' TEST TYPE 'EQ' 'LITERAL' (A) 'OR' TEST TYPE 'EQ' 'LITERAL' (D)
001273                                  *           'THEN'
001274                                  *               'GO TO' LETTER M
001275                                  *           'ELSE'
001276  00A8  6D44                         CMV     6,68
001277  00A9  0903                         BE      >L36
001278                                  *               'GO TO' ASK NEXT;
001279                                  **;
001280                                  ** LETTER M - PUNCH ASCII DECK;
001281                                  **;
001282                                  *       LETTER M;
001283  00AA  83C0 FFA6                    JMP     ASKNEX
001284                                  *           LETTER MP;
001285  00AC  D3C0 0000       P     L36     LNJ     5,KJLM
001286                                  *           'IF' TEST TYPE 'EQ' 'LITERAL' (A) 'OR'
001287  00AE  E840 FF56                     LDR     6,D+5
001288  00B0  6D41                         CMV     6,65
001289  00B1  0905                         BE      >L37
001290                                  *               TEST TYPE 'EQ' 'LITERAL' (D)
001291                                  *           'THEN'
001292                                  *               'GO TO' LETTER N
001293                                  *           'ELSE'
001294  00B2  6D44                         CMV     6,68
001295  00B3  0903                         BE      >L37
001296                                  *               'GO TO' ASK NEXT;
001297                                  **;
001298                                  ** LETTER N - READ ASCII DECK;
001299                                  **;
001300                                  *       LETTER N;
001301  00B4  83C0 FF9C                    JMP     ASKNEX
001302                                  *           LETTER NP;
001303  00B6  D3C0 0000       P     L37     LNJ     5,KJLN
001304                                  *           'IF' TEST TYPE 'EQ' 'LITERAL'(A) 'OR'
001305  00B8  E840 FF4C                     LDR     6,D+5
001306  00BA  6D41                         CMV     6,65
001307  00BB  0905                         BE      >L38
001308                                  *               TEST TYPE 'EQ' 'LITERAL'(D)
001309                                  *           'THEN'
001310                                  *               'GO TO' LETTER P
001311                                  *           'ELSE'
001312  00BC  6D44                         CMV     6,68
001313  00BD  0903                         BE      >L38
001314                                  *               'GO TO' ASK NEXT;
001315                                  **;
001316                                  ** LETTER P - PUNCH RANDOM OR INPUTTED DECK;
001317                                  **;
001318                                  *       LETTER P;
001319  00BE  83C0 FF92                    JMP     ASKNEX
001320                                  *           LETTER PP;
001321  00C0  D3C0 0000       P     L38     LNJ     5,KJLP
001322  00C2  E840 FF42                     LDR     6,D+5
001323  00C4  6D41                         CMV     6,65
001324  00C5  0910                         BE      >L39
001325                                  *           'IF' TEST TYPE 'EQ' 'LITERAL' (A) 'OR' TEST TYPE 'EQ' 'LITERAL' (D)
001326                                  *           'THEN'
001327                                  *               'GO TO' LETTER F
001328                                  *           'ELSE'
001329  00C6  6D44                         CMV     6,68
001330  00C7  090E                         BE      >L39
001331                                  *               'GO TO' ASK NEXT;
001332                                  **;
001333                                  ** LETTER R + V   READ ANY DECK IN R OR V MODE;
001334                                  **;
001335                                  *       LETTER R;
001336  00C8  83C0 FF88                    JMP     ASKNEX
001337        00CA                      LETR    EQU     $
001338                                  *           R OR V MODE := 'LITERAL'(R);
001339  00CA  6C52                         LDV     6,82
001340  00CB  EF40 FF4B                    STR     6,D+23
001341                                  *               'GO TO' KEEP R MODE;
001342                                  *       LETTER V;
001343  00CD  0F84                         B       >L40
001344        00CE                      LETV    EQU     $
001345                                  *           R OR V MODE := 'LITERAL' (V);
001346                                  *       KEEP R MODE;
001347  00CE  6C56                         LDV     6,86
001348  00CF  EF40 FF47                    STR     6,D+23
001349                                  *           LETTER RP;
001350  00D1  D3C0 0000       P     L40     LNJ     5,KJLR
001351                                  *               'GO TO' ASK NEXT;
001352                                  **;
001353                                  ** LETTER F - PUNCH AT 400 CPM;
001354                                  **;
001355                                  *       LETTER F;
001356  00D3  83C0 FF7D                    JMP     ASKNEX
001357                                  *           LETTER FP;
001358  00D5  D3C0 0000       P     L39     LNJ     5,KJLF
001359  00D7  E840 FF2D                     LDR     6,D+5
001360  00D9  6D41                         CMV     6,65
001361  00DA  0905                         BE      >L41
001362                                  *           'IF' TEST TYPE 'EQ' 'LITERAL' (A) 'OR' TEST TYPE 'EQ' 'LITERAL' (D)
001363                                  *           'THEN'
001364                                  *               'GO TO' LETTER K;
001365  00DB  6D44                         CMV     6,68
001366  00DC  0903                         BE      >L41
001367                                  *               'GO TO' ASK NEXT;
001368                                  **;
001369                                  ** LETTER K - READ AND PUNCH;
001370                                  **;
001371                                  *       LETTER K;
```

```
001372  00DD  83C0 FF73                        JMP       ASKNEX
001373                              *           LETTER KP;
001374  00DF  D3C0 0000       P     L41       LNJ       5,KJLK
001375                              *             'GO TO'ASK NEXT;
001376                              **;
001377                              **LETTER Z - PATCH ROUTINE;
001378                              **;
001379                              *         LETTER Z;
001380  00E1  83C0 FF6F                        JMP       ASKNEX
001381        00E3                  L17       EQU       $
001382                              *             'CODE' 'BEGIN'
001383                              *                   CALL ZV$PCH
001384                              *                   XLOC ZV$PCH
001385                              *             ]'END';
001386                                        CALL      ZV$PCH
        00E3  FBF0 0001
        00E5  D380 0000       X
001387                                        XLOC      ZV$PCH
001388                              *             'GO TO' ASK NEXT;
001389                              *         'END';
001390                              *         'FINISH'
001391  00E7  83C0 FF69                        JMP       ASKNEX
001392  00E9  0000                  A         RESV      541,0
001393  0007                        0         ORG       D+7
001394  0007  1300                            DC        Z'1300'
001395  0008                                  ORG       D+8
001396  0008  4400                            DC        Z'4400'
001397  0009                                  ORG       D+9
001398  0009  0000                            DC        Z'0000'
001399  000A                                  ORG       D+10
001400  000A  4000                            DC        Z'4000'
001401  000B                                  ORG       D+11
001402  000B  8000                            DC        Z'8000'
001403  0010                                  ORG       D+16
001404  0010  0018                            DC        Z'0018'
001405  0012                                  ORG       D+18
001406  0012  0064                            DC        Z'0064'
001407  001C                                  ORG       D+28
001408  001C  4018                            DC        Z'4018'
001409  001D                                  ORG       D+29
001410  001D  2000                            DC        Z'2000'
001411  001E                                  ORG       D+30
001412  001E  0000                            DC        Z'0000'
001413  001F                                  ORG       D+31
001414  001F  0500                            DC        Z'0500'
001415  0146                                  ORG       A+93
001416  0146  0009 A0C0 9E40                  DC        Z'0009A0C09E40980121008BEAB200990088EB'
              9801 2100 88EA
              B200 9900 88EB
001417  014F  B200                            DC        Z'B200'
001418  0153                                  ORG       A+106
001419  0153  0006 A0C0 9C40                  DC        Z'0006A0C09C40980188EAB2002100'
              9801 88EA B200
              2100
001420  015D                                  ORG       A+116
001421  015D  0010 9848 8100                  DC        Z'0010984881002034 88EB716A88EA716A0000'
              2034 88EB 716A
              88EA 716A 0000
001422  0166  8180 2034 88EB                  DC        Z'8180203488EBB200210088EAB2000008'
              B200 2100 88EA
              B200 0008
001423  0171                                  ORG       A+136
001424  0171  000F 9840 8100                  DC        Z'000F984081002034 88EA716A000984888EA'
              2034 88EA 716A
              0000 9848 88EA
001425  017A  B200 2100 8180                  DC        Z'B200210081802034 88EBB2000008'
              2034 88EB B200
              0008
001426  0183                                  ORG       A+154
001427  0183  000E 9848 8100                  DC        Z'000E984881002034 88EB716A88EA716A0000'
              2034 88EB 716A
              88EA 716A 0000
001428  018C  9840 8180 2034                  DC        Z'9840818020348 8EAB2000008'
              88EA B200 0008
001429  0195                                  ORG       A+172
001430  0195  0005 0008 8180                  DC        Z'0005000881802034818020034'
              2034 8180 2034
001431  0249                                  ORG       A+352
001432  0249  3031 3233 3435                  DC        Z'3031323334353637383941424344454647 48'
              3637 3839 4142
              4344 4546 4748
001433  0252  494A 4B4C 4D4E                  DC        Z'494A4B4C4D4E4F505152535455565758595A3A2340273D22'
              4F50 5152 5354
              5556 5758 595A
              3A23 4027 3D22
001434  025E  202F 5C2C                      DC        Z'202F5C2C'
001435  0260  255F 3E3F 2D5D                  DC        Z'255F3E3F2D5D242A295E265B2E3C282B217D7B2020202020'
              242A 295E 265B
              2E3C 282B 217D
              7B20 2020 2020
001436  026C  2020 2020                      DC        Z'20202020'
001437  026E  2020 2020 2020                DC        Z'2020202020 20'
001438  0271                                  ORG       A+392
001439  0271  020A 0180                      DC        Z'020A0180'
001440  0274                                  ORG       A+395
001441  0274  2039 3898 3704                  DC        Z'2039389837042 21A36963D9E3595271534 94'
              221A 3696 3D9E
              3595 2715 3494
001442  027D  4014 3393 239B                  DC        Z'4014339323 9B32163A9A3191609930 5A5988581B3F075717'
              3216 3A9A 3191
              6099 305A 5988
              581B 3F07 5717
001443  0289  3E06 560A                      DC        Z'3E06560A'
001444  028B  5F05 5584 258C                  DC        Z'5F055584258C54832C8B53825C8A2F81B9892D5251185087'
              5483 2C8B 5382
              5C8A 2F81 B989
              2D52 5118 5087
001445  0297  5E1F 4F08                      DC        Z'5E1F4F08'
001446  0299  3B1E 4E85 291D                  DC        Z'3B1E4E85291D4D9D2A1C4C13248F4B125D924A11B1197D7A'
              4D9D 2A1C 4C13
              248F 4B12 5D92
              4A11 B119 7D7A
001447  02A5  79B8 78B7                      DC        Z'79B878B7'
```

```
001446  02A7  D7F9 77B6 D6F8                    DC      Z'D7F977B6D6F876B5D5F775B4D4F674B3D3F573B2D2F47E9F'
                76B5 D5F7 75B4
                D4F6 74B3 D3F5
                73B2 D2F4 7E9F
001449  02B3  D180 2649                         DC      Z'D18026491
001450  02B5  4897 477F 210F                    DC      Z'4897477F210F46864B0E450928UD449C3C0C43032E0B4202'
                4686 2B0E 4509
                280D 449C 3C0C
                4303 2E0B 4202
001451  02C1  5B8E 4101                         DC      Z'5B8E4101'
001452  02C3  A88D 7B69 68A7                    DC      Z'A88D736968A767A6C9ED66A5C8EC65A4C7EB64A3C6EA63A2'
                67A6 C9ED 66A5
                C8EC 65A4 C7EB
                64A3 C6EA 63A2
001453  02CF  C5E9 62A1                         DC      Z'C5E962A1'
001454  02D1  C4E8 61A0 C300                    DC      Z'C4E861A0C3007C727180708FD0F36FAECFF26EADCEF16DAC'
                7C72 7180 708F
                D0F3 6FAE CFF2
                6EAD CEF1 6DAC
001455  02DD  CDF0 6CAB                         DC      Z'CDF06CAB'
001456  02DF  CCEF 6BAA CBEE                    DC      Z'CCEF6BAACBEE6AA9CA108AE1E0C2DFC1E7FFDEC0E6FEDDBF'
                6AA9 CA10 BAE1
                E0C2 DFC1 E7FF
                DEC0 E6FE DDBF
001457  02EB  E5FD DCBE                         DC      Z'E5FDDCBE'
001458  02ED  E4FC DBBD E3FB                    DC      Z'E4FCDBBDE3FBDABCE2FAD9BBD890'
                DABC E2FA D9BB
                D890
001459  02F5                                    ORG     A+524
001460  02F5  0001 0002 0003                    DC      Z'0001000200030004000600070008000900UA'
                0004 0006 0007
                0008 0009 000A
001461  02FE  000C 0010 0011                    DC      Z'000C00100011001800260036003F0049'
                0018 0026 003E
                003F 0049
001462  0306                                    ORG     0
001463                                          XLOC    VKPNT
001464                                          XLOC    VKANQ
001465                                          XLOC    VKGC
001466                                          XLOC    VKGB
001467                                          XLOC    J10
001468                                          XLOC    KJLB
001469                                          XLOC    KJLC
001470                                          XLOC    KJLF
001471                                          XLOC    KJLI
001472                                          XLOC    KJLK
001473                                          XLOC    KJLL
001474                                          XLOC    KJLM
001475                                          XLOC    KJLN
001476                                          XLOC    KJLP
001477                                          XLOC    KJLR
001478                                          XLOC    KJFS
001479  0306  004C                              END     CARD,CARD
0000  ERR COUNT
@L
```

```
E    ZVKJIB.L                    05/22/78  1507.5R W 05/22/78  1344.6      1028250000
 CORCOP REL 0200              SOURCE PROGRAM : ZVKJIB.K                    PAGE 01     1978/04/27 0756:49.65  @
                1         2         3         4         5         6         7         8@
LINE  12345678901234567890123456789012345678901234567890123456789012345678901234567890@@
@
   1  ********************************************************************************
   2  *
   3  *   CARD EQUIPMENT T&V LIBRARY - ZVKJIB
   4  *
   5  *   22 SEPT 77
   6  *
   7  * GLOBAL VARIABLES AND CONSTANTS
   8  *
   9  'EXTERNAL' (
  10       'INTEGER' ID,
  11                 STAT WAS,
  12                 STATUS SHOULD BE,
  13                 HERTZ,
  14                 CHANNEL,
  15                 PUESDO CARD STATUS,
  16                 TEST TYPE,
  17                 TYPE OF PUNCHES,
  18                 SET TO HOLLERITH,
  19                 SET TO ASCII,
```

```
                ASCII ERROR EJECT / ASEJCT,
                BINARY MODE,
                READ STATUS,
                NUMBER OF COLUMNS,
                LABEL1,
                LABEL2,
                MAXIMUM CARDS,
                CARDS SPECIFIED,
                CARDS READ,
                BIT COUNT,
                LOW ADDRESS,
                R OR V MODE,
                HAVE GENERATED CHECKSUMS,
                SPEED,
                TIME1,
                ERROR FLAG,
                CLOCK SEMAPHORE / CKSEM,
                L21   SEMAPHORE,
                DEVICELEVEL / DLVL,
                DEVICESEMAPHORE / DVSEM,
                CPLEVEL,
                LEVTYPE,
                TEMP CHARACTER,
                SREG,
                BASE,
                SUM,
                TOTAL,
                RESPONSE,
                SET CHANNEL READY          /  SCR,
                SET TEST MODE              /  IOTM,
                WAIT LOOP ADDRESS          /  IOX1,
                RETURN FROM TEST MODE      /  IOX2,
                ZHIAFB / ZHIAFB,
                ZVDTTY / ZV$TTY,
                READ OR PUNCH;);
*
* GLOBAL ARRAYS - BUFFERS
*
'EXTERNAL' (
     'INTEGER' 'ARRAY' CHECKSUM @0:6@,
                       BINCOUNT @0:12@,
                       ASCCNT   @0:9@,
                       BINOB    @0:19@,
                       PPROMS   @0:17@,
                       RPROMS   @0:17@,
                       ONE SHOT @0:8@,
                       WORD CONVERSION TABLE @0:126@,
                       BUFFER1 / BUFF1 @0:83@,
                       BUFFER2 / BUFF2 @0:79@,
                       CHAR     @0:39@,
                       ILLEGAL ASCII @0;2@;
                                          );
/
*
* GLOBAL INTERRUPT SAVE AREAS / VARIABLES
*
'EXTERNAL' (
'TABLE' ISA FOR CLOCK / ISAC @31,1@
     @ISACD                  'INTEGER' 2;
      ISACP                  'INTEGER' 5@;
'TABLE' ISA FOR DEVICE / ISAD @31,1@
     @ISARD                  'INTEGER' 2;
      ISARP                  'INTEGER' 5@;
'TABLE' ISA FOR PROGRAM / ISAP @31,1@
     @ISATD                  'INTEGER' 2;
      ISATP                  'INTEGER' 5@;
     ); (END OF INTERRUPT EXTERNALS)
*
* GLOBAL TABLE OF CONTROL WORDS
*
'EXTERNAL' (
'TABLE' IO WORD  @17,1@
     @OUTPUT CONTROL         'INTEGER' 0 ;
      INPUT INTERRUPT        'INTEGER' 1 ;
      OUTPUT INTERRUPT       'INTEGER' 2 ;
      INPUT FIRMWARE REV     'INTEGER' 3 ;
      INPUT TASK             'INTEGER' 4 ;
      OUTPUT TASK            'INTEGER' 5 ;
      INPUT MEMORY BYTE      'INTEGER' 6 ;
      READ                   'INTEGER' 7 ;
      INPUT MODULE ADDRESS   'INTEGER' 8 ;
      INPUT RANGE            'INTEGER' 9 ;
      INPUT CONFIGURATION    'INTEGER' 10;
      OUTPUT CONFIGURATION   'INTEGER' 11;
      INPUT STATUS WORD      'INTEGER' 12;
      INPUT DEVICE ID        'INTEGER' 13;
      GET WAIT ADDRESS       'INTEGER' 14;
      OUTPUT DATA            'INTEGER' 15;
      PUNCH                  'INTEGER' 16@;);
*
* GLOBAL DEFINITIONS
*
*
'DEFINE' CHECK STATUS FOR ERRORS " CHECK STATUS ERRORS ";
'DEFINE'  SHOULD BE ON  " 1 ";
'DEFINE'  SHOULD BE OFF " 0 ";
'DEFINE' CLEAR          " SET TO ASCII ";
'DEFINE' INITIALIZE   " BINARY MODE ";
'DEFINE' CHANNEL NUMBER "CHANNEL ";
'DEFINE' INPUT ID " INPUT DEVICE ID ";
'DEFINE' INPUT STATUS " INPUT STATUS WORD ";
'DEFINE' PUNCH A CARD " READ A CARD ";
'DEFINE' CARD COUNT " CARDS READ ";
'DEFINE' CREATE TEST CHECKSUMS " PUNCH A DECK ";
'DEFINE' ERROR LABEL "LABEL1";
'DEFINE' STATUS WAS  " STAT WAS ";
'DEFINE'  BUFFER   " BUFFER1 ";
*
'DEFINE' CLOCK ON
     "'BEGIN'
          'CODE''BEGIN']
                    DC 4
          ]'END';
```

```
      'END'";
'DEFINE' CLOCK OFF
    "'BEGIN'
          'CODE''BEGIN']
                    DC 5
          ]'END';
    'END'";
*
'DEFINE' LEV (TYPE,LEVEL)
    "'BEGIN'
          LEVTYPE := TYPE 'UNION' LEVEL;
          'CODE' 'BEGIN']
               LEV ]LEVTYPE
          'END';
    'END'";
*
* MACRO USED TO SET AN INTEGER EQUAL TO THE ADDRESS OF AN EXTERNAL LABEL
'DEFINE' ADDRESS OF LABEL (DESTINATION,LABEL)
"'BEGIN'
    'CODE''BEGIN']
          STB $B1,]TEMP CHARACTER]   SAVE B1
          B >+$A                      DON'T EXECUTE JUMP
     ]'END';
     'GO TO' LABEL;       (DUMMY JUMP - GETS ADDRESS ONLY)
     'CODE''BEGIN']
       ORG    $-1-$AF
$E     ORG    $+1                   $E IS THE JUMP
       ]CB]
$B     ORG    $-1+$AF               $B IS THE ENTIRE ADDRESS
$C     ORG    $+1                   $C IS THE DISPLACEMENT
       ]CB]
$A     LB     -$E,=X'40'            DETERMINE IF THE EXPRESSION IS
       BBF    +$D                   A DISPLACEMENT OR AN IMA
       ]CB]
       LAB    $B2,-$C               GET THE ADDRESS OF THE DISPLACEMNET
       LDR    $R1,$B2               GET THE DISPLACEMENT
       ]CB]
       LAB    $B1,$B2,$R1           CALCULATE THE EFFECTIVE ADDRESS
       B      +$F
       ]CB]
$D     LAB    $B1,*-$B              GET THE IMA ADDRESS
$F     STB    $B1,]DESTINATION]+4+$AF     DESTINATION := 'LOCATION' (LABEL);
          ]CB]
          LDB $B1,]TEMP CHARACTER
     'END';
'END'";
*
*  LOAD TWO REGISTERS FOR ERROR DISPLAY
'DEFINE' LOAD R4 R5 (INTO R4, INTO R5)
     "'BEGIN'
          'CODE''BEGIN']
               LDR $R5,]INTO R5]
               LDR $R4,]INTO R4
          'END';
     'END'";
*
*  SAVE THE RETURN ADDRESS (CALLING ROUTINE ADDRESS)
'DEFINE' SAVE RETURN ADDRESS
"'INTEGER' RETURN ADDRESS;
'CODE''BEGIN']
                    XLOC ZV$ER
          STB $B5,]RETURN ADDRESS
     'END'";
*
*  SAVE IN REGISTERS INFORMATION FOR ERROR HALT
'DEFINE' LOAD R4 R5 AND REPORT ERROR (INTO R4, INTO R5)
     "'BEGIN'
          'CODE''BEGIN']
               LDR $R4,]INTO R4]
               LDR $R5,]INTO R5]
               ]CB]
                LDB $B5,]RETURN ADDRESS]
                STB $B5,+$A        ADDRESS OF CALLING ROUTINE
                ]CB]
                LNJ $B5,<ZV$ER     REPORT ERROR
                ]CB]
                B >+$B
$A              DC <$
                ]CB]
                DC <]ERROR LABEL]
$B              ORG $]
          'END';
     'END'";
*
* DECLARATION OF EXTERNAL PROCEDURES - SECONDARY PROCEDURES USED BY TEST
*
'EXTERNAL'(
     'PROCEDURE' GENERATE TEST CHECKSUMS /   JGT;
     'PROCEDURE' GENERATE CHECKSUMS      /   JGC    ('VALUE' 'INTEGER');
     'PROCEDURE' REPORT ERROR            /   JRE;
     'PROCEDURE' INOUT                   /   JIO ('LOCATION' 'INTEGER',
                                                  'LOCATION' 'INTEGER');
     'PROCEDURE' IOLD                    /   JIL ('LOCATION' 'INTEGER',
                                                  'LOCATION' 'INTEGER',
                                                    'VALUE' 'INTEGER');
     'PROCEDURE' PUT NUMBER              /   JPN    ('VALUE' 'INTEGER');
     'PROCEDURE' COMPARE ASCII           /   JCA    ('VALUE' 'INTEGER',
                                                     'VALUE' 'INTEGER',
                                                     'VALUE' 'INTEGER');
     'PROCEDURE' WAIT FOR RETURN         /   JWR;
     'PROCEDURE' CHECK STATUS ERRORS     /   JSE;
     'PROCEDURE' CHECK STATE             /   JCS    ('VALUE' 'INTEGER');
     'PROCEDURE' CREATE FUNCTION CODES   /   JFC    ('VALUE' 'INTEGER');
     'PROCEDURE' CALCULATE ELAPSED TIME  /   JET    ('VALUE' 'INTEGER');
     'PROCEDURE' COMPARE CHECKSUMS       /   JCC;
     'PROCEDURE' FILL BUFFER             /   JFB    ('INTEGER' 'ARRAY',
                                                     'VALUE' 'INTEGER',
                                                     'VALUE' 'INTEGER');
     'PROCEDURE' PUNCH A DECK            /   JPD    ('VALUE' 'INTEGER',
                                                     'VALUE' 'INTEGER');
     'PROCEDURE' MICRO                   /   JMC    ('INTEGER' 'ARRAY');
     'PROCEDURE' TIME DELAY              /   JTD;
     );  (END OF SECONDARY PROCEDURES)
```

```
246   *
247   * DECLARATION OF EXTERNAL PROCEDURES - RESPONSES TO NEXT?
248   *
249   'EXTERNAL' (
250         'PROCEDURE' LETTER BP       /   KJLB;
251         'PROCEDURE' LETTER CP       /   KJLC;
252         'PROCEDURE' LETTER FP       /   KJLF;
253         'PROCEDURE' LETTER IP       /   KJLI;
254         'PROCEDURE' LETTER KP       /   KJLK;
255         'PROCEDURE' LETTER LP       /   KJLL;
256         'PROCEDURE' LETTER MP       /   KJLM;
257         'PROCEDURE' LETTER NP       /   KJLN;
258         'PROCEDURE' LETTER PP       /   KJLP;
259         'PROCEDURE' LETTER RP       /   KJLR;
260         'PROCEDURE' FRESH START P   /   KJFS;);
261   *
262   * DECLARATION OF EXTERNAL LABELS
263   *
264   'EXTERNAL' (
265         'LABEL'     FRESH START,
266                     TIMER,
267                     CP RUPT,
268                     DEVICEHANDLER / DEVIH,
269                     LEV20,
270                     LEV21,
271                     LETTER R / LETR,
272                     LETTER V / LETV,
273                     ASK NEXT ;);
eeL
```

```
E    ZVKGB.LIST                          05/19/78  1422.5R W 05/19/78  1422.6       286110000
     000001                                         TITLE     ZVKGB,'REV 00'
     000002          0016                           XDEF      VKGB
     000003          0012                           XDEF      ZVKGB
     000004          0008                ZKCOM      COMM      8
     000005  0000   0000                 D          RESV      3,0
     000006  0003   001B                 T4         DC        27
     000007  0004   2A2A 2A2A 2A20                  TEXT      '***** BREAK DETECT'
             0007   4252 4541 4B20
                    4445 5445 4354
     000008  000D   4544 202A 2A2A                  TEXT      'ED *****$'
             0010   2A2A 2400
     000009          0012                ZVKGB      EQU       $
     000010  0012   8F00 0001       K              SAVE      <ZKCOM+1,Z'0011'
             0014   0011
     000011                                         *       'INTEGER' 'PROCEDURE' GET BREAK;
     000012                                         *       'BEGIN'
     000013  0015   0F94                            B         >L1
     000014  0016   DFC0 FFE9            VKGB       STB       5,D
     000015                                         *         ZVDBRK; (CALL ZV$BRK)
     000016  0018   D3C0 0000       P              LNJ       5,ZV$BRK
     000017                                         *           'IF' ZVDBKF 'EQ' 1
     000018                                         *             'THEN'
     000019  001A   E840 0000       P              LDR       6,ZV$BKF
     000020  001C   6D01                           CMV       6,1
     000021  001D   0988                           BNE       >L3
     000022                                         *                   PUT NEW TEXT ("***** BREAK DETECTED *****$");
     000023  001E   FBC0 FFE2                       LAB       7,D+1
     000024  0020   9BC0 FFE2                       LAB       1,T4
     000025  0022   9F87                           STB       1,$B7
     000026  0023   D3C0 0000       P              LNJ       5,VKPNT
     000027  0025   E840 0000       P       L3     LDR       6,ZV$BKF
     000028  0027   83C8 FFD8                      JMP       *D
     000029                                         *       'ANSWER' ZVDBKF; (RETURN BREAK STATUS)
     000030                                         *       'END';
     000031                                         *       'END'
     000032                                         *       'FINISH'
     000033          0029                L1         EQU       $
     000034  0029   83C0 0000       P              JMP       ZK$EX
     000035                                         XLOC      ZK$EX
     000036                                         CTRL LINK ZK$EX
     000037                                         XLOC      ZV$BRK
     000038                                         CTRL LINK ZV$BR
     000039                                         XLOC      ZV$BKF
     000040                                         CTRL LINK ZV$BK
     000041                                         XLOC      VKPNT
     000042  002B   0012                           END       ZVKGB,ZVKGB
0000  ERR COUNT
L
```

```
E    ZVKGC.LIST                          05/19/78  1422.8R W 05/19/78  1422.8        350910000
     000001                                          TITLE     ZVKGC,'REV 00'
     000002          0009                            XDEF      VKGC
     000003          0004                            XDEF      ZVKGC
     000004          0008              ZKCOM         COMM      8
     000005  0000    0000              D             RESV      4,0
     000006          0004              ZVKGC         EQU       $
     000007  0004    8F00 0001    K                  SAVE      <ZKCOM+1,Z'0011'
             0006    0011
     000008  0007    0F81 001C                       B         L1
     000009                            *         'PROCEDURE' GET CHAR ('LOCATION' 'INTEGER' TARGET);
     000010                            *         'BEGIN'
     000011                            *             'INTEGER' A;
     000012  0009    DFC0 FFF6                       STB       5,D
     000013  000B    E807                            LDR       6,$B7
     000014  000C    EF40 FFF4                       STR       6,D+1
     000015                            *                 'CODE' 'BEGIN'
     000016                            *                       STR  $R6,+$A              FUNNY, THERE IS AN ADDRESS IN R REG
     000017                            *                      XLOC ZV$IA
     000018  000E    EF40 0000    I                  STR       $R6,+$A
     000019                                          XLOC      ZV$IA
     000020                            *                     ]CB]
     000021                            *                      LDV  $R1,4        TWO BYTE INPUT
     000022  0010    1C04                            LDV       $R1,4
     000023  0011    D840 0000    P                  LDR       $R5,ZV$TTY
     000024                            *                     ]CB]
     000025                            *                     BNEZ $R5,+$E
     000026                            *                     ADV  $R1,-2     YES, DONT LOOK FOR C/R
     000027  0013    5981 0000    T                  BNEZ      $R5,+$E
     000028  0015    1EFE                            ADV       $R1,-2
     000029                            *                     ]CB]
     000030  0016    9F40 FFEB                  $E     STR       $R1,D+2
     000031  0018    D3C0 0000    P                  LNJ       $B5,ZV$IA
     000032  001A    0F80         T                  B         >+$B
     000033                            *                    CODE BREA<]
     000034  001B         0000    X                  DC        <ZVCONS
     000035  001C    001C                       $A     DC        <$
     000036                            *                     ]CB]
     000037  001D    0002                            DC        <D+2
     000038  001E                               $B     ORG       $
     000039  001E    E2C8 FFE2                       LLH       6,*D+1
     000040  0020    EF48 FFE0                       STR       6,*D+1
     000041                            *                 TARGET := 'BITS' @8,8@ TARGET;          (CHAR IN RT BYTE)
     000042                            *         'END';
     000043                            *         'END'
     000044                            *         'FINISH'
     000045  0022    83C8 FFDD                       JMP       *D
     000046          0024              L1            EQU       $
     000047  0024    83C0 0000    P                  JMP       ZK$EX
     000048                                          XLOC      ZK$EX
     000049                                          CTRL LINK ZK$EX
     000050                                          XLOC      ZVCONS
     000051                                          XLOC      ZV$TTY
     000052                                          CTRL LINK ZV$TT
     000053  0026    0004                            END       ZVKGC,ZVKGC
0000  ERR COUNT
L
```

```
E   ZVKJCA.LIST                          05/19/78  1422.8R W 05/19/78  1422.8       745380000
000001                                              TITLE     ZVKJCA,'REV 00'
000002          0025                                XDEF      JCA
000003          0020                                XDEF      ZVKJCA
000004          0008                        ZKCOM   COMM      8
000005  0000    0000                        D       RESV      7,0
000006  0007    0012                        T7      DC        18
000007  0008    5641 4C55 4520                      TEXT      'VALUE READ IS HEX$'
        000B    5245 4144 2049
                5320 4845 5824
000008  0011    000C                        T8      DC        12
000009  0012    2C20 5348 4F55                      TEXT      ', SHOULD BE$'
        0015    4C44 2042 4524
000010  0018    000D                        T9      DC        13
000011  0019    2C20 494E 2043                      TEXT      ', IN COLUMN $'
        001C    4F4C 554D 4E20
                2400
000012          0020                        ZVKJCA  EQU       $
000013  0020    8F00 0001       K                   SAVE      <ZKCOM+1,Z'0011'
        0022    0011
000014  0023    0F81 0059                           B         L1
000015                                      *       'PROCEDURE'COMPARE ASCII('VALUE''INTEGER'TEMP BUFFER,TEMP CHAR,COLUMN);
000016                                      *       'BEGIN'
000017  0025    DFC0 FFDA                   JCA     STB       5,D
000018  0027    8F87                                RSTR      $B7,Z'0700'
        0028    0700
000019  0029    8F40 FFD7                           SAVE      D+1,Z'0700'
        002B    0700
000020                                              XLOC      ZV$ER
000021  002C    DFC0 FFD7                           STB       $B5,D+4
000022                                      *           'IF' ZVDTTY 'NE' 0
000023                                      *               'THEN'
000024                                      *               'BEGIN'
000025  002E    E840 0000       P                   LDR       6,ZV$TTY
000026  0030    6938                                BEZ       6,>L3
000027                                      *                   'IF'
000028                                      *                       GET BREAK 'EQ' TRUE
000029  0031    D3C0 0000       P                   LNJ       5,VKGB
000030                                      *                   'THEN'
000031                                      *                       'GO TO' ASK NEXT;
000032  0033    6D01                                CMV       6,1
000033  0034    0901 0000       P                   BE        ASKNEX
000034                                      *                   'IF'TEMP BUFFER 'NE' TEMP CHAR
000035                                      *                   'THEN'
000036                                      *                   'BEGIN'
000037  0036    E840 FFCA                           LDR       6,D+1
000038  0038    E940 FFC9                           CMR       6,D+2
000039  003A    0901 0040                           BE        L6
000040                                      *                       PUT NEW TEXT ("VALUE READ IS HEX$");
000041  003C    FBC0 FFC8                           LAB       7,D+5
000042  003E    9BC0 FFC8                           LAB       1,T7
000043  0040    9F87                                STB       1,$B7
000044  0041    D3C0 0000       P                   LNJ       5,VKPNT
000045  0043    FBC0 FFC1                           LAB       7,D+5
000046  0045    E840 FFBB                           LDR       6,D+1
000047  0047    EF07                                STR       6,$B7
000048  0048    D3C0 0000       P                   LNJ       5,VKPH
000049                                      *                         PUT HEX       (TEMP BUFFER);
000050                                      *                         PUT TEXT      (", SHOULD BE$");
000051  004A    FBC0 FFBA                           LAB       7,D+5
000052  004C    9BC0 FFC4                           LAB       1,T8
000053  004E    9F87                                STB       1,$B7
000054  004F    D3C0 0000       P                   LNJ       5,VKPT
000055  0051    FBC0 FFB3                           LAB       7,D+5
000056  0053    E840 FFAE                           LDR       6,D+2
000057  0055    EF07                                STR       6,$B7
000058  0056    D3C0 0000       P                   LNJ       5,VKPH
000059                                      *                       PUT HEX       (TEMP CHAR);
000060  0058    FBC0 FFAC                           LAB       7,D+5
000061  005A    9BC0 FFBD                           LAB       1,T9
000062  005C    9F87                                STB       1,$B7
000063  005D    D3C0 0000       P                   LNJ       5,VKPT
000064                                      *                       PUT TEXT      (", IN COLUMN $");
000065  005F    FBC0 FFA5                           LAB       7,D+5
000066  0061    E840 FFA1                           LDR       6,D+3
000067  0063    EF07                                STR       6,$B7
000068  0064    D3C0 0000       P                   LNJ       5,JPN
000069                                      *                       PUT NUMBER    (COLUMN);
000070                                      *                       'END';        (OF IF)
000071                                      *               'END'
000072                                      *               'ELSE'
000073                                      *               'BEGIN'
000074  0066    0F81 0014                           B         L6
000075          0068                        L3      EQU       $
000076                                      *                   LABEL 2 := 'LITERAL' (CA);
000077  0068    E870 4341                           LDR       6,=17217
000078  006A    EF40 0000       P                   STR       6,LABEL2
000079                                      *                 'CODE''BEGIN']
000080                                      *                       LDR $R6,]TEMP CHAR
000081  006C    E840 FF95                           LDR       $R6,D+2
000082  006E    C840 FF92                           LDR       $R4,D+1
000083  0070    D840 FF92                           LDR       $R5,D+3
000084  0072    DCC0 FF91                           LDB       $B5,D+4
000085  0074    DFC0 0000       T                   STB       $B5,+$A
000086  0076    D380 0000       X                   LNJ       $B5,,<ZV$ER
000087  0078    0F80            T                   B         >+$B
000088  0079    0079                        $A      DC        <$
000089  007A            0000    X                   DC        <LABEL1
000090  007B                                $B      ORG       $
000091                                      *                 LOAD R4 R5 AND REPORT ERROR (TEMP BUFFER, COLUMN);
000092                                      *               'END';
000093                                      *       'END';
000094                                      *       'END';      (OF SEGMENT)
000095                                      *       'FINISH'
000096  007B    83C8 FF84                   L6      JMP       *D
000097          007D                        L1      EQU       $
000098  007D    83C0 0000       P                   JMP       ZK$EX
000099                                              XLOC      ZK$EX
000100                                              CTRL LINK ZK$EX
000101                                              XLOC      VKPT
000102                                              XLOC      VKPNT
000103                                              XLOC      VKPH
000104                                              XLOC      VKGB
```

```
     000105                                          XLOC    LABEL1
     000106                                          XLOC    LABEL2
     000107                                          XLOC    ZV$TTY
     000108                                          CTRL LINK ZV$TT
     000109                                          XLOC    JPN
     000110                                          XLOC    ASKNEX
     000111  007F  0020                              END     ZVKJCA,ZVKJCA
0000  ERR COUNT
@L
```

```
E   ZVKJCC.LIST                          05/19/78  1422.8R W 05/19/78  1422.2      3091140000
000001                                                TITLE     ZVKJCC,'REV 00'
000002          002F                                  XDEF      JCC
000003          002A                                  XDEF      ZVKJCC
000004          0008                    ZKCOM         COMM      8
000005  0000    0000                    D             RESV      13,0
000006  000D    0007                    T10           DC        7
000007  000E    0D0A 4341 5244                        TEXT      Z'0D0A','CARD$'
        0011    2400
000008  0012    0008                    T17           DC        8
000009  0013    434F 4C55 4D4E                        TEXT      'COLUMN $'
        0016    2024
000010  0017    0003                    T21           DC        3
000011  0018    2C20 2400                             TEXT      ', $'
000012  001A    0005                    T29           DC        5
000013  001B    524F 5720 2400                        TEXT      'ROW $'
000014  001E    0002                    T35           DC        2
000015  001F    2C24                                  TEXT      ',$'
000016  0020    0008                    T39           DC        8
000017  0021    4452 4F50 5045                        TEXT      'DROPPED$'
        0024    4424
000018  0025    0007                    T42           DC        7
000019  0026    5049 434B 4544                        TEXT      'PICKED$'
        0029    2400
000020          002A                    ZVKJCC        EQU       $
000021  002A    8F00 0001       K                     SAVE      <ZKCOM+1,Z'0011'
        002C    0011
000022                                  *           'BEGIN'
000023  002D    0F81 019C                             B         L1
000024                                  *               'INTEGER' J,I;
000025                                  *               'INTEGER' MASK;
000026                                  *               'INTEGER' RANGE;
000027                                  *               'INTEGER' PRINT CARD;
000028                                  *               'INTEGER' PRINT COLUMN;
000029                                  *               'INTEGER' PRINT ROW;
000030  002F    0FC0 FFD0               JCC           STB       5,D
000031                                  *           PRINT CARD := -1;
000032  0031    6CFF                                  LDV       6,-1
000033  0032    EF40 FFD2                             STR       6,D+5
000034                                  *           PRINT ROW :=10;
000035  0034    4C0A                                  LDV       4,10
000036  0035    CF40 FFD1                             STR       4,D+7
000037                                  *           I:=0;
000038  0037    8740 FFCA                             CL        D+2
000039  0039    FBC0 FFCE                             LAB       7,D+8
000040  003B    9BC0 FFC8                             LAB       1,D+4
000041  003D    9F87                                  STB       1,$B7
000042                                  *           INOUT (RANGE,INPUT RANGE @0@);
000043  003E    0F00 0009       X       N1            NOP       <IOWORD+9
000044  0040    D840 FFFE                             LDR       5,N1+1
000045  0042    DF40 FFC6                             STR       5,D+9
000046  0044    D3C0 0000       P                     LNJ       5,JIO
000047  0046    FBC0 FFC1                             LAB       7,D+8
000048  0048    9BC0 0000       P                     LAB       1,SETTOH
000049  004A    9F87                                  STB       1,$B7
000050                                  *           INOUT (SET TO HOLLERITH, OUTPUT TASK @0@);
000051  004B    0F00 0005       X       N2            NOP       <IOWORD+5
000052  004D    E840 FFFE                             LDR       6,N2+1
000053  004F    EF40 FFB9                             STR       6,D+9
000054  0051    D3C0 0000       P                     LNJ       5,JIO
000055  0053    D3C0 0000       P                     LNJ       5,VKGB
000056                                  *           'IF' GET BREAK 'EQ' TRUE
000057                                  *           'THEN'
000058                                  *               'GO TO' ASK NEXT;
000059  0055    6D01                                  CMV       6,1
000060  0056    0901 0000       P                     BE        ASKNEX
000061          0058                    L5            EQU       $
000062                                  *           'FOR' J :=0 'STEP' 1 'UNTIL' 6
000063  0058    8740 FFA8                             CL        D+1
000064                                  *           'DO'
000065                                  *               'BEGIN'
000066  005A    E840 FFA6               L6            LDR       6,D+1
000067  005C    6EFA                                  ADV       6,-6
000068  005D    6A01 0163                             BGZ       6,L7
000069                                  *           PRINT COLUMN := J*16+1;
000070  005F    E840 FFA1                             LDR       6,D+1
000071  0061    6004                                  SOL       6,4
000072  0062    6E01                                  ADV       6,1
000073  0063    EF40 FFA2                             STR       6,D+6
000074                                  *                       'IF' CHECKSUM @J@ 'NE' @LOWADDRESS   +J@
000075                                  *                       'THEN'
000076  0065    9840 0000       P                     LDR       1,LOWADD
000077  0067    9A40 FF99                             ADD       1,D+1
000078                                  *                       'BEGIN'
000079  0069    9F40 FF9E                             STR       1,D+8
000080  006B    C848 FF9C                             LDR       4,*D+8
000081  006D    A840 FF93                             LDR       2,D+1
000082  006F    C920 0000       X                     CMR       4,<CHECKS.$R2
000083  0071    0901 014B                             BE        L8
000084                                  *                               'IF' PRINT CARD 'EQ' -1
000085                                  *                               'THEN'
000086                                  *                               'BEGIN'
000087  0073    C840 FF91                             LDR       4,D+5
000088  0075    4DFF                                  CMV       4,-1
000089  0076    09B0                                  BNE       >L9
000090                                  *                                       'INTEGER' EJECT;
000091                                  *                                       EJECT := 'HEX' (4600);
000092  0077    D870 4600                             LDR       5,=17920
000093  0079    DF40 FF8E                             STR       5,D+8
000094  007B    FBC0 FF8D                             LAB       7,D+9
000095  007D    9BC0 FF8A                             LAB       1,D+8
000096  007F    9F87                                  STB       1,$B7
000097                                  *                                       INOUT (EJECT,OUTPUT TASK @0@);
000098  0080    0F00 0005       X       N3            NOP       <IOWORD+5
000099  0082    E840 FFFE                             LDR       7,N3+1
000100  0084    FF40 FF85                             STR       7,D+10
000101  0086    D3C0 0000       P                     LNJ       5,JIO
000102                                  *                                       PUT NEW TEXT ("!C L!CARD$");
000103  0088    FBC0 FF80                             LAB       7,D+9
000104  008A    9BC0 FF82                             LAB       1,T10
000105  008C    9F87                                  STB       1,$B7
000106  008D    D3C0 0000       P                     LNJ       5,VKPNT
000107                                  *                                       PRINT CARD := CARDS READ -1;
```

```
000108  008F  E840 0000      P             LDR       6,CARDSR
000109  0091  6EFF                        ADV       6,-1
000110  0092  EF40 FF72                    STR       6,D+5
000111  0094  FBC0 FF74                    LAB       7,D+9
000112  0096  EF07                        STR       6,$B7
000113  0097  D3C0 0000      P             LNJ       5,VKPD
000114                            *                                 PUT DECIMAL  (PRINT CARD);
000115                            *                                 'IF' CARDS READ 'GT' 4
000116                            *                                 'THEN'
000117                            *                                 'BEGIN'
000118  0099  E840 0000      P             LDR       6,CARDSR
000119  009B  6D04                        CMV       6,4
000120  009C  0A8A                        BALE      >L9
000121                            *                       SUM := SUM+TIME1-100;(SUM OF ERROR SPEEDS)
000122  009D  C840 0000      P             LDR       4,SUM
000123  009F  CA40 0000      P             ADD       4,TIME1
000124  00A1  4L9C                        ADV       4,-100
000125  00A2  CF40 0000      P             STR       4,SUM
000126                            *                        TOTAL := TOTAL + 1; (TOTAL ERRORS)
000127  00A4  8AC0 0000      P             INC       TOTAL
000128                            *                                 'END';
000129                            *                       'END';
000130                            **;
000131                            **                    COMPARE COLUMN CHECKSUMS;
000132                            **;
000133        00A6                L9        EQU       $
000134                            *                         MASK := 'HEX' (8000);
000135                            *           CHECK NEXT COLUMN;
000136  00A6  E870 8000                      LDR       6,=-32768
000137  00A8  EF40 FF5A                      STR       6,D+3
000138        00AA                L12       EQU       $
000139                            *                       'IF' CHECKSUM @J@ 'MASK' MASK 'NE' @LOWADDRESS  +J@
000140  00AA  E840 FF58                      LDR       6,D+3
000141  00AC  9840 FF54                      LDR       1,D+1
000142  00AE  E510 0000      X               AND       6,<CHECKS,$R1
000143  00B0  9A40 0000      P               ADD       1,LOWADD
000144  00B2  9F40 FF56                      STR       1,D+9
000145  00B4  C848 FF54                      LDR       4,*D+9
000146  00B6  C540 FF4C                      AND       4,D+3
000147  00B8  E954                          CMR       6,=$R4
000148  00B9  0901 0045                      BE        L13
000149                            *                       'MASK' MASK 'AND' J 'LI' 5
000150                            *                       'THEN'
000151                            *                       'BEGIN'
000152  00BB  E840 FF45                      LDR       6,D+1
000153  00BD  6D05                          CMV       6,5
000154  00BE  0881 0040                      BAGE      L13
000155                            *                                 LABEL 2 := 'LITERAL' (CC);
000156  00C0  C870 4343                      LDR       4,=17219
000157  00C2  CF40 0000      P               STR       4,LABEL2
000158                            *                                 'IF' ZVDTTY 'EQ' 0
000159                            *                                 'THEN'
000160                            *                                  'BEGIN'
000161  00C4  D840 0000      P               LDR       5,ZV$TTY
000162  00C6  5981 0007                      BNEZ      5,L14
000163  00C8  D840 FF3C                      LDR       $R5,D+5
000164  00CA  C840 FF3B                      LDR       $R4,D+6
000165  00CC  D3C0 0000      P               LNJ       5,JRE
000166                            *                            REPORT ERROR;
000167                            *                       'END';
000168                            *                       'IF' I  'EQ' 0
000169        00CE                L14       EQU       $
000170                            *                       'THEN'
000171  00CE  E840 FF33                      LDR       6,D+2
000172  00D0  6988                          BNEZ      6,>L16
000173        00D1                L15       EQU       $
000174                            *           PUT COLUMN;                 PUT NEW TEXT ("COLUMN $");
000175  00D1  FBC0 FF36                      LAB       7,D+8
000176  00D3  9BC0 FF3E                      LAB       1,T17
000177  00D5  9F87                          STB       1,$B7
000178  00D6  D3C0 0000      P               LNJ       5,VKPNT
000179                            *                       'IF' I 'LT' 16
000180        00D8                L16       EQU       $
000181                            *                       'THEN'
000182                            *                       'BEGIN'
000183  00D8  E840 FF29                      LDR       6,D+2
000184  00DA  6D10                          CMV       6,16
000185  00DB  08A1                          BAGE      >L18
000186                            *                        'INTEGER' CORRECT COLUMN;
000187                            *                        'IF' PUESDO CARD STATUS 'EQ' 'HEX' (1800)
000188                            *                        'THEN'
000189  00DC  C840 0000      P               LDR       4,PUESDO
000190  00DE  C970 1800                      CMR       4,=6144
000191  00E0  0987                          BNE       >L19
000192                            *                                  CORRECT COLUMN := PRINT COLUMN * 2
000193                            *                        'ELSE'
000194  00E1  D840 FF24                      LDR       5,D+6
000195  00E3  5001                          SOL       5,1
000196  00E4  DF40 FF23                      STR       5,D+8
000197  00E6  0F85                          B         >L20
000198  00E7  E840 FF1E      L19             LDR       6,D+6
000199  00E9  EF40 FF1E                      STR       6,D+8
000200                            *                             CORRECT COLUMN := PRINT COLUMN;
000201        00EB                L20       EQU       $
000202                            *                        PUT NUMBER (CORRECT COLUMN);
000203  00EB  FBC0 FF1D                      LAB       7,D+9
000204  00ED  E840 FF1A                      LDR       6,D+8
000205  00EF  EF07                          STR       6,$B7
000206  00F0  D3C0 0000      P               LNJ       5,JPN
000207                            *                       PUT TEXT  (", $");
000208  00F2  FBC0 FF16                      LAB       7,D+9
000209  00F4  9BC0 FF22                      LAB       1,T21
000210  00F6  9F87                          STB       1,$B7
000211  00F7  D3C0 0000      P               LNJ       5,VKPT
000212                            *                        I := I +1;
000213                            *                       'END'
000214  00F9  8AC0 FF08                      INC       D+2
000215                            *                       'ELSE'
000216                            *                       'BEGIN'
000217  00FB  0F84                          B         >L13
000218        00FC                L18       EQU       $
000219                            *                        I :=0;
000220  00FC  8740 FF05                      CL        D+2
```

```
000221                                  *                                       'GO TO' PUT COLUMN;
000222  00FE  0FD3                             B           >L15
000223                                  *                                   'END';
000224                                  *                               'END';
000225        00FF                      L13       EQU       $
000226                                  *                               MASK := 'BITS' @15,1@ MASK;
000227  00FF  E840 FF03                         LDR       6,D+3
000228  0101  6041                             SOR       6,1
000229  0102  EF40 FF00                        STR       6,D+3
000230                                  *                               PRINT COLUMN := PRINT COLUMN + 1;
000231  0104  8AC0 FF01                        INC       D+6
000232                                  *                               'IF' MASK 'NE' 0
000233                                  *                               'THEN'
000234                                  *                                   'GO TO' CHECK NEXT COLUMN;
000235  0106  6981 FFA3                        BNEZ      6,L12
000236                                  *                               'IF' J 'LT' 5
000237                                  *                               'THEN'
000238                                  *                                   'GO TO' END OF LOOP;
000239  0108  C840 FEF8                         LDR       4,D+1
000240  010A  4D05                             CMV       4,5
000241  010B  0801 00B1                        BAL       L8
000242                                  *                               'IF' J 'GT' 5
000243                                  *                               'THEN'
000244                                  *                                   'GO TO' SKIP ROW;
000245  010D  4D05                             CMV       4,5
000246  010E  0A01 0057                        BAG       L27
000247                                  *                               'IF' CHECKSUM @5@ 'NE' @LOWADDRESS+5@ 'AND' J 'EQ' 5
000248  0110  D800 0005       X                LDR       5,<CHECKS+5
000249  0112  1C05                             LDV       1,5
000250  0113  D918 0000       X                CMR       5,*<LOWADD.$R1
000251  0115  090A                             BE        >L28
000252                                  *                               'THEN'
000253  0116  C951                             CMR       4,=$R1
000254  0117  0988                             BNE       >L28
000255                                  *                                   PUT NEW TEXT ("ROW $");
000256  0118  FBC0 FEEF                        LAB       7,D+8
000257  011A  9BC0 FEFF                        LAB       1,T29
000258  011C  9F87                             STB       1,$B7
000259  011D  D3C0 0000       P                LNJ       5,VKPNT
000260        011F                      L28       EQU       $
000261                                  *                               MASK := 'HEX' (800);
000262                                  *       CHECK NEXT ROW:
000263  011F  E870 0800                         LDR       6,=2048
000264  0121  EF40 FEE1                        STR       6,D+3
000265        0123                      L30       EQU       $
000266                                  *                               PRINT ROW := PRINT ROW -1;
000267  0123  88C0 FEE3                        DEC       D+7
000268                                  *                               'IF' PRINT ROW 'EQ' -1
000269                                  *                           'THEN'
000270  0125  E840 FEE1                         LDR       6,D+7
000271  0127  6DFF                             CMV       6,-1
000272  0128  0984                             BNE       >L31
000273                                  *                                   PRINT ROW := 11;
000274  0129  4C0B                             LDV       4,11
000275  012A  CF40 FEDC                        STR       4,D+7
000276        012C                      L31       EQU       $
000277                                  *                               'IF' PRINT ROW 'EQ' 10
000278                                  *                               'THEN'
000279  012C  E840 FEDA                         LDR       6,D+7
000280  012E  6D0A                             CMV       6,10
000281  012F  0984                             BNE       >L32
000282                                  *                                   PRINT ROW := 12;
000283  0130  4C0C                             LDV       4,12
000284  0131  CF40 FED5                        STR       4,D+7
000285        0133                      L32       EQU       $
000286                                  *                               'IF' CHECKSUM@5@ 'MASK' MASK 'NE' @LOWADDRESS+5@
000287  0133  E840 FECF                         LDR       6,D+3
000288  0135  E500 0005       X                AND       6,<CHECKS+5
000289  0137  C840 FECB                         LDR       4,D+3
000290  0139  1C05                             LDV       1,5
000291  013A  C518 0000       X                AND       4,*<LOWADD.$R1
000292  013C  E954                             CMR       6,=$R4
000293  013D  0901 0021                        BE        L33
000294                                  *                               'MASK' MASK 'AND' J 'EQ' 5
000295                                  *                               'THEN'
000296                                  *                               'BEGIN'
000297  013F  9940 FEC1                         CMR       1,D+1
000298  0141  0981 001D                        BNE       L33
000299                                  *                                   LABEL 2 := 'LITERAL' (CR);
000300  0143  E870 4352                         LDR       6,=17234
000301  0145  EF40 0000       P                STR       6,LABEL2
000302                                  *                                   'IF' ZVDITY 'EQ' 0
000303                                  *                                   'THEN'
000304                                  *                                   'BEGIN'
000305  0147  C840 0000       P                LDR       4,ZV$TTY
000306  0149  4981 0007                        BNEZ      4,L34
000307  014B  D840 FEB9                        LDR       $R5,D+5
000308  014D  C840 FEB9                        LDR       $R4,D+7
000309                                  *                                       LOAD R4 R5 (PRINT ROW, PRINT CARD);
000310  014F  D3C0 0000       P                LNJ       5,JRE
000311                                  *                                       REPORT ERROR;
000312                                  *                                   'END';
000313  0151  FBC0 FEB6                 L34     LAB       7,D+8
000314  0153  E840 FEB3                         LDR       6,D+7
000315  0155  EF07                             STR       6,$B7
000316  0156  D3C0 0000       P                LNJ       5,JPN
000317                                  *                                   PUT NUMBER (PRINT ROW);
000318                                  *                                   PUT TEXT (",$");
000319  0158  FBC0 FEAF                         LAB       7,D+8
000320  015A  9BC0 FEC3                        LAB       1,T35
000321  015C  9F87                             STB       1,$B7
000322  015D  D3C0 0000       P                LNJ       5,VKPT
000323                                  *                               'END';
000324        015F                      L33       EQU       $
000325                                  *                               MASK := MASK/2;
000326  015F  E840 FEA3                         LDR       6,D+3
000327  0161  6061                             SAR       6,1
000328  0162  EF40 FEA0                        STR       6,D+3
000329                                  *                               'IF' MASK 'NE' 0
000330                                  *                               'THEN'
000331                                  *                                   'GO TO' CHECK NEXT ROW;
000332                                  *       SKIP ROW:
000333  0164  6981 FFBE                         BNEZ      6,L30
```

```
000334          0166                    L27     EQU     $
000335                                  *               'IF' CHECKSUM @6@ 'LT' @LOW ADDRESS+6@ 'AND' J 'EQ' 6
000336  0166    E800 0006       X               LDR     6,<CHECKS+6
000337  0168    1C06                            LDV     1,6
000338  0169    E918 0000       X               CMR     6,*<LOWADD.$R1
000339  016B    0881 0026                       BAGE    L37
000340                                  *                       'THEN'
000341                                  *                       'BEGIN'
000342  016D    9940 FE93                       CMR     1,D+1
000343  016F    0981 0022                       BNE     L37
000344                                  *                               BIT COUNT:= @LOWADDRESS+6@ - CHECKSUM @6@;
000345  0171    E218 0000       X               SUB     6,*<LOWADD.$R1
000346  0173    8256                            NEG     =$R6
000347  0174    EF40 0000       P               STR     6,BITCOU
000348                                  *                               LABEL 2 := 'LITERAL' (CD);
000349  0176    C870 4344                       LDR     4,=17220
000350  0178    CF40 0000       P               STR     4,LABEL2
000351                                  *                               'IF' ZVDITY 'EQ' 0
000352                                  *                               'THEN'
000353                                  *                               'BEGIN'
000354  017A    D840 0000       P               LDR     5,ZV$TTY
000355  017C    5981 0007                       BNEZ    5,L38
000356  017E    D840 FE86                       LDR     $R5,D+5
000357  0180    C840 0000       P               LDR     $R4,BITCOU
000358                                  *                                       LOAD R4 R5 (BIT COUNT, PRINT CARD);
000359  0182    D3C0 0000       P               LNJ     5,JRE
000360                                  *                                       REPORT ERROR;
000361                                  *                               'END';
000362          0184                    L38     EQU     $
000363                                  *                               PUT NEW TEXT ("DROPPED$");
000364  0184    FBC0 FE83                       LAB     7,D+8
000365  0186    9BC0 FE99                       LAB     1,T39
000366  0188    9F87                            STB     1,$B7
000367  0189    D3C0 0000       P               LNJ     5,VKPNT
000368  018B    FBC0 FE7C                       LAB     7,D+8
000369  018D    E840 0000       P               LDR     6,BITCOU
000370  018F    EF07                            STR     6,$B7
000371  0190    D3C0 0000       P               LNJ     5,VKPD
000372                                  *                               PUT DECIMAL  (BIT COUNT );
000373                                  *                       'END';
000374          0192                    L37     EQU     $
000375                                  *               'IF' CHECKSUM @6@ 'GT' @LOW ADDRESS+6@ 'AND' J 'EQ' 6
000376  0192    E800 0006       X               LDR     6,<CHECKS+6
000377  0194    1C06                            LDV     1,6
000378  0195    E918 0000       X               CMR     6,*<LOWADD.$R1
000379  0197    0A81 0025                       BALE    L8
000380                                  *                       'THEN'
000381                                  *                       'BEGIN'
000382  0199    9940 FE67                       CMR     1,D+1
000383  019B    0981 0021                       BNE     L8
000384                                  *                               BIT COUNT := CHECKSUM @6@ - @LOWADDRESS+6@;
000385  019D    E218 0000       X               SUB     6,*<LOWADD.$R1
000386  019F    EF40 0000       P               STR     6,BITCOU
000387                                  *                               LABEL 2 := 'LITERAL' (CG);
000388  01A1    C870 4347                       LDR     4,=17223
000389  01A3    CF40 0000       P               STR     4,LABEL2
000390                                  *                               'IF' ZVDITY 'EQ' 0
000391                                  *                               'THEN'
000392                                  *                               'BEGIN'
000393  01A5    D840 0000       P               LDR     5,ZV$TTY
000394  01A7    5981 0007                       BNEZ    5,L41
000395  01A9    D840 FE5B                       LDR     $R5,D+5
000396  01AB    C840 0000       P               LDR     $R4,BITCOU
000397                                  *                                       LOAD R4 R5 (BIT COUNT, PRINT CARD);
000398  01AD    D3C0 0000       P               LNJ     5,JRE
000399                                  *                                       REPORT ERROR;
000400                                  *                               'END';
000401          01AF                    L41     EQU     $
000402                                  *                               PUT NEW TEXT ("PICKED$");
000403  01AF    FBC0 FE58                       LAB     7,D+8
000404  01B1    9BC0 FE73                       LAB     1,T42
000405  01B3    9F87                            STB     1,$B7
000406  01B4    D3C0 0000       P               LNJ     5,VKPNT
000407  01B6    FBC0 FE51                       LAB     7,D+8
000408  01B8    E840 0000       P               LDR     6,BITCOU
000409  01BA    EF07                            STR     6,$B7
000410  01BB    D3C0 0000       P               LNJ     5,VKPD
000411                                  *                               PUT DECIMAL  (BIT COUNT);
000412                                  *                       'END';
000413                                  *               'END';                  (OF IF CHECKSUM @J@)
000414                                  *       END OF LOOP;
000415                                  *           'END';                      (OF FOR LOOP )
000416                                  *       END OF PROCEDURE;
000417  01BD    8AC0 FE43               L8        INC     D+1
000418  01BF    0F81 FE9A                         B       L6
000419          01C1                    L7        EQU     $
000420                                  *           LOW ADDRESS := LOW ADDRESS + 7;
000421  01C1    E840 0000       P                 LDR     6,LOWADD
000422  01C3    6E07                              ADV     6,7
000423  01C4    EF40 0000       P                 STR     6,LOWADD
000424                                  *             TIME DELAY;
000425  01C6    D3C0 0000       P                 LNJ     5,JTD
000426                                  *       'END';                          (OF PROCEDURE)
000427                                  *   'END';      (OF SEGMENT)
000428                                  *   'FINISH'
000429  01C8    83C8 FE37                         JMP     *D
000430          01CA                    L1        EQU     $
000431  01CA    83C0 0000       P                 JMP     ZK$EX
000432                                            XLOC    ZK$EX
000433                                            CTRL LINK ZK$EX
000434                                            XLOC    VKPT
000435                                            XLOC    VKPNT
000436                                            XLOC    VKPD
000437                                            XLOC    VKGB
000438                                            XLOC    PUESDO
000439                                            XLOC    SETTOH
000440                                            XLOC    LABEL2
000441                                            XLOC    CARDSR
000442                                            XLOC    BITCOU
000443                                            XLOC    LOWADD
000444                                            XLOC    TIME1
000445                                            XLOC    SUM
000446                                            XLOC    TOTAL
```

```
000447                                  XLOC    ZV$TTY
000448                                  CTRL_LINK ZV$TT
000449                                  XLOC    CHECKS
000450                                  XLOC    IOWORD
000451                                  XLOC    JRE
000452                                  XLOC    JIO
000453                                  XLOC    JPN
000454                                  XLOC    JTD
000455                                  XLOC    ASKNEX
000456  01CC  002A                      END     ZVKJCC,ZVKJCC
0000  ERR COUNT
@L
```

```
E   ZVKJCS.LIST                          05/19/78  1422.9R W 05/19/78  1422.9      1166670000
000001                                              TITLE     ZVKJCS,'REV 00'
000002          002F                                XDEF      JCS
000003          002A                                XDEF      ZVKJCS
000004          0008                    ZKCOM       COMM      8
000005  0000    0000                    D           RESV      11,0
000006  000B    0014                    T6          DC        20
000007  000C    5455 524E 2044                      TEXT      'TURN DEVICE ON-LIN'
        000F    4556 4943 4520
                4F4E 2D4C 494E
000008  0015    4524                                TEXT      'E$'
000009  0016    0015                    T8          DC        21
000010  0017    5455 524E 2044                      TEXT      'TURN DEVICE OFF-LI'
        001A    4556 4943 4520
                4F46 462D 4C49
000011  0020    4E45 2400                           TEXT      'NE$'
000012  0022    000E                    T20         DC        14
000013  0023    2C20 5752 4F4E                      TEXT      ', WRONG STATE$'
        0026    4720 5354 4154
                4524
000014          002A                    ZVKJCS      EQU       $
000015  002A    8F00 0001       K                   SAVE      <ZKCOM+1,Z'0011'
        002C    0011
000016  002D    0F81 0098                           B         L1
000017                                  *       'PROCEDURE' CHECK STATE ('VALUE' 'INTEGER' STATE SHOULD BE);
000018                                  *       'BEGIN'
000019                                  *           'INTEGER' K,I,J;
000020  002F    DFC0 FFD0               JCS         STB       5,D
000021  0031    E807                                LDR       6,$B7
000022  0032    EF40 FFCE                           STR       6,D+1
000023                                              XLOC      ZV$ER
000024  0034    DFC0 FFD0                           STB       $B5,D+5
000025                                  *           K := 0;
000026                                  *       READ STATUS AGAIN:
000027  0036    8740 FFCB                           CL        D+2
000028  0038    FBC0 FFCD               L3          LAB       7,D+6
000029  003A    9BC0 0000       P                   LAB       1,STATWA
000030  003C    9F87                                STB       1,$B7
000031                                  *               INOUT (STATUS WAS,INPUT STATUS WORD @0@);
000032  003D    0F00 000C       X       N1          NOP       <IOWORD+12
000033  003F    E840 FFFE                           LDR       6,N1+1
000034  0041    EF40 FFC5                           STR       6,D+7
000035  0043    D3C0 0000       P                   LNJ       5,JIO
000036                                  *           'IF' 'BITS' @1,15@ STATUS WAS 'NE' STATE SHOULD BE
000037                                  *           'THEN'
000038                                  *           'BEGIN'
000039  0045    E2C0 0000       P                   LLH       6,STATWA
000040  0047    6047                                SOR       6,7
000041  0048    E940 FFB8                           CMR       6,D+1
000042  004A    0901 006C                           BE        L4
000043                                  *                   ERROR FLAG := ERROR FLAG + 1;
000044  004C    8AC0 0000       P                   INC       ERRORF
000045                                  *                   'IF' STATE SHOULD BE 'EQ' 1
000046                                  *                   'THEN'
000047  004E    C840 FFB2                           LDR       4,D+1
000048  0050    4D01                                CMV       4,1
000049  0051    0989                                BNE       >L5
000050                                  *                           PUT NEW TEXT ("TURN DEVICE ON-LINE$")
000051                                  *                   'ELSE'
000052  0052    FBC0 FFB3                           LAB       7,D+6
000053  0054    9BC0 FFB6                           LAB       1,T6
000054  0056    9F87                                STB       1,$B7
000055  0057    D3C0 0000       P                   LNJ       5,VKPNT
000056  0059    0F88                                B         >L7
000057          005A                    L5          EQU       $
000058                                  *                   PUT NEW TEXT ("TURN DEVICE OFF-LINE$");
000059  005A    FBC0 FFAB                           LAB       7,D+6
000060  005C    9BC0 FFB9                           LAB       1,T8
000061  005E    9F87                                STB       1,$B7
000062  005F    D3C0 0000       P                   LNJ       5,VKPNT
000063          0061                    L7          EQU       $
000064  0061    8740 FFA2               L9          CL        D+4
000065                                  *                   'FOR' J:=0 'STEP' 1 'UNTIL' SPEED/4      'DO'
000066  0063    E840 0000       P                   LDR       6,SPEED
000067  0065    6062                                SAR       6,2
000068  0066    EF40 FF9F                           STR       6,D+6
000069  0068    E840 FF9B               L10         LDR       6,D+4
000070  006A    E240 FF9B                           SUB       6,D+6
000071  006C    6A27                                BGZ       6,>L11
000072  006D    8740 FF95               L12         CL        D+3
000073                                  *                   'FOR' I:=0 'STEP' 1 'UNTIL' 'HEX' (7FFF) 'DO'
000074                                  *                   'BEGIN'
000075  006F    E840 FF93               L13         LDR       6,D+3
000076  0071    E270 7FFF                           SUB       6,=32767
000077  0073    6A1D                                BGZ       6,>L14
000078                                  *                       'IF'
000079                                  *                               GET BREAK 'EQ' TRUE
000080  0074    D3C0 0000       P                   LNJ       5,VKGB
000081                                  *                       'THEN'
000082                                  *                           'GO TO' ASK NEXT;
000083  0076    6D01                                CMV       6,1
000084  0077    0901 0000       P                   BE        ASKNEX
000085  0079    FBC0 FF8D                           LAB       7,D+7
000086  007B    9BC0 0000       P                   LAB       1,STATWA
000087  007D    9F87                                STB       1,$B7
000088                                  *                       INOUT (STATUS WAS, INPUT STATUS WORD @0@);
000089  007E    0F00 000C       X       N2          NOP       <IOWORD+12
000090  0080    E840 FFFE                           LDR       6,N2+1
000091  0082    EF40 FF85                           STR       6,D+8
000092  0084    D3C0 0000       P                   LNJ       5,JIO
000093                                  *                           'IF' 'BITS' @1,15@ STATUS WAS 'EQ' STATE SHOULD BE
000094                                  *                           'THEN'
000095                                  *                               'GO TO' READ STATUS AGAIN;
000096  0086    E2C0 0000       P                   LLH       6,STATWA
000097  0088    6047                                SOR       6,7
000098  0089    E940 FF77                           CMR       6,D+1
000099  008B    0901 FFAC                           BE        L3
000100                                  *                   'END';
000101  008D    8AC0 FF75                           INC       D+3
000102  008F    0FE0                                B         >L13
000103  0090    8AC0 FF73               L14         INC       D+4
000104  0092    0FD6                                B         >L10
000105          0093                    L11         EQU       $
```

```
000106                                  *                   K;= K+1;
000107                                  **  REPORT ERROR ONLY IF WITHOUT CONSOLE - WRONG STATE;
000108                                  **  OR AFTER 5TH MESSAGE;
000109  0093  8AC0 FF6E                        INC     D+2
000110                                  *                   LABEL2 ;= 'LITERAL' (C5);
000111  0095  E870 4353                        LDR     6,=17235
000112  0097  EF40 0000       P                STR     6,LABEL2
000113                                  *                   'IF' K     'EQ' 5
000114  0099  C840 FF68                        LDR     4,D+2
000115  009B  4D05                            CMV     4,5
000116  009C  0905                            BE      >L18
000117                                  *                   'OR' ZVDTTY 'EQ' 0
000118                                  *                   'THEN'
000119                                  *                   'BEGIN'
000120                                  *                       LOAD R4 R5 AND REPORT ERROR (STATUS WAS,
000121  009D  D840 0000       P                LDR     5,ZV$TTY
000122  009F  5981 FF98                        BNEZ    5,L3
000123        00A1                      L18     EQU     $
000124  00A1  C840 0000       P                LDR     $R4,STATWA
000125  00A3  D840 0000       P                LDR     $R5,STATUS
000126  00A5  DCC0 FF5F                        LDB     $B5,D+5
000127  00A7  DFC0 0000       T                STB     $B5,+$A
000128  00A9  D380 0000       X                LNJ     $B5,<ZV$ER
000129  00AB  0F80            T                B       >+$B
000130  00AC  00AC                      $A      DC      <$
000131  00AD        0000      X                DC      <LABEL1
000132  00AE                      $B      ORG     $
000133                                  *                                                   STATUS SHOULD BE);
000134  00AE  FBC0 FF57                        LAB     7,D+6
000135  00B0  9BC0 FF71                        LAB     1,T20
000136  00B2  9F87                            STB     1,$B7
000137  00B3  D3C0 0000       P                LNJ     5,VKPT
000138                                  *                       PUT TEXT (", WRONG STATE»");
000139                                  *                       'GO TO' ASK NEXT
000140                                  *                   'END'
000141                                  *                   'ELSE'
000142  00B5  83C0 0000       P                JMP     ASKNEX
000143                                  *                       'GO TO' READ STATUS AGAIN;
000144                                  *               'END';
000145  00B7  FBC0 FF4E                L4      LAB     7,D+6
000146  00B9  9BC0 0000       P                LAB     1,STATWA
000147  00BB  9F87                            STB     1,$B7
000148                                  *           INOUT (STATUS WAS,INPUT STATUS @0@);    (CLEAR ATTENTION BIT)
000149                                  *       'END';
000150  00BC  0F00 000C       X        N3      NOP     <IOWORD+12
000151  00BE  E840 FFFE                        LDR     6,N3+1
000152  00C0  EF40 FF46                        STR     6,D+7
000153  00C2  D3C0 0000       P                LNJ     5,JIO
000154                                  *       'END';          (OF SEGMENT)
000155                                  *       'FINISH'
000156  00C4  83C8 FF3B                        JMP     *D
000157        00C6                      L1      EQU     $
000158  00C6  83C0 0000       P                JMP     ZK$EX
000159                                          XLOC    ZK$EX
000160                                          CTRL LINK ZK$EX
000161                                          XLOC    VKPT
000162                                          XLOC    VKPNT
000163                                          XLOC    VKGB
000164                                          XLOC    STATWA
000165                                          XLOC    STATUS
000166                                          XLOC    LABEL1
000167                                          XLOC    LABEL2
000168                                          XLOC    SPEED
000169                                          XLOC    ERRORF
000170                                          XLOC    ZV$TTY
000171                                          CTRL LINK ZV$TT
000172                                          XLOC    IOWORD
000173                                          XLOC    JIO
000174                                          XLOC    ASKNEX
000175  00C8  002A                            END     ZVKJCS,ZVKJCS
000  ERR COUNT
```

```
E    ZVKJET.LIST                                05/19/78  1422.9R W 05/19/78  1422.9        54738000
000001                                                      TITLE     ZVKJET,'REV 00'
000002          0018                                        XDEF      JET
000003          0013                                        XDEF      ZVKJET
000004          0008                                ZKCOM   COMM      8
000005  0000    0000                                D       RESV      5,0
000006  0005    0006                                T5      DC        6
000007  0006    2C20 4154 2024                              TEXT      ', AT $'
000008  0009    0012                                T6      DC        18
000009  000A    2043 4152 4453                              TEXT      ' CARDS PER MINUTE$'
        000D    2050 4552 204D
                494E 5554 4524
000010          0013                                ZVKJET  EQU       $
000011  0013    8F00 0001       K                           SAVE      <ZKCOM+1,Z'0011'
        0015    0011
000012  0016    0F81 003A                                   B         L1
000013                                              *       'PROCEDURE' CALCULATE ELAPSED TIME('VALUE''INTEGER' NUMBER OF CARDS);
000014                                              *       'BEGIN'
000015                                              *           'INTEGER' TIME;
000016  0018    DFC0 FFE7                           JET       STB       5,D
000017  001A    E807                                          LDR       6,$B7
000018  001B    EF40 FFE5                                     STR       6,D+1
000019                                              *           PUT NUMBER (NUMBER OF CARDS);
000020  001D    FBC0 FFE5                                     LAB       7,D+3
000021  001F    EF07                                          STR       6,$B7
000022  0020    D3C0 0000       P                             LNJ       5,JPN
000023                                              *             'IF' ERROR FLAG 'NE' 0
000024                                              *               'THEN'
000025                                              *                 'GO TO' END CPM;
000026  0022    E840 0000       P                             LDR       6,ERRORF
000027  0024    6981 002A                                     BNEZ      6,L4
000028                                              *           PUT TEXT (", AT $");
000029  0026    FBC0 FFDC                                     LAB       7,D+3
000030  0028    9BC0 FFDC                                     LAB       1,T5
000031  002A    9F87                                          STB       1,$B7
000032  002B    D3C0 0000       P                             LNJ       5,VKPT
000033                                              *           'CODE''BEGIN';
000034  002D    E840 0000       P                             LDR       $R6,CKSEM
000035  002F    F840 0000       P                             LDR       $R7,ZHRTCC
000036                                              *                    ]CB[
000037                                              *                    NEG =$R7                    CONVERT TO TICKS COUNTED THIS TIME
000038                                              *                    DIV $R7,=60                 DIVIDE BY SECONDS/MIN
000039  0031    8257                                          NEG       =$R7
000040  0032    F370 003C                                     DIV       $R7,=60
000041                                              *                    ]CB[
000042                                              *                    SUB $R6,=30                 CHECK REMAINDER
000043                                              *                    BLZ $R6,>+$A                GREATER THEN 1/2
000044  0034    E270 001E                                     SUB       $R6,=30
000045  0036    6800            T                             BLZ       $R6,>+$A
000046                                              *                    ]CB[
000047                                              *                    INC =$R7                    THEN INC QUOTIENT
000048                                              *           $A       CL =$R6                     UPPER WORD OF DOUBLE PRECISION =0
000049  0037    8AD7                                          INC       =$R7
000050  0038    8756                                $A        CL        =$R6
000051                                              *                    ]CB[
000052  0039    FE40 FFC7                                     SWR       $R7,D+1
000053  003B    FB40 0000       P                             MUL       $R7,HERTZ
000054                                              *                    ]CB[
000055  003D    F340 FFC3                                     DIV       $R7,D+1
000056  003F    FF40 FFC2                                     STR       $R7,D+2
000057  0041    FBC0 FFC1                                     LAB       7,D+3
000058  0043    E840 FFBE                                     LDR       6,D+2
000059  0045    EF07                                          STR       6,$B7
000060  0046    D3C0 0000       P                             LNJ       5,JPN
000061                                              *           PUT NUMBER (TIME);
000062  0048    FBC0 FFBA                                     LAB       7,D+3
000063  004A    9BC0 FFBE                                     LAB       1,T6
000064  004C    9F87                                          STB       1,$B7
000065  004D    D3C0 0000       P                             LNJ       5,VKPT
000066                                              *           PUT TEXT (" CARDS PER MINUTE$");
000067                                              *       END CPM;
000068                                              *       'END';
000069                                              *       'END';
000070                                              *       'FINISH'
000071  004F    83C8 FFB0                           L4        JMP       *D
000072          0051                                L1        EQU       $
000073  0051    83C0 0000       P                             JMP       ZK$EX
000074                                                        XLOC      ZK$EX
000075                                                        CTRL LINK ZK$EX
000076                                                        XLOC      ZHRTCC
000077                                                        XLOC      VKPT
000078                                                        XLOC      HERTZ
000079                                                        XLOC      ERRORF
000080                                                        XLOC      CKSEM
000081                                                        XLOC      JPN
000082  0053    0013                                        END       ZVKJET,ZVKJET
0000  ERR COUNT
@L
```

```
E   ZVKJFB.LIST                          05/19/78  1423.0R W 05/19/78  1423.0       27765000O
    000001                                                TITLE     ZVKJFB,'REV 00'
    000002          000D                                  XDEF      JFB
    000003          0009                                  XDEF      ZVKJFB
    000004          0008                        ZKCOM     COMM      8
    000005  0000    0000                        D         RESV      9,0
    000006          0009                        ZVKJFB    EQU       $
    000007  0009    8F00 0001       K                     SAVE      <ZKCOM+1,Z'0011'
            000B    0011
    000008  000C    0F9F                                  B         >L1
    000009                                      *       'PROCEDURE' FILL BUFFER('INTEGER' 'ARRAY' BUFFERNAME;
    000010                                      *                                 'VALUE' 'INTEGER' WORDS,VALUE);
    000011                                      *       'BEGIN'
    000012                                      *           'INTEGER' I;
    000013  000D    DFC0 FFF2                   JFB       STB       5,D
    000014  000F    8F87                                  RSTR      $B7,Z'0F00'
            0010    0F00
    000015  0011    8F40 FFEF                             SAVE      D+1,Z'0F00'
            0013    0F00
    000016  0014    8740 FFF0                   L3        CL        D+5
    000017                                      *             'FOR' I:=0 'STEP' 1 'UNTIL' WORDS-1
    000018                                      *             'DO'
    000019                                      *             'BEGIN'
    000020  0016    E840 FFEC                             LDR       6,D+3
    000021  0018    6EFF                                  ADV       6,-1
    000022  0019    EF40 FFEC                             STR       6,D+6
    000023  001B    E840 FFE9                   L4        LDR       6,D+5
    000024  001D    E240 FFE8                             SUB       6,D+6
    000025  001F    6A0A                                  BGZ       6,>L5
    000026                                      *                   BUFFERNAME @I@ := VALUE;
    000027  0020    9840 FFE4                             LDR       1,D+5
    000028  0022    E840 FFE1                             LDR       6,D+4
    000029  0024    EF18 0001                             STR       6,*<D+1,$R1
    000030                                      *             'END';
    000031                                      *       'END';
    000032  0026    8AC0 FFDE                             INC       D+5
    000033  0028    0FF3                                  B         >L4
    000034                                      *       'END';      (OF SEGMENT)
    000035                                      *       'FINISH'
    000036  0029    83C8 FFD6                   L5        JMP       *D
    000037          002B                        L1        EQU       $
    000038  002B    83C0 0000       P                     JMP       ZK$EX
    000039                                                XLOC      ZK$EX
    000040                                                CTRL LINK ZK$EX
    000041  002D    0009                                  END       ZVKJFB,ZVKJFB
0000  ERR COUNT
@@L
```

```
E   ZVKJFC.LIST                          05/19/78  1423.0R W 05/19/78  1423.0       2646900000
    000001                                          TITLE     ZVKJFC,'REV 00'
    000002          0009                            XDEF      JFC
    000003          0005                            XDEF      ZVKJFC
    000004          0008                ZKCOM       COMM      8
    000005  0000    0000                D           RESV      5,0
    000006          0005                ZVKJFC      EQU       $
    000007  0005    8F00 0001       K               SAVE      <ZKCOM+1,Z'0011'
            0007    0011
    000008  0008    0F9C                            B         >L1
    000009                              *       'PROCEDURE' CREATE FUNCTION CODES ('VALUE' 'INTEGER' CHANNEL);
    000010                              *       'BEGIN'
    000011                              *           'INTEGER' I;
    000012  0009    DFC0 FFF6           JFC         STB       5,D
    000013  000B    E807                            LDR       6,$B7
    000014  000C    EF40 FFF4                       STR       6,D+1
    000015  000E    8740 FFF3           L3          CL        D+2
    000016                              *               'FOR' I:=0 'STEP' 1 'UNTIL' 16 'DO'
    000017                              *               'BEGIN'
    000018  0010    E840 FFF1           L4          LDR       6,D+2
    000019  0012    6EF0                            ADV       6,-16
    000020  0013    6A0F                            BGZ       6,>L5
    000021                              *                     IO WORD@I@ := 'BITS' @7,0@ IO WORD@I@
    000022  0014    E840 FFEC                       LDR       6,D+1
    000023  0016    9840 FFEB                       LDR       1,D+2
    000024  0018    C810 0000       X               LDR       4,<IOWORD,$R1
    000025  001A    C570 007F                       AND       4,=Z'007F'
    000026  001C    E454                            OR        6,=$R4
    000027  001D    EF10 0000       X               STR       6,<IOWORD,$R1
    000028                              *                   'UNION' CHANNEL;
    000029                              *               'END';
    000030                              *       'END';
    000031  001F    8AC0 FFE2                       INC       D+2
    000032  0021    0FEF                            B         >L4
    000033                              *       'END';      (OF SEGMENT)
    000034                              *       'FINISH'
    000035  0022    83C8 FFDD           L5          JMP       *D
    000036          0024                L1          EQU       $
    000037  0024    83C0 0000       P               JMP       ZK$EX
    000038                                          XLOC      ZK$EX
    000039                                          CTRL LINK ZK$EX
    000040                                          XLOC      IOWORD
    000041  0026    0005                            END       ZVKJFC,ZVKJFC
0000  ERR COUNT
e@L
```

```
E   ZVKJGC.LIST                    05/19/78  1423.0R W 05/19/78  1422.4      1921770000
000001                                          TITLE     ZVKJGC,'REV 00'
000002          0012                            XDEF      JGC
000003          000D                            XDEF      ZVKJGC
000004          0008                ZKCOM       COMM      8
000005  0000    0000                D           RESV      13,0
000006          000D                ZVKJGC      EQU       $
000007  000D    8F00 0001       K               SAVE      <ZKCOM+1,Z'0011'
        000F    0011
000008  0010    0F81 0109                       B         L1
000009                              *       'PROCEDURE' GENERATE CHECKSUMS ('VALUE''INTEGER' READ OR PUNCH);
000010                              *       'BEGIN'
000011                              *             'INTEGER' J,I;
000012                              *             'INTEGER' COLUMN CHECK;
000013                              *             'INTEGER' MASK;
000014                              *             'INTEGER' ROW CHECK;
000015                              *             'INTEGER' TEMP;
000016  0012    DFC0 FFED               JGC       STB       5,D
000017  0014    E807                              LDR       6,$B7
000018  0015    EF40 FFEB                         STR       6,D+1
000019                              *             'IF' READ OR PUNCH 'NE' 0
000020                              *             'THEN'
000021                              *             'BEGIN'
000022  0017    6936                              BEZ       6,>L3
000023                              *                   'INTEGER' BYTES;
000024                              *                   BYTES := 171;
000025  0018    C870 00AB                         LDR       4,=171
000026  001A    CF40 FFED                         STR       4,D+8
000027                              *                   'IF' 'BITS' @8,8@ LABEL 1 'EQ' 'LITERAL' (F)
000028                              *                   'THEN'
000029  001C    D2C0 0000       P                 LLH       5,LABEL1
000030  001E    5D46                              CMV       5,70
000031  001F    0984                              BNE       >L4
000032                              *                         BYTES := 16;
000033  0020    7C10                              LDV       7,16
000034  0021    FF40 FFE6                         STR       7,D+8
000035          0023                L4          EQU       $
000036                              *                   READ STATUS := FALSE;
000037  0023    8740 0000       P               *   CL        READST
000038                              *                     CHECK STATE (SHOULD BE ON);
000039  0025    FBC0 FFE3                         LAB       7,D+9
000040  0027    6C01                              LDV       6,1
000041  0028    EF07                              STR       6,$B7
000042  0029    D3C0 0000       P                 LNJ       5,JCS
000043                              *                   'IF' 'BITS' @1,0@ CARDS READ 'EQ' 1
000044                              *                   'THEN'
000045  002B    E840 0000       P                 LDR       6,CARDSR
000046  002D    6B11                              BEVN      6,>L5
000047  002E    FBC0 FFDA                         LAB       7,D+9
000048  0030    9BC0 0000       P                 LAB       1,BUFF2
000049  0032    9F87                              STB       1,$B7
000050                              *                         IOLD (BUFFER2 @0@,READ OR PUNCH,BYTES)
000051  0033    ABC0 FFCD                         LAB       2,D+1
000052  0035    AFC0 FFD4                         STB       2,D+10
000053                              *                   'ELSE'
000054  0037    C840 FFD0                         LDR       4,D+8
000055  0039    CF40 FFD1                         STR       4,D+11
000056  003B    D3C0 0000       P                 LNJ       5,JIL
000057  003D    0F90                              B         >L3
000058  003E    FBC0 FFCA           L5            LAB       7,D+9
000059  0040    9BC0 0000       P                 LAB       1,BUFF1
000060  0042    9F87                              STB       1,$B7
000061  0043    ABC0 FFBD                         LAB       2,D+1
000062  0045    AFC0 FFC4                         STB       2,D+10
000063                              *                         IOLD (BUFFER1 @0@,READ OR PUNCH,BYTES);
000064  0047    E840 FFC0                         LDR       6,D+8
000065  0049    EF40 FFC1                         STR       6,D+11
000066  004B    D3C0 0000       P                 LNJ       5,JIL
000067                              *             'END';
000068                              **;
000069                              **        ROW CHECKSUMS;
000070                              **;
000071          004D                L3          EQU       $
000072                              *             CHECKSUM @5@ :=0;
000073  004D    8740 0005       P               *   CL        CHECKS+5
000074                              *             MASK := 1;
000075  004F    6C01                              LDV       6,1
000076  0050    EF40 FFB4                         STR       6,D+5
000077                              *             BIT COUNT := 0;
000078  0052    8740 0000       P               *   CL        BITCOU
000079          0054                L7          EQU       $
000080                              *             'FOR' J:= 0 'STEP' 1 'UNTIL' 11
000081  0054    8740 FFAD                       *   CL        D+2
000082                              *             'DO'
000083                              *             'BEGIN'
000084  0056    E840 FFAB           L8          LDR       6,D+2
000085  0058    6EF5                            ADV       6,-11
000086  0059    6A01 003E                       BGZ       6,L9
000087                              *                   ROW CHECK := 0;
000088  005B    8740 FFAA                       *   CL        D+6
000089  005D    8740 FFA5           L10         *   CL        D+3
000090                              *                   'FOR' I:= 0 'STEP' 1 'UNTIL' NUMBER OF COLUMNS - 1
000091                              *                   'DO'
000092                              *                   'BEGIN'
000093  005F    E840 0000       P               *   LDR       6,NUMBER
000094  0061    6EFF                            ADV       6,-1
000095  0062    EF40 FFA5                       STR       6,D+8
000096  0064    E840 FF9E           L11         LDR       6,D+3
000097  0066    E240 FFA1                       SUB       6,D+8
000098  0068    6A20                            BGZ       6,>L12
000099                              *                         'IF' 'BITS' @1,0@ CARDS READ 'EQ' 0
000100                              *                         'THEN'
000101  0069    E840 0000       P               *   LDR       6,CARDSR
000102  006B    6B8A                            BODD      6,>L13
000103                              *                               TEMP := BUFFER2 @I@ 'MASK'MASK
000104                              *                         'ELSE'
000105  006C    C840 FF98                       *   LDR       4,D+5
000106  006E    9840 FF94                       *   LDR       1,D+3
000107  0070    C510 0000       X               *   AND       4,<BUFF2.$R1
000108  0072    CF40 FF94                       *   STR       4,D+7
000109  0074    0F89                            *   B         >L14
000110  0075    E840 FF8F           L13         *   LDR       6,D+5
000111  0077    9840 FF8B                       *   LDR       1,D+3
```

```
000112  0079  E510 0000      X             AND     6,<BUFF1.$R1
000113  007B  EF40 FF8B                    STR     6,D+7
000114                            *                       TEMP := BUFFER1 @I@ 'MASK' MASK;
000115                            *               'IF' TEMP 'NE' 0
000116        007D                L14         EQU     $
000117                            *               'THEN'
000118                            *               'BEGIN'
000119  007D  E840 FF89                    LDR     6,D+7
000120  007F  6906                        BEZ     6,>L15
000121                            *                       ROW CHECK := ROW CHECK 'DIFFER' 1;
000122  0080  8B40 FF85                    LBC     D+6,Z'0001'
        0082  0001
000123                            *                       BIT COUNT := BIT COUNT +1;
000124  0083  8AC0 0000      P             INC     BITCOU
000125                            *                   'END';
000126                            *                 'END';
000127  0085  8AC0 FF7D                L15 INC     D+3
000128  0087  0FDD                        B       >L11
000129        0088                L12     EQU     $
000130                            *                 CHECKSUM @5@ := CHECKSUM @5@*2 + ROWCHECK;(SHIFT LEFT)
000131  0088  E800 0005      X             LDR     6,<CHECKS+5
000132  008A  6001                        SOL     6,1
000133  008B  EA40 FF7A                    ADD     6,D+6
000134  008D  EF40 0005      P             STR     6,CHECKS+5
000135  008F  C840 FF75                    LDR     4,D+5
000136  0091  4001                        SOL     4,1
000137  0092  CF40 FF72                    STR     4,D+5
000138                            *                 MASK := MASK *2;                    (SHIFT LEFT)
000139                            *             'END';                                (END OF J FOR )
000140                            **;
000141                            **        COLUMN CHECKSUMS;
000142                            **;
000143  0094  8AC0 FF6D                    INC     D+2
000144  0096  0F81 FFBF                    B       L8
000145        0098                L9      EQU     $
000146  0098  8740 FF6A                L16 CL      D+3
000147                            *           'FOR' I:=0 'STEP' 1 'UNTIL' NUMBER OF COLUMNS - 1
000148                            *           'DO'
000149                            *           'BEGIN'
000150  009A  E840 0000      P             LDR     6,NUMBER
000151  009C  6EFF                        ADV     6,-1
000152  009D  EF40 FF6A                    STR     6,D+8
000153  009F  E840 FF63                L17 LDR     6,D+3
000154  00A1  E240 FF66                    SUB     6,D+8
000155  00A3  6A01 003D                    BGZ     6,L18
000156                            *                 'INTEGER' K,TEMP;
000157                            *                 'IF' 'BITS' @1,0@ CARDS READ 'EQ' 0
000158                            *                 'THEN'
000159  00A5  E840 0000      P             LDR     6,CARDSR
000160  00A7  6B88                        BODD    6,>L19
000161                            *                       TEMP := BUFFER2 @I@
000162                            *                 'ELSE'
000163  00A8  9840 FF5A                    LDR     1,D+3
000164  00AA  C810 0000      X             LDR     4,<BUFF2.$R1
000165  00AC  CF40 FF5D                    STR     4,D+10
000166  00AE  0F87                        B       >L20
000167        00AF                L19     EQU     $
000168                            *                       TEMP := BUFFER1 @I@;
000169  00AF  9840 FF53                    LDR     1,D+3
000170  00B1  E810 0000      X             LDR     6,<BUFF1.$R1
000171  00B3  EF40 FF56                    STR     6,D+10
000172        00B5                L20     EQU     $
000173                            *                 COLUMN CHECK := 'BITS' @1,0@ TEMP;
000174  00B5  E840 FF54                    LDR     6,D+10
000175  00B7  E570 0001                    AND     6,=Z'0001'
000176  00B9  EF40 FF4A                    STR     6,D+4
000177        00BB                L21     EQU     $
000178                            *                 'FOR' J:=0 'STEP' 1 'UNTIL' 10
000179  00BB  8740 FF46                    CL      D+2
000180                            *                 'DO'
000181                            *                 'BEGIN'
000182  00BD  E840 FF44                L22 LDR     6,D+2
000183  00BF  6EF6                        ADV     6,-10
000184  00C0  6A10                        BGZ     6,>L23
000185                            *                       TEMP := 'BITS' @11,1@ TEMP;
000186  00C1  E840 FF48                    LDR     6,D+10
000187  00C3  6004                        SOL     6,4
000188  00C4  6045                        SOR     6,5
000189  00C5  EF40 FF44                    STR     6,D+10
000190                            *                       COLUMN CHECK := COLUMN CHECK,DIFFER''BITS'@1,0@TEMP;
000191  00C7  E570 0001                    AND     6,=Z'0001'
000192  00C9  E640 FF3A                    XOR     6,D+4
000193  00CB  EF40 FF38                    STR     6,D+4
000194                            *                 'END';
000195  00CD  8AC0 FF34                    INC     D+2
000196  00CF  0FEE                        B       >L22
000197        00D0                L23     EQU     $
000198                            *                 K := I/16;                    (FIND CORRECT CHECKWORD)
000199  00D0  E840 FF32                    LDR     6,D+3
000200  00D2  6064                        SAR     6,4
000201  00D3  EF40 FF35                    STR     6,D+9
000202  00D5  9856                        LDR     1,=$R6
000203  00D6  C810 0000      X             LDR     4,<CHECKS.$R1
000204  00D8  4001                        SOL     4,1
000205  00D9  CA40 FF2A                    ADD     4,D+4
000206  00DB  CF10 0000      X             STR     4,<CHECKS.$R1
000207                            *                 CHECKSUM @K@ := CHECKSUM @K@ *2 + COLUMN CHECK;
000208                            *             'END';
000209  00DD  8AC0 FF25                    INC     D+3
000210  00DF  0F81 FFBF                    B       L17
000211                            *           'IF' I 'EQ' 40
000212        00E1                L18     EQU     $
000213                            *           'THEN'
000214  00E1  E840 FF21                    LDR     6,D+3
000215  00E3  6D28                        CMV     6,40
000216  00E4  0986                        BNE     >L24
000217                            *                 CHECKSUM @2@ := CHECKSUM @2@ *256; (SHIFT LEFT 8 PLACES)
000218  00E5  C800 0002      X             LDR     4,<CHECKS+2
000219  00E7  4008                        SOL     4,8
000220  00E8  CF40 0002      P             STR     4,CHECKS+2
000221        00EA                L24     EQU     $
000222                            *           CHECKSUM@6@ := BIT COUNT;
000223  00EA  E840 0000      P             LDR     6,BITCOU
```

```
000224  00EC  EF40 0006      P                STR       6,CHECKS+6
000225                       *                   'IF' R OR V MODE 'EQ' 'LITERAL' (R)
000226  00EE  C840 0000      P                LDR       4,RORVMD
000227  00F0  4D52                            CMV       4,82
000228  00F1  0906                            BE        >L25
000229                       *                   'OR' RESPONSE 'EQ' 'HEX' (AAAA)
000230                       *                   'THEN'
000231                       *                   'BEGIN'
000232  00F2  D840 0000      P                LDR       5,RESPON
000233  00F4  D970 AAAA                        CMR       5,=-21846
000234  00F6  0992                            BNE       >L26
000235        00F7                     L25     EQU       $
000236                       *                      'INTEGER' K;
000237        00F7                     L27     EQU       $
000238                       *                      'FOR' K := 0 'STEP' 1 'UNTIL' 6
000239  00F7  8740 FF10                        CL        D+8
000240                       *                      'DO'
000241                       *                      'BEGIN'
000242  00F9  E840 FF0E              L28       LDR       6,D+8
000243  00FB  6EFA                             ADV       6,-6
000244  00FC  6A0C                             BGZ       6,>L26
000245                       *                         @LOW ADDRESS@ := CHECKSUM @K@;
000246  00FD  9840 FF0A                        LDR       1,D+8
000247  00FF  E810 0000      X                 LDR       6,<CHECKS.$R1
000248  0101  EF48 0000      P                 STR       6,*LOWADD
000249                       *                            LOW ADDRESS := LOW ADDRESS + 1;
000250  0103  8AC0 0000      P                 INC       LOWADD
000251                       *                      'END';
000252  0105  8AC0 FF02                        INC       D+8
000253  0107  0FF2                             B         >L28
000254                       *                   'END';
000255        0108                     L26     EQU       $
000256                       *                   HAVE GENERATED CHECKSUMS := TRUE;
000257  0108  6C01                             LDV       6,1
000258  0109  EF40 0000      P                 STR       6,HAVEGE
000259  010B  FBC0 FEFC                        LAB       7,D+8
000260  010D  9BC0 0000      P                 LAB       1,STATWA
000261  010F  9F87                             STB       1,$B7
000262                       *                !INOUT (STATUS WAS, INPUT STATUS WORD @0@);     (WAIT TIL DONE)
000263                       *             'END';
000264  0110  0F00 000C      X       N1        NOP       <IOWORD+12
000265  0112  C840 FFFE                        LDR       4,N1+1
000266  0114  CF40 FEF4                        STR       4,D+9
000267  0116  D3C0 0000      P                 LNJ       5,JIO
000268                       *             'END';
000269                       *             'FINISH'
000270  0118  83C8 FEE7                        JMP       *D
000271        011A                     L1      EQU       $
000272  011A  83C0 0000      P                 JMP       ZK$EX
000273                                         XLOC      ZK$EX
000274                                         CTRL LINK ZK$EX
000275                                         XLOC      STATWA
000276                                         XLOC      READST
000277                                         XLOC      NUMBER
000278                                         XLOC      LABEL1
000279                                         XLOC      CARDSR
000280                                         XLOC      BITCOU
000281                                         XLOC      LOWADD
000282                                         XLOC      RORVMD
000283                                         XLOC      HAVEGE
000284                                         XLOC      RESPON
000285                                         XLOC      CHECKS
000286                                         XLOC      BUFF1
000287                                         XLOC      BUFF2
000288                                         XLOC      IOWORD
000289                                         XLOC      JIO
000290                                         XLOC      JIL
000291                                         XLOC      JCS
000292  011C  000D                             END       ZVKJGC,ZVKJGC
0000  ERR COUNT
@@L
```

```
E   ZVKJIO.LIST                          05/19/78  1423.1R W 05/19/78  1423.1      555750000
    000001                                                TITLE     ZVKJIO,'REV 00'
    000002          001A                                  XDEF      JIO
    000003          0015                                  XDEF      ZVKJIO
    000004          0008                        ZKCOM     COMM      8
    000005  0000    0000                        D         RESV      8,0
    000006  0008    0018                        T10       DC        24
    000007  0009    2C20 494F 2049                        TEXT      ', IO INSTRUCTION N'
            000C    4E53 5452 5543
                    5449 4F4E 204E
    000008  0012    414B 2744 2024                        TEXT      'AK''D $'
    000009          0015                        ZVKJIO    EQU       $
    000010  0015    8F00 0001       K                     SAVE      <ZKCOM+1,Z'0011'
            0017    0011
    000011  0018    0F81 0041                             B         L1
    000012                                      *       'PROCEDURE' INOUT ('LOCATION' 'INTEGER' X,Y);
    000013                                      *       'BEGIN'
    000014                                      *           'INTEGER' I,J;
    000015  001A    DFC0 FFE5                   JIO       STB       5,D
    000016  001C    8C87                                  LDI       $B7
    000017  001D    8D40 FFE3                             SDI       D+1
    000018                                      *             LABEL2 := 'LITERAL' (IN);
    000019  001F    C870 494E                             LDR       4,=18766
    000020  0021    CF40 0000       P                     STR       4,LABEL2
    000021  0023    6C01                        L3        LDV       6,1
    000022  0024    EF40 FFDF                             STR       6,D+4
    000023                                      *           'FOR' J:=1 'STEP' 1 'UNTIL' SPEED / 4      'DO'
    000024  0026    C840 0000       P                     LDR       4,SPEED
    000025  0028    4062                                  SAR       4,2
    000026  0029    CF40 FFDB                             STR       4,D+5
    000027  002B    E840 FFD8                   L4        LDR       6,D+4
    000028  002D    E240 FFD7                             SUB       6,D+5
    000029  002F    6A01 0018                             BGZ       6,L5
    000030          0031                        L6        EQU       $
    000031                                      *           'FOR' I:=1 'STEP' 1 'UNTIL' 'HEX' (7FFF)
    000032  0031    6C01                                  LDV       6,1
    000033  0032    EF40 FFD0                             STR       6,D+3
    000034                                      *           'DO' 'COMMENT' INITIALIZE LOOP;
    000035                                      *           'BEGIN'
    000036  0034    E840 FFCE                   L7        LDR       6,D+3
    000037  0036    E270 7FFF                             SUB       6,=32767
    000038  0038    6A01 000B                             BGZ       6,L8
    000039                                      *                   'CODE' 'BEGIN']
    000040  003A    8048 FFC6                             IO        *D+1,*D+2
            003C    0048 FFC5
    000041  003E    0701 0019                             BIOT      L9
    000042                                      *           'END';
    000043  0040    8AC0 FFC2                             INC       D+3
    000044  0042    0F81 FFF1                             B         L7
    000045  0044    8AC0 FFBF                   L8        INC       D+4
    000046  0046    0F81 FFE4                             B         L4
    000047          0048                        L5        EQU       $
    000048                                      *                   'CODE''BEGIN']
    000049                                      *                   STB $B5,+$A       ADDRESS OF CALLING ROUTINE
    000050                                      *                   LNJ $B5,<ZV$ER     REPORT ERROR
    000051  0048    DFC0 0000       T                     STB       $B5,+$A
    000052  004A    D380 0000       X                     LNJ       $B5,<ZV$ER
    000053                                      *                   ]CB]
    000054                                      *                   XLOC ZV$ER
    000055                                      *                   B >+$B
    000056                                                XLOC      ZV$ER
    000057  004C    0F80            T                     B         >+$B
    000058                                      *                   ]CB]
    000059                                      *         $A        DC <$
    000060  004D    004D                        $A        DC        <$
    000061  004E            0000    X                     DC        <LABEL1
    000062                                      *                   ]CB]
    000063                                      *         $B        ORG $]
    000064  004F                                $B        ORG       $
    000065                                      *             PUT TEXT (", IO INSTRUCTION NAK'D $");
    000066  004F    FBC0 FFB5                             LAB       7,D+5
    000067  0051    9BC0 FFB6                             LAB       1,T10
    000068  0053    9F87                                  STB       1,$B7
    000069  0054    D3C0 0000       P                     LNJ       5,VKPT
    000070                                      *             'GO TO' FRESH START;
    000071  0056    83C0 0000       P                     JMP       FRESHS
    000072                                      *         NEXT; 'END';
    000073                                      *         'END';    (OF SEGMENT)
    000074                                      *         'FINISH'
    000075  0058    83C8 FFA7                   L9        JMP       *D
    000076          005A                        L1        EQU       $
    000077  005A    83C0 0000       P                     JMP       ZK$EX
    000078                                                XLOC      ZK$EX
    000079                                                CTRL LINK ZK$EX
    000080                                                XLOC      VKPT
    000081                                                XLOC      LABEL1
    000082                                                XLOC      LABEL2
    000083                                                XLOC      SPEED
    000084                                                XLOC      FRESHS
    000085  005C    0015                                  END       ZVKJIO,ZVKJIO
0000  ERR COUNT
L
```

```
 ZVKJIL.LIST                          05/19/78  1423.1R W 05/19/78  1423.1       618390000
000001                                             TITLE     ZVKJIL,'REV 00'
000002          0016                               XDEF      JIL
000003          0011                               XDEF      ZVKJIL
000004          0008                    ZKCOM      COMM      8
000005  0000    0000                    D          RESV      9,0
000006  0009    000D                    T7         DC        13
000007  000A    2C20 494F 4C44                     TEXT      ', IOLD NAK''D$'
        000D    204E 414B 2744
                2400
000008          0011                    ZVKJIL     EQU       $
000009  0011    8F00 0001       K                  SAVE      <ZKCOM+1,Z'0011'
        0013    0011
000010  0014    0F81 0047                          B         L1
000011                                  *        'PROCEDURE' IOLD ('LOCATION' 'INTEGER' Z,Y;'VALUE' 'INTEGER' X);
000012                                  *         'BEGIN'
000013                                  *            'INTEGER' I;
000014  0016    DFC0 FFE9               JIL        STB       5,D
000015  0018    8F87                               RSTR      $B7,Z'0700'
        0019    0700
000016  001A    8F40 FFE6                          SAVE      D+1,Z'0700'
        001C    0700
000017                                  *              LABEL2 := 'LITERAL' (IO);
000018  001D    C870 494F                          LDR       4,=18767
000019  001F    CF40 0000       P                  STR       4,LABEL2
000020  0021    6C01                    L3         LDV       6,1
000021  0022    EF40 FFE1                          STR       6,D+4
000022                                  *            'FOR' I:=1 'STEP' 1 'UNTIL' 'HEX'(7FFF) 'DO'
000023                                  *            'BEGIN'
000024  0024    E840 FFDF               L4         LDR       6,D+4
000025  0026    E270 7FFF                          SUB       6,=32767
000026  0028    6A01 000D                          BGZ       6,L5
000027                                  *                   'CODE''BEGIN']
000028  002A    81C8 FFD6                          IOLD      *D+1,*D+2,D+3
        002C    0048 FFD5
        002E    0040 FFD4
000029  0030    0701 0015                          BIOT      L6
000030                                  *              'END';
000031  0032    8AC0 FFD1                          INC       D+4
000032  0034    0F81 FFEF                          B         L4
000033          0036                    L5         EQU       $
000034                                  *                   'CODE''BEGIN']
000035                                  *                   STB $B5,+$A       ADDRESS OF CALLING ROUTINE
000036                                  *                   LNJ $B5,<ZV$ER     REPORT ERROR
000037  0036    DFC0 0000       T                  STB       $B5,+$A
000038  0038    D380 0000       X                  LNJ       $B5,<ZV$ER
000039                                  *                   ]CB]
000040                                  *                   XLOC ZV$ER
000041                                  *                   B >+$B
000042                                             XLOC      ZV$ER
000043  003A    0F80            T                  B         >+$B
000044                                  *                   ]CB]
000045                                  *          $A       DC <$
000046  003B    003B                    $A         DC        <$
000047  003C            0000    X                  DC        <LABEL1
000048                                  *                   ]CB]
000049                                  *          $B       ORG $]
000050  003D                            $B         ORG       $
000051  003D    FBC0 FFC7                          LAB       7,D+5
000052  003F    9BC0 FFC9                          LAB       1,T7
000053  0041    9F87                               STB       1,$B7
000054  0042    D3C0 0000       P                  LNJ       5,VKPT
000055                                  *               PUT TEXT (", IOLD NAK'D$");
000056                                  *              'GO TO' ASK NEXT;
000057                                  *          NEXT;
000058  0044    83C0 0000       P                  JMP       ASKNEX
000059          0046                    L6         EQU       $
000060                                  *              'IF' READ STATUS 'EQ' TRUE
000061                                  *              'THEN'
000062  0046    E840 0000       P                  LDR       6,READST
000063  0048    6D01                               CMV       6,1
000064  0049    098E                               BNE       >L9
000065  004A    FBC0 FFBA                          LAB       7,D+5
000066  004C    9BC0 0000       P                  LAB       1,STATWA
000067  004E    9F87                               STB       1,$B7
000068                                  *                   INOUT (STATUS WAS,INPUT STATUS WORD @0@);
000069  004F    0F00 000C       X       N1         NOP       <IOWORD+12
000070  0051    C840 FFFE                          LDR       4,N1+1
000071  0053    CF40 FFB2                          STR       4,D+6
000072  0055    D3C0 0000       P                  LNJ       5,JIO
000073          0057                    L9         EQU       $
000074                                  *              READ STATUS := TRUE;
000075                                  *          'END';
000076  0057    6C01                               LDV       6,1
000077  0058    EF40 0000       P                  STR       6,READST
000078                                  *          'END';     (OF SEGMENT)
000079                                  *          'FINISH'
000080  005A    83C8 FFA5                          JMP       *D
000081          005C                    L1         EQU       $
000082  005C    83C0 0000       P                  JMP       ZK$EX
000083                                             XLOC      ZK$EX
000084                                             CTRL LINK ZK$EX
000085                                             XLOC      VKPT
000086                                             XLOC      STATWA
000087                                             XLOC      READST
000088                                             XLOC      LABEL1
000089                                             XLOC      LABEL2
000090                                             XLOC      IOWORD
000091                                             XLOC      JIO
000092                                             XLOC      ASKNEX
000093  005E    0011                               END       ZVKJIL,ZVKJIL
0000  ERR COUNT
L
```

```
 ZVKJIR.LIST                              05/19/78  1423.1R W 05/19/78  1422.2      2312730000
000001                                                TITLE     ZVKJIR,'REV 00'
000002          0064                                  XDEF      TIMER
000003          0081                                  XDEF      CPRUPT
000004          0092                                  XDEF      DEVIH
000005          004F                                  XDEF      LEV20
000006          005A                                  XDEF      LEV21
000007          0048                                  XDEF      ZVKJIR
000008          0008                        ZKCOM     COMM      8
000009  0000    0000                        D         RESV      6,0
000010  0006    000E                        T4        DC        14
000011  0007    2C20 5354 5543                        TEXT      ', STUCK AT 20$'
        000A    4B20 4154 2032
                3024
000012  000E    0011                        T7        DC        17
000013  000F    2C20 4350 2052                        TEXT      ', CP RUPT DEVICE$'
        0012    5550 5420 4445
                5649 4345 2400
000014  0018    0022                        T10       DC        34
000015  0019    2C20 4E4F 5420                        TEXT      ', NOT READER OR PU'
        001C    5245 4144 4552
                204F 5220 5055
000016  0022    4E43 4820 5448                        TEXT      'NCH THAT RUPTED$'
        0025    4154 2052 5550
                5445 4424
000017  002A    0011                        T13       DC        17
000018  002B    2C20 4953 4120                        TEXT      ', ISA SREG WRONG$'
        002E    5352 4547 2057
                524F 4E47 2400
000019  0034    000E                        T23       DC        14
000020  0035    2C20 464C 4147                        TEXT      ', FLAG NOT ON$'
        0038    204E 4F54 204F
                4E24
000021  003C    0015                        T26       DC        21
000022  003D    2C20 4E4F 5420                        TEXT      ', NOT DEV OR CP FL'
        0040    4445 5620 4F52
                2043 5020 464C
000023  0046    4147 2400                             TEXT      'AG$'
000024          0048                        ZVKJIR    EQU       $
000025  0048    8F00 0001       K                     SAVE      <ZKCOM+1,Z'0011'
        004A    0011
000026                                                *             'INTEGER' TEMP,I, PROGRAM LEVEL;
000027                                                *           ZHIAFB := ZHIAFB;       (FORCE XLOC)
000028                                                **;
000029                                                ** CLOCK HANDLER;
000030                                                **;
000031                                                *       LEV20:
000032  004B    E840 0000       P                       LDR       6,ZHIAFB
000033  004D    EF40 0000       P                       STR       6,ZHIAFB
000034                                                ** REMEMBER THAT THE CLOCK RUPTED;
000035          004F                        LEV20     EQU       $
000036                                                *             CLOCK SEMAPHORE := CLOCKSEMAPHORE + 1;
000037  004F    8AC0 0000       P                       INC       CKSEM
000038  0051    0005                                    DC        5
000039                                                *             CLOCK OFF;
000040  0052    E870 803F                               LDR       6,=-32705
000041  0054    EF40 0000       P                       STR       6,LEVTYP
000042                                                *             LEV ('HEX' (8000),63);(SCHEDULE LEVEL, SUSPEND, SCAN AND DISPATCH)
000043  0056    8E40 0000       P                       LEV       LEVTYP
000044                                                *             'GO TO' LEV20;
000045                                                **;
000046                                                ** DEVICE INTERRUPT HANDLER - L21 HANDLER;
000047                                                **;
000048                                                *       LEV21:
000049  0058    83C0 FFF6                               JMP       LEV20
000050                                                ** REMEMBER THAT YOU'VE BEEN HERE;
000051          005A                        LEV21     EQU       $
000052                                                *             L21SEMAPHORE := L21SEMAPHORE + 1;
000053  005A    8AC0 0000       P                       INC       L21SEM
000054  005C    E870 803F                               LDR       6,=-32705
000055  005E    EF40 0000       P                       STR       6,LEVTYP
000056                                                *             LEV ('HEX' (8000),63);
000057  0060    8E40 0000       P                       LEV       LEVTYP
000058                                                *             'GO TO' LEV21;
000059                                                **;
000060                                                ** CALIBRATION TIMER INTERRUPT HANDLER;
000061                                                **;
000062                                                *       TIMER:
000063  0062    83C0 FFF7                               JMP       LEV21
000064          0064                        TIMER     EQU       $
000065                                                *             LABEL2 := 'LITERAL' (CI);
000066  0064    E870 4349                               LDR       6,=17225
000067  0066    EF40 0000       P                       STR       6,LABEL2
000068                                                *             'CODE' BEGIN
000069                                                *                   XLOC STOPTI,RUPTI
000070                                                            XLOC      STOPTI,RUPTI
000071  0068    9FC0 0000       P                       STB       $B1,TEMPCH
000072                                                *                   JCB]
000073                                                *                   LAB $B1,<STOPTI
000074  006A    9B80 0000       X                       LAB       $B1,<STOPTI
000075  006C    9FC0 0005       P                       STB       $B1,ISAP+4+$AF
000076                                                *                   JCB]
000077                                                *                   LDB $B1,]TEMP CHARACTER
000078  006E    9CC0 0000       P                       LDB       $B1,TEMPCH
000079  0070    E870 803F                               LDR       6,=-32705
000080  0072    EF40 0000       P                       STR       6,LEVTYP
000081                                                *             LEV ('HEX' (8000),63);  (GO BACK TO PROGRAM LEVEL)
000082  0074    8E40 0000       P                       LEV       LEVTYP
000083  0076    D3C0 0000       P                       LNJ       5,JRE
000084                                                *           REPORT ERROR;             (STUCK AT LEVEL 20)
000085  0078    FBC0 FF8A                               LAB       7,D+3
000086  007A    9BC0 FF8B                               LAB       1,T4
000087  007C    9F87                                    STB       1,$B7
000088  007D    D3C0 0000       P                       LNJ       5,VKPT
000089                                                *               PUT TEXT (", STUCK AT 20$");
000090                                                *             'GO TO' FRESH START;
000091                                                **;
000092                                                ** IF CP SHOULD RUPT DEVICE;;
000093                                                **;
000094                                                *       CP RUPT:
000095  007F    83C0 0000       P                       JMP       FRESHS
000096          0081                        CPRUPT    EQU       $
000097                                                *             LABEL2;= 'LITERAL' (CP);
```

```
000098  0081  E870 4350                    LDR     6,=17232
000099  0083  EF40 0000      P             STR     6,LABEL2
000100  0085  D3C0 0000      P             LNJ     5,JRE
000101                              *               REPORT ERROR;
000102  0087  FBC0 FF7B                    LAB     7,D+3
000103  0089  9BC0 FF84                    LAB     1,T7
000104  008B  9F87                        STB     1,$B7
000105  008C  D3C0 0000      P             LNJ     5,VKPT
000106                              *               PUT TEXT (", CP RUPT DEVICE$");
000107                              *               'GO TO' FRESH START;
000108  008E  83C0 0000      P             JMP     FRESHS
000109                              *               'GO TO ' CP RJPT;
000110                              **;
000111                              ** GO HERE AND CHECK RUPT CONDITIONS;
000112                              **;
000113                              *       DEVICE HANDLER;
000114  0090  83C0 FFF0                    JMP     CPRUPT
000115                              ** RESET RETURN LOCATION - AVOID RETURN TO BDEC;
000116  0092  9840 0000      P     DEVIH   LDR     1,ZV$AF
000117  0094  1E04                        ADV     1,4
000118  0095  A840 0000      P             LDR     2,ZV$AF
000119  0097  2E04                        ADV     2,4
000120                              *               ISA FOR PROGRAM @ZVDAF+4@:=ISA FOR PROGRAM @ZVDAF+4@+2;
000121  0098  E820 0000      X             LDR     6,<ISAP,$R2
000122  009A  6E02                        ADV     6,2
000123  009B  EF10 0000      X             STR     6,<ISAP,$R1
000124                              *               DEVICE SEMAPHORE := DEVICE SEMAPHORE + 1; (REMEMBER WAS HERE)
000125  009D  8AC0 0000      P             INC     DVSEM
000126                              *               TEMP := CHANNEL 'UNION' DEVICE LEVEL;
000127  009F  E840 0000      P             LDR     6,CHANNE
000128  00A1  E440 0000      P             OR      6,DLVL
000129  00A3  EF40 FF5C                    STR     6,D
000130                              *               'IF' ISARD @0@ 'MASK' 'HEX' (FFBF) 'NE' TEMP
000131                              *               'THEN'
000132                              *               'BEGIN'
000133  00A5  C800 0002      X             LDR     4,<ISAD+2
000134  00A7  C570 FFBF                    AND     4,=-65
000135  00A9  C956                        CMR     4,=$R6
000136  00AA  0901 0014                    BE      L9
000137                              *                       'INTEGER' DEV;
000138                              *                       LABEL2 := 'LITERAL' (WD);
000139  00AC  C870 5744                    LDR     4,=22340
000140  00AE  CF40 0000      P             STR     4,LABEL2
000141  00B0  D840 FF4F                    LDR     $R5,D
000142  00B2  C840 FF50                    LDR     $R4,D+3
000143  00B4  D3C0 0000      P             LNJ     5,JRE
000144                              *                       REPORT ERROR;
000145                              *                       PUT TEXT (", NOT READER OR PUNCH THAT RUPTED$");
000146  00B6  FBC0 FF4D                    LAB     7,D+4
000147  00B8  9BC0 FF5F                    LAB     1,T10
000148  00BA  9F87                        STB     1,$B7
000149  00BB  D3C0 0000      P             LNJ     5,VKPT
000150                              *               'GO TO' FRESH START;
000151  00BD  83C0 0000      P             JMP     FRESHS
000152                              *               'END';
000153        00BF                  L9      EQU     $
000154                              *               'CODE'('BEGIN')
000155                              *                       STS =$R7
000156                              *                       STR $R7,)TEMP
000157  00BF  8C57                        STS     =$R7
000158  00C0  FF40 FF3F                    STR     $R7,D
000159                              *               'IF' TEMP 'MASK' 'HEX'(63FF) 'NE' SREG 'MASK' 'HEX'(63FF) 'UNION'
000160                              *                       DEVICE LEVEL
000161  00C2  E840 FF3D                    LDR     6,D
000162  00C4  E570 63FF                    AND     6,=25599
000163                              *               'THEN'
000164                              *               'BEGIN'
000165  00C6  C840 0000      P             LDR     4,SREG
000166  00C8  C570 63FF                    AND     4,=25599
000167  00CA  C440 0000      P             OR      4,DLVL
000168  00CC  E954                        CMR     6,=$R4
000169  00CD  0901 0014                    BE      L12
000170                              *                       LABEL2 := 'LITERAL' (SR);
000171  00CF  E870 5352                    LDR     6,=21330
000172  00D1  EF40 0000      P             STR     6,LABEL2
000173  00D3  D840 0000      P             LDR     $R5,DLVL
000174  00D5  C840 0000      P             LDR     $R4,SREG
000175                              *                       LOAD R4 R5 (SREG, DEVICE LEVEL);
000176  00D7  D3C0 0000      P             LNJ     5,JRE
000177                              *                       REPORT ERROR;
000178  00D9  FBC0 FF29                    LAB     7,D+3
000179  00DB  9BC0 FF4E                    LAB     1,T13
000180  00DD  9F87                        STB     1,$B7
000181  00DE  D3C0 0000      P             LNJ     5,VKPT
000182                              *                       PUT TEXT (", ISA SREG WRONG$");
000183                              *                       'GO TO' FRESH START;
000184  00E0  83C0 0000      P             JMP     FRESHS
000185                              *               'END';
000186        00E2                  L12     EQU     $
000187                              *               'IF' CP LEVEL 'LE' DEVICE LEVEL
000188                              *               'THEN'
000189                              *                       PROGRAM LEVEL := 63
000190  00E2  E840 0000      P             LDR     6,CPLEVE
000191  00E4  E940 0000      P             CMR     6,DLVL
000192  00E6  0A05                        BAG     >L14
000193                              *               'ELSE'
000194  00E7  4C3F                        LDV     4,63
000195  00E8  CF40 FF19                    STR     4,D+2
000196  00EA  0F85                        B       >L15
000197        00EB                  L14     EQU     $
000198                              *                       PROGRAM LEVEL := CP LEVEL;
000199  00EB  E840 0000      P             LDR     6,CPLEVE
000200  00ED  EF40 FF14                    STR     6,D+2
000201        00EF                  L15     EQU     $
000202        00EF                  L16     EQU     $
000203                              *               'FOR' I:=0 'STEP' 1 'UNTIL' 63
000204  00EF  8740 FF11                    CL      D+1
000205                              *               'DO'
000206                              *               'BEGIN'
000207  00F1  E840 FF0F              L17     LDR     6,D+1
000208  00F3  6EC1                        ADV     6,-63
000209  00F4  6A01 0059                    BGZ     6,L18
000210                              *                       'INTEGER' FLAG;
```

```
000211                                  *                       'CODE''BEGIN']
000212  00F6  AF40 FF09                 *       STR             $R2,D
000213  00F8  A840 FF08                 *       LDR             $R2,D+1
000214                                  *                               ]CB]
000215  00FA  8740 FF08                 *       CL              D+3
000216  00FC  82A0 0000         X       *       LB              <ZHIAFB.$R2
000217                                  *                               ]CB]
000218                                  *                           BBF >+$A
000219  00FE  0580              I       *       BBF             >+$A
000220  00FF  8AC0 FF03                 *       INC             D+3
000221                                  *                               ]CB]
000222                                  *   $A                          LDR $R2,]TEMP
000223  0101  A840 FEFE                 $A      LDR             $R2,D
000224                                  *       'IF' I 'EQ' PROGRAM LEVEL
000225  0103  E840 FEFD                 *       LDR             6,D+1
000226  0105  E940 FEFC                 *       CMR             6,D+2
000227  0107  0905                      *       BE              >L19
000228                                  *       'OR' I 'EQ' DEVICE LEVEL
000229                                  *       'THEN'
000230                                  *       'BEGIN'
000231  0108  E940 0000         P       *       CMR             6,DLVL
000232  010A  0981 0028                 *       BNE             L20
000233        010C                      L19     EQU             $
000234                                  *                       'IF' 'BITS' @1,0@ FLAG 'EQ' 1
000235                                  *                       'THEN'
000236                                  *                       'BEGIN'
000237  010C  E840 FEF6                 *       LDR             6,D+3
000238  010E  6B01 0011                 *       BEVN            6,L21
000239                                  *                               'IF' PROGRAM LEVEL 'NE' I
000240                                  *                               'THEN'
000241                                  *                               'BEGIN'
000242  0110  C840 FEF1                 *       LDR             4,D+2
000243  0112  C940 FEEE                 *       CMR             4,D+1
000244  0114  0901 0035                 *       BE              L22
000245                                  *                                       'CODE''BEGIN']
000246  0116  AF40 FEE9                 *       STR             $R2,D
000247  0118  A840 FEE8                 *       LDR             $R2,D+1
000248                                  *                                               ]CB]
000249                                  *                                               LBF <ZHIAFB.$R2  TURN OFF FLAG
000250                                  *                                               LDR $R2,]TEMP
000251  011A  8820 0000         X       *       LBF             <ZHIAFB.$R2
000252  011C  A840 FEE3                 *       LDR             $R2,D
000253                                  *                               'END';
000254                                  *                       'END'
000255                                  *                       'ELSE'
000256                                  *                       'BEGIN'
000257  011E  0F81 002B                 *       B               L22
000258        0120                      L21     EQU             $
000259                                  *                               LABEL 2 ;= 'LITERAL' (FO);
000260  0120  E870 464F                 *       LDR             6,=17999
000261  0122  EF40 0000         P       *       STR             6,LABEL2
000262  0124  D840 0000         P       *       LDR             $R5,DLVL
000263  0126  C840 FEDA                 *       LDR             $R4,D+1
000264                                  *                               LOAD R4 R5 (I,DEVICE LEVEL);
000265  0128  D3C0 0000         P       *       LNJ             5,JRE
000266                                  *                               REPORT ERROR;
000267  012A  FBC0 FED9                 *       LAB             7,D+4
000268  012C  9BC0 FF07                 *       LAB             1,T23
000269  012E  9F87                      *       STB             1,$B7
000270  012F  D3C0 0000         P       *       LNJ             5,VKPT
000271                                  *                               PUT TEXT (', FLAG NOT ON$");
000272                                  *                               'GO TO' FRESH START;
000273  0131  83C0 0000         P       *       JMP             FRESHS
000274                                  *                       'END';
000275                                  *                       'END'
000276                                  *                       'ELSE'
000277                                  *                       'BEGIN'
000278        0133                      L20     EQU             $
000279                                  *                       'IF' 'BITS' @1,0@ FLAG 'EQ' 1
000280                                  *                       'THEN'
000281                                  *                       'BEGIN'
000282  0133  E840 FECF                 *       LDR             6,D+3
000283  0135  6B01 0014                 *       BEVN            6,L22
000284                                  *                               LABEL 2 ;= 'LITERAL' (WF);
000285  0137  C870 5746                 *       LDR             4,=22342
000286  0139  CF40 0000         P       *       STR             4,LABEL2
000287  013B  D840 0000         P       *       LDR             $R5,DLVL
000288  013D  C840 FEC3                 *       LDR             $R4,D+1
000289                                  *                               LOAD R4 R5 (I,DEVICE LEVEL);
000290  013F  D3C0 0000         P       *       LNJ             5,JRE
000291                                  *                               REPORT ERROR;
000292                                  *                               PUT TEXT (', NOT DEV OR CP FLAG$");
000293  0141  FBC0 FEC2                 *       LAB             7,D+4
000294  0143  9BC0 FEF8                 *       LAB             1,T26
000295  0145  9F87                      *       STB             1,$B7
000296  0146  D3C0 0000         P       *       LNJ             5,VKPT
000297                                  *                               'GO TO' FRESH START;
000298  0148  83C0 0000         P       *       JMP             FRESHS
000299                                  *                       'END';
000300                                  *                       'END';
000301                                  *               'END';  (OF FOR LOOP)
000302  014A  8AC0 FEB6                 L22     INC             D+1
000303  014C  0F81 FFA4                 *       B               L17
000304        014E                      L18     EQU             $
000305                                  *           'IF' DEVICE LEVEL 'EQ' CP LEVEL
000306                                  *           'THEN'
000307                                  *               'BEGIN'
000308  014E  E840 0000         P       *       LDR             6,DLVL
000309  0150  E940 0000         P       *       CMR             6,CPLEVE
000310  0152  0981 0005                 *       BNE             L27
000311                                  *                               'CODE''BEGIN']
000312                                  *                                   JMP <RUPTI
000313                                  *                               ]'END';
000314                                  *               'END'
000315  0154  8380 0000         X       *       JMP             <RUPTI
000316                                  *           'ELSE'
000317  0156  83C0 FF3B                 *       JMP             DEVIH
000318  0158  E840 0000         P       L27     LDR             6,CPLEVE
000319  015A  E470 8000                 *       OR              6,=-32768
000320  015C  EF40 0000         P       *       STR             6,LEVTYP
000321  015E  8E40 0000         P       *       LEV             LEVTYP
000322                                  *                       LEV ('HEX'(8000),CP LEVEL);
000323                                  *           'GO TO' DEVICE HANDLER;
```

```
000324                                  *       'END';
000325                                  *       'FINISH'
000326  0160  83C0 FF31                          JMP       DEVIH
000327                                          XLOC      ZV$AF
000328                                          CTRL LINK ZV$AF
000329                                          XLOC      VKPT
000330                                          XLOC      CHANNE
000331                                          XLOC      LABEL2
000332                                          XLOC      CKSEM
000333                                          XLOC      L21SEM
000334                                          XLOC      DLVL
000335                                          XLOC      DVSEM
000336                                          XLOC      CPLEVE
000337                                          XLOC      LEVTYP
000338                                          XLOC      TEMPCH
000339                                          XLOC      SREG
000340                                          XLOC      ZHIAFB
000341                                          XLOC      ISAD
000342                                          XLOC      ISAP
000343                                          XLOC      JRE
000344                                          XLOC      FRESHS
000345  0162  0048                             END       ZVKJIR,ZVKJIR
0000  ERR COUNT
@@L
```

```
E   ZVKJMC.LIST                          05/19/78  1424.9R W 05/19/78  1423.8       518400000
000001                                                 TITLE     ZVKJMC,'REV 00'
000002          000E                                   XDEF      JMC
000003          0009                                   XDEF      ZVKJMC
000004          0008                     ZKCOM         COMM      8
000005  0000    0000                     D             RESV      9,0
000006          0009                     ZVKJMC        EQU       $
000007  0009    8F00 0001       K                      SAVE      <ZKCOM+1,Z'0011'
        000B    0011
000008                                                 *         'PROCEDURE'  MICRO           ('INTEGER' 'ARRAY'  RAY);
000009                                                 *         'BEGIN'
000010  000C    0F81 0054                              B         L1
000011                                                 *             'INTEGER' INSTRUCTION COUNTER, I;
000012  000E    DFC0 FFF1                JMC           STB       5,D
000013  0010    8C87                                   LDI       $B7
000014  0011    8D40 FFEF                              SDI       D+1
000015                                                 *             INSTRUCTION COUNTER := RAY @0@;
000016  0013    C848 FFED                              LDR       4,*D+1
000017  0015    CF40 FFED                              STR       4,D+3
000018  0017    FBC0 FFED                              LAB       7,D+5
000019  0019    9BC0 0000       P                      LAB       1,IOTM
000020  001B    9F87                                   STB       1,$B7
000021                                                 *             INOUT (SET TEST MODE, OUTPUT CONTROL @0@);
000022  001C    ABC0 0000       P                      LAB       2,IOWORD
000023  001E    AFC0 FFE7                              STB       2,D+6
000024  0020    D3C0 0000       P                      LNJ       5,JIO
000025          0022                     L3            EQU       $
000026                                                 *             'FOR' I := 1 'STEP' 1 'UNTIL' INSTRUCTION COUNTER
000027  0022    6C01                                   LDV       6,1
000028  0023    EF40 FFE0                              STR       6,D+4
000029                                                 *             'DO'
000030  0025    C840 FFDD                              LDR       4,D+3
000031  0027    CF40 FFDD                              STR       4,D+5
000032  0029    E840 FFDA                L4            LDR       6,D+4
000033  002B    E240 FFD9                              SUB       6,D+5
000034  002D    6A11                                   BGZ       6,>L5
000035  002E    E840 FFD2                              LDR       6,D+1
000036  0030    EA40 FFD3                              ADD       6,D+4
000037  0032    FBC0 FFD3                              LAB       7,D+6
000038  0034    EF07                                   STR       6,$B7
000039                                                 *             INOUT ( RAY @I@, OUTPUT CONTROL @0@);
000040  0035    9BC0 0000       P                      LAB       1,IOWORD
000041  0037    9FC0 FFCF                              STB       1,D+7
000042  0039    D3C0 0000       P                      LNJ       5,JIO
000043  003B    8AC0 FFC8                              INC       D+4
000044  003D    0FEC                                   B         >L4
000045  003E    FBC0 FFC6                L5            LAB       7,D+5
000046  0040    9BC0 0000       P                      LAB       1,SCR
000047  0042    9F87                                   STB       1,$B7
000048  0043    ABC0 0000       P                      LAB       2,IOWORD
000049  0045    AFC0 FFC0                              STB       2,D+6
000050  0047    D3C0 0000       P                      LNJ       5,JIO
000051                                                 *             INOUT (SET CHANNEL READY, OUTPUT CONTROL @0@);
000052  0049    FBC0 FFBB                              LAB       7,D+5
000053  004B    9BC0 0000       P                      LAB       1,IOX1
000054  004D    9F87                                   STB       1,$B7
000055  004E    ABC0 0000       P                      LAB       2,IOWORD
000056  0050    AFC0 FFB5                              STB       2,D+6
000057  0052    D3C0 0000       P                      LNJ       5,JIO
000058                                                 *             INOUT (WAIT LOOP ADDRESS, OUTPUT CONTROL @0@);
000059  0054    FBC0 FFB0                              LAB       7,D+5
000060  0056    9BC0 0000       P                      LAB       1,IOX2
000061  0058    9F87                                   STB       1,$B7
000062  0059    ABC0 0000       P                      LAB       2,IOWORD
000063  005B    AFC0 FFAA                              STB       2,D+6
000064  005D    D3C0 0000       P                      LNJ       5,JIO
000065                                                 *             INOUT (RETURN FROM TEST MODE, OUTPUT CONTROL @0@);
000066                                                 *         'END';
000067                                                 *         'END';
000068                                                 *         'FINISH'
000069  005F    83C8 FFA0                              JMP       *D
000070          0061                     L1            EQU       $
000071  0061    83C0 0000       P                      JMP       ZK$EX
000072                                                 XLOC      ZK$EX
000073                                                 CTRL LINK ZK$EX
000074                                                 XLOC      SCR
000075                                                 XLOC      IOTM
000076                                                 XLOC      IOX1
000077                                                 XLOC      IOX2
000078                                                 XLOC      IOWORD
000079                                                 XLOC      JIO
000080  0063    0009                                   END       ZVKJMC,ZVKJMC
0000  ERR COUNT
```

```
ZVKJPD.LIST                              05/19/78  1425.1R W 05/19/78  1422.3      1529010000
000001                                              TITLE     ZVKJPD,'REV 00'
000002          0010                                XDEF      JPD
000003          000B                                XDEF      ZVKJPD
000004          0008                        ZKCOM   COMM      8
000005  0000    0000                        D       RESV      11,0
000006          000B                        ZVKJPD  EQU       $
000007  000B    8F00 0001       K                   SAVE      <ZKCOM+1,Z'0011'
        000D    0011
000008  000E    0F81 00D5                           B         L1
000009                                      *       'PROCEDURE' PUNCH A DECK ('VALUE''INTEGER' CARDS SPECIFIED,
000010                                      *                                 TYPE OF PUNCHES);
000011                                      *       'BEGIN'
000012                                      *           'INTEGER' N,I;
000013  0010    DFC0 FFEF                   JPD       STB       5,D
000014  0012    8C87                                  LDI       $B7
000015  0013    8D40 FFED                             SDI       D+1
000016                                      *           N := 0;
000017  0015    8740 FFED                             CL        D+3
000018                                      *             R OR V MODE := 'LITERAL' (R);
000019  0017    4C52                                  LDV       4,82
000020  0018    CF40 0000       P                     STR       4,RORVMD
000021                                      *             READ OR PUNCH := PUNCH @0@;
000022  001A    D800 0010       X                     LDR       5,<IOWORD+16
000023  001C    DF40 0000       P                     STR       5,READOR
000024                                      *             CHECK STATE (SHOULD BE ON);
000025  001E    FBC0 FFE6                             LAB       7,D+5
000026  0020    1C01                                  LDV       1,1
000027  0021    9F07                                  STR       1,$B7
000028  0022    D3C0 0000       P                     LNJ       5,JCS
000029  0024    FBC0 FFE0                             LAB       7,D+5
000030  0026    9BC0 0000       P                     LAB       1,BUFF1
000031  0028    9F87                                  STB       1,$B7
000032  0029    0F00 0010       X       N1            NOP       <IOWORD+16
000033  002B    E840 FFFE                             LDR       6,N1+1
000034  002D    EF40 FFD8                             STR       6,D+6
000035                                      *             IOLD (BUFFER @0@,PUNCH @0@,171);
000036  002F    C870 00AB                             LDR       4,=171
000037  0031    CF40 FFD5                             STR       4,D+7
000038  0033    D3C0 0000       P                     LNJ       5,JIL
000039                                      *             CHECK STATE (SHOULD BE ON);
000040  0035    FBC0 FFCF                             LAB       7,D+5
000041  0037    6C01                                  LDV       6,1
000042  0038    EF07                                  STR       6,$B7
000043  0039    D3C0 0000       P                     LNJ       5,JCS
000044  003B    D3C0 0000       P                     LNJ       5,JSE
000045                                      *             CHECK STATUS FOR ERRORS;
000046                                      **;
000047                                      ** EQUATE BUFFER2 WITH BUFFER1 IF SAME DATA TO BE PUNCHED ELSE Z
000048                                      *       ERO OUT;
000049                                      **;
000050                                      *             'IF' TYPE OF PUNCHES 'EQ' 'LITERAL' (R)
000051                                      *             'THEN'
000052  003D    E840 FFC4                             LDR       6,D+2
000053  003F    6D52                                  CMV       6,82
000054  0040    0990                                  BNE       >L3
000055  0041    FBC0 FFC3                             LAB       7,D+5
000056  0043    9BC0 0000       P                     LAB       1,BUFF2
000057  0045    9F87                                  STB       1,$B7
000058  0046    4C50                                  LDV       4,80
000059  0047    CF40 FFBE                             STR       4,D+6
000060                                      *                   FILL BUFFER (BUFFER 2  ,080,0)
000061  0049    CF40 FFBD                             STR       4,D+7
000062                                      *             'ELSE'
000063  004B    8740 FFBC                             CL        D+8
000064  004D    D3C0 0000       P                     LNJ       5,JFB
000065                                      *             'BEGIN'
000066  004F    0F90                                  B         >L4
000067          0050                        L3      EQU       $
000068          0050                        L5      EQU       $
000069                                      *                   'FOR' I ;=0 'STEP' 1 'UNTIL'79
000070  0050    8740 FFB3                             CL        D+4
000071                                      *                   'DO'
000072  0052    E840 FFB1                   L6        LDR       6,D+4
000073  0054    6EB1                                  ADV       6,-79
000074  0055    6A0A                                  BGZ       6,>L4
000075                                      *                         BUFFER2@I@:=BUFFER1@I@;
000076  0056    9840 FFAD                             LDR       1,D+4
000077  0058    E810 0000       X                     LDR       6,<BUFF1.$R1
000078  005A    EF10 0000       X                     STR       6,<BUFF2.$R1
000079  005C    8AC0 FFA7                             INC       D+4
000080  005E    0FF4                                  B         >L6
000081                                      *             'END';
000082          005F                        L4      EQU       $
000083                                      *             ERROR FLAG := 0;
000084  005F    8740 0000       P                     CL        ERRORF
000085  0061    0004                                  DC        4
000086                                      *             CLOCK ON;
000087  0062    6C01                        L8        LDV       6,1
000088  0063    EF40 0000       P                     STR       6,CARDSR
000089                                      *             'FOR' CARD COUNT := 1 'STEP' 1 'UNTIL' CARDS SPECIFIED-1
000090                                      *             'DO'
000091                                      *             'BEGIN'
000092  0065    E240 FF9B                             SUB       6,D+1
000093  0067    8256                                  NEG       =$R6
000094  0068    EF40 FF9C                             STR       6,D+5
000095  006A    E840 0000       P       L9            LDR       6,CARDSR
000096  006C    E240 FF98                             SUB       6,D+5
000097  006E    6A01 0069                             BGZ       6,L10
000098  0070    D3C0 0000       P                     LNJ       5,VKGB
000099                                      *                   'IF' GET BREAK 'EQ' TRUE
000100                                      *                   'THEN'  'GO TO' STOP PUNCHING;
000101                                      *
000102  0072    6D01                                  CMV       6,1
000103  0073    0901 0064                             BE        L10
000104                                      *                   'IF' TYPE OF PUNCHES 'EQ' 'LITERAL' (R)
000105                                      *                   'THEN'
000106                                      *                   'BEGIN'
000107  0075    E840 FF8C                             LDR       6,D+2
000108  0077    6D52                                  CMV       6,82
000109  0078    0981 003B                             BNE       L13
000110                                      *                         'IF' CARD COUNT 'LT' 81
000111                                      *                         'THEN'
```

```
000112                                  *                       'BEGIN'
000113  007A  C840 0000       P          LDR     4,CARDSR
000114  007C  4D51                       CMV     4,81
000115  007D  089F                       BAGE    >L14
000116                                  *                               'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000117                                  *                               'THEN'
000118                                  *                               'BEGIN'
000119  007E  4B0D                       BEVN    4,>L15
000120  007F  4EFD                       ADV     4,-3
000121                                  *                                       BUFFER2 @CARDCOUNT-3@:=0;
000122  0080  9854                       LDR     1,=$R4
000123  0081  8710 0000       X          CL      <BUFF2,$R1
000124  0083  9840 0000       P          LDR     1,CARDSR
000125  0085  1EFF                       ADV     1,-1
000126                                  *                                       BUFFER2 @CARDCOUNT-1@:=CARDCOUNT;
000127                                  *                               'END'
000128  0086  C840 0000       P          LDR     4,CARDSR
000129  0088  CF10 0000       X          STR     4,<BUFF2,$R1
000130                                  *                               'ELSE'
000131                                  *                               'BEGIN'
000132  008A  0FAA                       B       >L13
000133  008B  9840 0000       P  L15     LDR     1,CARDSR
000134  008D  9240 FF75                   SUB     1,D+3
000135                                  *                                       BUFFER1 @CARDCOUNT-N@:=0;
000136  008F  8710 0000       X          CL      <BUFF1,$R1
000137  0091  9840 0000       P          LDR     1,CARDSR
000138  0093  1EFF                       ADV     1,-1
000139                                  *                                       BUFFER1 @CARDCOUNT-1@:=CARDCOUNT;
000140  0094  E840 0000       P          LDR     6,CARDSR
000141  0096  EF10 0000       X          STR     6,<BUFF1,$R1
000142                                  *                                       N := 3;
000143  0098  6C03                       LDV     6,3
000144  0099  EF40 FF69                   STR     6,D+3
000145                                  *                               'END';
000146                                  *                       'END'
000147                                  *                       'ELSE'
000148  009B  0F99                       B       >L13
000149        009C                      L14     EQU     $
000150                                  *                               'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000151                                  *                               'THEN'
000152  009C  E840 0000       P          LDR     6,CARDSR
000153  009E  6B0C                       BEVN    6,>L17
000154                                  *                                       FILL WITH RANDOM DATA (BUFFER 2@0@,80)
000155  009F  FBC0 FF66                   LAB     7,D+6
000156  00A1  9BC0 0000       P          LAB     1,BUFF2
000157  00A3  9F87                       STB     1,$B7
000158                                  *                               'ELSE'
000159  00A4  4C50                       LDV     4,80
000160  00A5  CF40 FF61                   STR     4,D+7
000161  00A7  D3C0 0000       P          LNJ     5,VKFRD
000162  00A9  0F8B                       B       >L13
000163  00AA  FBC0 FF5B           L17     LAB     7,D+6
000164  00AC  9BC0 0000       P          LAB     1,BUFF1
000165  00AE  9F87                       STB     1,$B7
000166                                  *                                       FILL WITH RANDOM DATA (BUFFER 1@0@,80);
000167  00AF  6C50                       LDV     6,80
000168  00B0  EF40 FF56                   STR     6,D+7
000169  00B2  D3C0 0000       P          LNJ     5,VKFRD
000170                                  *                               'END';
000171        00B4                      L13     EQU     $
000172                                  *                       GENERATE CHECKSUMS (READ OR PUNCH);
000173  00B4  FBC0 FF51                   LAB     7,D+6
000174  00B6  E840 0000       P          LDR     6,READOR
000175  00B8  EF07                       STR     6,$B7
000176  00B9  D3C0 0000       P          LNJ     5,JGC
000177  00BB  D3C0 0000       P          LNJ     5,JSE
000178                                  *                       CHECK STATUS FOR ERRORS;
000179                                  *                       'IF' CARDCOUNT 'EQ' 1
000180  00BD  E840 0000       P          LDR     6,CARDSR
000181  00BF  6D01                       CMV     6,1
000182  00C0  0994                       BNE     >L19
000183                                  *                       'AND' TYPE OF PUNCHES 'EQ' 'LITERAL' (R)
000184                                  *                       'THEN'
000185  00C1  C840 FF40                   LDR     4,D+2
000186  00C3  4D52                       CMV     4,82
000187  00C4  0990                       BNE     >L19
000188  00C5  FBC0 FF40                   LAB     7,D+6
000189  00C7  9BC0 0000       P          LAB     1,BUFF1
000190  00C9  9F87                       STB     1,$B7
000191  00CA  5C54                       LDV     5,84
000192  00CB  DF40 FF3B                   STR     5,D+7
000193  00CD  7C50                       LDV     7,80
000194  00CE  FF40 FF39                   STR     7,D+8
000195                                  *                               FILL BUFFER (BUFFER 1,080,0);
000196  00D0  8740 FF38                   CL      D+9
000197  00D2  D3C0 0000       P          LNJ     5,JFB
000198                                  *               'END';
000199                                  *           STOP PUNCHING;
000200  00D4  8AC0 0000       P  L19       INC     CARDSR
000201  00D6  0F81 FF93                    B       L9
000202        00D8                      L10     EQU     $
000203  00D8  0005                        DC      5
000204                                  *             CLOCK OFF;
000205                                  *             READ OR PUNCH :=FALSE;
000206  00D9  8740 0000       P            CL      READOR
000207                                  *             GENERATE CHECKSUMS (READ OR PUNCH);
000208                                  *           'END';
000209  00DB  FBC0 FF29                    LAB     7,D+5
000210  00DD  E840 0000       P            LDR     6,READOR
000211  00DF  EF07                        STR     6,$B7
000212  00E0  D3C0 0000       P            LNJ     5,JGC
000213                                  *           'END';      (OF SEGMENT)
000214                                  *           'FINISH'
000215  00E2  83C8 FF1D                    JMP     *D
000216        00E4                      L1      EQU     $
000217  00E4  83C0 0000       P            JMP     ZK$EX
000218                                     XLOC    ZK$EX
000219                                     CTRL LINK ZK$EX
000220                                     XLOC    VKGB
000221                                     XLOC    VKFRD
000222                                     XLOC    CARDSR
000223                                     XLOC    RORVMO
000224                                     XLOC    ERRORF
```

```
000225                                  XLOC    READOR
000226                                  XLOC    BUFF1
000227                                  XLOC    BUFF2
000228                                  XLOC    IOWORD
000229                                  XLOC    JGC
000230                                  XLOC    JIL
000231                                  XLOC    JSE
000232                                  XLOC    JCS
000233                                  XLOC    JFB
000234  00E6  000B                      END     ZVKJPD,ZVKJPD
0000  ERR COUNT
@@L
```

```
E    ZVKJPN.LIST                        05/19/78  1425.2R W 05/19/78  1423.8       236250000
     000001                                         TITLE     ZVKJPN,'REV 00'
     000002         0009                            XDEF      JPN
     000003         0004                            XDEF      ZVKJPN
     000004         0008                ZKCOM       COMM      8
     000005  0000  0000                 D           RESV      4,0
     000006         0004                ZVKJPN      EQU       $
     000007  0004  8F00 0001       K                SAVE      <ZKCOM+1,Z'0011'
             0006  0011
     000008  0007  0F81 0010                        B         L1
     000009                             *       'PROCEDURE' PUT NUMBER  ('VALUE''INTEGER' NUMBER IN R6);
     000010                             *       'BEGIN'
     000011                             *           'INTEGER' A;
     000012  0009  DFC0 FFF6            JPN         STB       5,D
     000013  000B  E807                             LDR       6,$B7
     000014  000C  EF40 FFF4                        STR       6,D+1
     000015                             *             A := 'HEX' (5C);
     000016  000E  4C5C                             LDV       4,92
     000017  000F  CF40 FFF2                        STR       4,D+2
     000018                             *               'CODE''BEGIN')
     000019                             *                     XLOC ZV$TD
     000020                             *                     LNJ $B5,ZV$TD
     000021                                         XLOC      ZV$TD
     000022  0011  D3C0 0000       P                LNJ       $B5,ZV$TD
     000023                             *                     ICB]
     000024                             *                     B >+$B
     000025  0013  0F80            1                B         >+$B
     000026  0014  0001                             DC        <D+1
     000027                             *                     ICB]
     000028  0015  0002                             DC        <D+2
     000029  0016                       $B          ORG       $
     000030                             *       'END';        (OF SEGMENT)
     000031                             *       'FINISH'
     000032  0016  83C8 FFE9                        JMP       *D
     000033         0018                L1          EQU       $
     000034  0018  83C0 0000       P                JMP       ZK$EX
     000035                                         XLOC      ZK$EX
     000036                                         CTRL LINK ZK$EX
     000037  001A  0004                             END       ZVKJPN,ZVKJPN
0000  ERR COUNT
@@L
```

```
E    ZVKJRE.LIST                    05/19/78  1425.2R W 05/19/78  1423.8       268470000
     000001                                     TITLE     ZVKJRE,'REV 00'
     000002          0007                       XDEF      JRE
     000003          0002                       XDEF      ZVKJRE
     000004          0008             ZKCOM     COMM      8
     000005  0000      0000           D         RESV      2,0
     000006          0002             ZVKJRE    EQU       $
     000007  0002  8F00  0001    K               SAVE      <ZKCOM+1,Z'0011'
             0004  0011
     000008                                     *         'BEGIN'
     000009  0005  0F81  000E                    B         L1
     000010  0007  DFC0  FFF8            JRE     STB       5,D
     000011                                     *               ERROR FLAG := ERROR FLAG + 1;
     000012  0009  8AC0  0000    P               INC       ERRORF
     000013                                     *                    'CODE' 'BEGIN']
     000014                                     *                         STB $B5,+$A       ADDSESS OF CALL FOR ERROR REPORT
     000015                                     *                         LNJ $B5,<ZV$ER          CALL FOR "ERR XXXX @ AAAA"
     000016  000B  DFC0  0000    T               STB       $B5,+$A
     000017  000D  D380  0000    X               LNJ       $B5,<ZV$ER
     000018                                     *                         JCB ]
     000019                                     *                         XLOC ZV$ER
     000020                                     *                         B >+$B
     000021                                             XLOC      ZV$ER
     000022  000F  0F80          T                      B         >+$B
     000023                                     *                         JCB ]
     000024                                     *         $A              DC <$                   THIS HOLDS THE AAAA ADDRESS
     000025  0010  0010                         $A        DC        <$
     000026  0011        0000    X                      DC        <LABEL1
     000027                                     *                         JCB ]
     000028                                     *         $B              ORG $]
     000029  0012                               $B        ORG       $
     000030                                     *         'END';    (OF SEGMENT)
     000031                                     *         'FINISH'
     000032  0012  83C8  FFED                            JMP       *D
     000033  0014                              L1        EQU       $
     000034  0014  83C0  0000    P                      JMP       ZK$EX
     000035                                             XLOC      ZK$EX
     000036                                             CTRL LINK ZK$EX
     000037                                             XLOC      LABEL1
     000038                                             XLOC      ERRORF
     000039  0016  0002                                 END       ZVKJRE,ZVKJRE
0000  ERR COUNT
L
```

E ZVKJSE.LIST 05/19/78 1425.2R W 05/19/78 1422.5 3330990000

```
000001                                          TITLE   ZVKJSE,'REV 00'
000002                  00D0                    XDEF    JSE
000003                  00CB                    XDEF    ZVKJSE
000004                  0008            ZKCOM   COMM    8
000005  0000    0000                    D       RESV    7,0
000006  0007    0018                    T8      DC      24
000007  0008    2C20 5354 524F                  TEXT    ', STROBE ERROR DRO'
        000B    4245 2045 5252
                4F52 2044 524F
000008  0011    5050 4544 2024                  TEXT    'PPED $'
000009  0014    000D                    T10     DC      13
000010  0015    2C20 5354 4154                  TEXT    ', STATJS WAS$'
        0018    5553 2057 4153
                2400
000011  001C    0013                    T11     DC      19
000012  001D    2C20 5354 4154                  TEXT    ', STATJS SHOULD BE'
        0020    5553 2053 484F
                554C 4420 4245
000013  0026    2400                            TEXT    '$'
000014  0027    0012                    T13     DC      18
000015  0028    4441 5441 2053                  TEXT    'DATA SERVICE RATE$'
        002B    4552 5649 4345
                2052 4154 4524
000016  0031    0011                    T17     DC      17
000017  0032    4153 4349 4920                  TEXT    'ASCII CODE ERROR$'
        0035    434F 4445 2045
                5252 4F52 2400
000018  003B    000F                    T20     DC      15
000019  003C    5245 4144 2052                  TEXT    'READ REG ERROR$'
        003F    4547 2045 5252
                4F52 2400
000020  0044    000D                    T23     DC      13
000021  0045    5354 524F 4245                  TEXT    'STROBE ERROR$'
        0048    2045 5252 4F52
                2400
000022  004C    0011                    T25     DC      17
000023  004D    4C49 4748 542F                  TEXT    'LIGHT/DARK ERROR$'
        0050    4441 524B 2045
                5252 4F52 2400
000024  0056    0011                    T28     DC      17
000025  0057    5055 4E43 4820                  TEXT    'PUNCH ECHO ERROR$'
        005A    4543 484F 2045
                5252 4F52 2400
000026  0060    000F                    T30     DC      15
000027  0061    4F44 4420 5055                  TEXT    'ODD PUNCH ECHO$'
        0064    4E43 4820 4543
                484F 2400
000028  0069    001E                    T32     DC      30
000029  006A    4A41 4D2C 2052                  TEXT    'JAM, RECORD MAINTE'
        006D    4543 4F52 4420
                4D41 494E 5445
000030  0073    4E41 4E43 4520                  TEXT    'NANCE PANEL$'
        0076    5041 4E45 4C24
000031  0079    000D                    T34     DC      13
000032  007A    484F 5050 4552                  TEXT    'HOPPER EMPTY$'
        007D    2045 4D50 5459
                2400
000033  0081    000B                    T36     DC      11
000034  0082    4D41 524B 2053                  TEXT    'MARK SENSE$'
        0085    454E 5345 2400
000035  0088    0007                    T38     DC      7
000036  0089    3430 2043 4F4C                  TEXT    '40 COL$'
        008C    2400
000037  008D    0007                    T40     DC      7
000038  008E    3531 2043 4F4C                  TEXT    '51 COL$'
        0091    2400
000039  0092    000E                    T42     DC      14
000040  0093    4558 5420 434C                  TEXT    'EXT CLOCK TRK$'
        0096    4F43 4B20 5452
                4B24
000041  009A    000B                    T44     DC      11
000042  009B    5245 4144 2043                  TEXT    'READ CHECK$'
        009E    4845 434B 2400
000043  00A1    000B                    T46     DC      11
000044  00A2    4153 4349 4920                  TEXT    'ASCII CODE$'
        00A5    434F 4445 2400
000045  00A8    0012                    T48     DC      18
000046  00A9    434F 5252 4543                  TEXT    'CORRECTED MEM ERR$'
        00AC    5445 4420 4D45
                4D20 4552 5224
000047  00B2    0011                    T50     DC      17
000048  00B3    4E4F 4E20 4558                  TEXT    'NON EXT RESOURCE$'
        00B6    5420 5245 534F
                5552 4345 2400
000049  00BC    000B                    T52     DC      11
000050  00BD    4255 5320 5041                  TEXT    'BUS PARITY$'
        00C0    5249 5459 2400
000051  00C3    000E                    T54     DC      14
000052  00C4    554E 434F 5220                  TEXT    'UNCOR MEM ERR$'
        00C7    4D45 4D20 4552
                5224
000053                  00CB            ZVKJSE  EQU     $
000054  00CB    8F00 0001       K               SAVE    <ZKCOM+1,Z'0011'
        00CD    0011
000055                                  *       'BEGIN'
000056  00CE    0F81 019E                       B       L1
000057                                  *       'INTEGER' RANGE;
000058  00D0    DFC0 FF2F               JSE     STB     5,D
000059                                          XLOC    ZV$ER
000060  00D2    DFC0 FF2F                       STB     $B5,D+2
000061  00D4    FBC0 FF2E                       LAB     7,D+3
000062  00D6    9BC0 FF2A                       LAB     1,D+1
000063  00D8    9F87                            STB     1,$B7
000064                                  *               INOUT (RANGE, INPUT RANGE @0@);
000065  00D9    0F00 0009       X       N1      NOP     <IOWORD+9
000066  00DB    E840 FFFE                       LDR     6,N1+1
000067  00DD    EF40 FF26                       STR     6,D+4
000068  00DF    D3C0 0000       P               LNJ     5,JIO
000069                                  *               LABEL 2 := 'LITERAL' (RE);
000070                                  *       'COMMENT' TESTS M + N, CHECK RANGE, DON'T BOTHER HERE;
000071  00E1    E870 5245                       LDR     6,=21061
000072  00E3    EF40 0000       P               STR     6,LABEL2
```

```
000073                                  *           'IF' 'BITS' @8,8@ LABEL 1 'EQ' 'LITERAL' (M)
000074  00E5  C2C0 0000        P               LLH     4,LABEL1
000075  00E7  4D4D                      CMV     4,77
000076  00E8  0901 003D                  BE      L7
000077                                  *           'OR' 'BITS' @8,8@ LABEL 1 'EQ' 'LITERAL' (N)
000078  00EA  4D4E                      CMV     4,78
000079  00EB  0901 003A                  BE      L7
000080                                  *           'OR' 'BITS' @8,8@ LABEL 1 'EQ' 'LITERAL' (I)
000081  00ED  4D49                      CMV     4,73
000082  00EE  0901 0037                  BE      L7
000083                                  *           'OR' 'BITS' @8,8@ LABEL 1 'EQ' 'LITERAL' (L)
000084  00F0  4D4C                      CMV     4,76
000085  00F1  0901 0034                  BE      L7
000086                                  *           'OR' 'BITS' @8,8@ LABEL 1 'EQ' 'LITERAL' (C)
000087                                  *           'THEN'
000088                                  *               'GO TO' CHECK STATUS;
000089                                  *       'COMMENT' 400 CPM HAS DIFFERENT RANGE;
000090  00F3  4D43                      CMV     4,67
000091  00F4  0901 0031                  BE      L7
000092                                  *           'IF' 'BITS' @8,8@ LABEL 1 'NE' 'LITERAL' (F)
000093                                  *           'THEN'
000094  00F6  4D46                      CMV     4,70
000095  00F7  0906                      BE      >L5
000096                                  *           RANGE := RANGE - 11;
000097                                  *       'COMMENT' 40 COLUMN MARK SENSE HAS 80 STROBES LESS;
000098  00F8  D840 FF08                  LDR     5,D+1
000099  00FA  5EF5                      ADV     5,-11
000100  00FB  DF40 FF05                  STR     5,D+1
000101        00FD                  L5  EQU     $
000102                                  *           'IF' PUESDO CARD STATUS 'EQ' 'HEX' (1800)
000103                                  *           'THEN'
000104  00FD  E840 0000        P        LDR     6,PUESDO
000105  00FF  E970 1800                  CMR     6,=6144
000106  0101  0986                      BNE     >L6
000107                                  *           RANGE := RANGE - 80;
000108                                  *       'COMMENT' IS THERE A RANGE ERROR;
000109  0102  C840 FEFE                  LDR     4,D+1
000110  0104  4EB0                      ADV     4,-80
000111  0105  CF40 FEFB                  STR     4,D+1
000112                                  *           'IF' RANGE 'NE' 0
000113        0107                  L6  EQU     $
000114                                  *           'THEN'
000115                                  *           'BEGIN'
000116  0107  E840 FEF9                  LDR     6,D+1
000117  0109  6901 001C                  BEZ     6,L7
000118  010B  C840 FEF5                  LDR     $R4,D+1
000119  010D  D840 0000        P        LDR     $R5,SETTOA
000120  010F  DCC0 FEF2                  LDB     $B5,D+2
000121  0111  DFC0 0000        T        STB     $B5,+$A
000122  0113  D380 0000        X        LNJ     $B5,<ZV$ER
000123  0115  0F80              T        B       >+$B
000124  0116  0116              $A      DC      <$
000125  0117        0000        X       DC      <LABEL1
000126  0118                      $B      ORG     $
000127                                  *           LOAD R4 R5 AND REPORT ERROR (RANGE, CLEAR);
000128                                  *           PUT TEXT (", STROBE ERROR DROPPED $");
000129  0118  FBC0 FEEA                  LAB     7,D+3
000130  011A  9BC0 FEEC                  LAB     1,T8
000131  011C  9F87                      STB     1,$B7
000132  011D  D3C0 0000        P        LNJ     5,VKPT
000133  011F  FBC0 FEE3                  LAB     7,D+3
000134  0121  E840 FEDF                  LDR     6,D+1
000135  0123  EF07                      STR     6,$B7
000136  0124  D3C0 0000        P        LNJ     5,JPN
000137                                  *           PUT NUMBER   (RANGE);
000138                                  *           'END';
000139                                  *       CHECK STATUS;
000140        0126                  L7  EQU     $
000141                                  *           LABEL2 := 'LITERAL' (SE);
000142  0126  E870 5345                  LDR     6,=21317
000143  0128  EF40 0000        P        STR     6,LABEL2
000144  012A  C840 0000        P        LDR     4,STATUS
000145  012C  C440 0000        P        OR      4,PUESDO
000146  012E  CF40 0000        P        STR     4,STATUS
000147                                  *           STATUS SHOULD BE := STATUS SHOULD BE 'UNION' PUESDO CARD STATUS;
000148                                  *           'IF' STATUS SHOULD BE 'NE' STATUS WAS
000149                                  *           'THEN'
000150                                  *           'BEGIN'
000151  0130  C940 0000        P        CMR     4,STATWA
000152  0132  0901 0138                  BE      L9
000153                                  *               ERROR FLAG := ERROR FLAG + 1;
000154  0134  8AC0 0000        P        INC     ERRORF
000155  0136  C840 0000        P        LDR     $R4,STATWA
000156  0138  D840 0000        P        LDR     $R5,STATUS
000157  013A  DCC0 FEC7                  LDB     $B5,D+2
000158  013C  DFC0 0000        T        STB     $B5,+$A
000159  013E  D380 0000        X        LNJ     $B5,<ZV$ER
000160  0140  0F80              T        B       >+$B
000161  0141  0141              $A      DC      <$
000162  0142        0000        X       DC      <LABEL1
000163  0143                      $B      ORG     $
000164                                  *           LOAD R4 R5 AND REPORT ERROR (STATUS WAS, STATUS SHOULD BE);
000165  0143  FBC0 FEBF                  LAB     7,D+3
000166  0145  9BC0 FECE                  LAB     1,T10
000167  0147  9F87                      STB     1,$B7
000168  0148  D3C0 0000        P        LNJ     5,VKPT
000169                                  *           PUT TEXT (", STATUS WAS$");
000170  014A  FBC0 FEB8                  LAB     7,D+3
000171  014C  E840 0000        P        LDR     6,STATWA
000172  014E  EF07                      STR     6,$B7
000173  014F  D3C0 0000        P        LNJ     5,VKPH
000174                                  *           PUT HEX (STATUS WAS);
000175  0151  FBC0 FEB1                  LAB     7,D+3
000176  0153  9BC0 FEC8                  LAB     1,T11
000177  0155  9F87                      STB     1,$B7
000178  0156  D3C0 0000        P        LNJ     5,VKPT
000179                                  *           PUT TEXT(", STATUS SHOULD BE$");
000180  0158  FBC0 FEAA                  LAB     7,D+3
000181  015A  E840 0000        P        LDR     6,STATUS
000182  015C  EF07                      STR     6,$B7
000183  015D  D3C0 0000        P        LNJ     5,VKPH
000184                                  *           PUT HEX (STATUS SHOULD BE);
000185                                  *           'IF' 'BITS' @1,13@ STATUS WAS 'EQ' 1
```

```
000186                                  *                       'THEN'
000187  015F  82C0 0000        P                LB      STATWA,2'2000'
        0161  2000
000188  0162  0588                             BBF     >L12
000189                                  *                       PUT NEW TEXT ("DATA SERVICE RATE$");
000190  0163  FBC0 FE9F                         LAB     7,D+3
000191  0165  9BC0 FEC1                         LAB     1,T13
000192  0167  9F87                              STB     1,$B7
000193  0168  D3C0 0000        P                LNJ     5,VKPNT
000194        016A                      L12     EQU     $
000195                                  *                       'IF' ID 'EQ' 'HEX' (208A)
000196  016A  E840 0000        P                LDR     6,ID
000197  016C  E970 208A                         CMR     6,=8330
000198  016E  090B                              BE      >L14
000199                                  *                       'OR' ID 'EQ' 'HEX' (2088)
000200  016F  E970 2088                         CMR     6,=8328
000201  0171  0981 007A                         BNE     L15
000202                                  *                       'AND' READ OR PUNCH 'EQ' PUNCH @0@
000203                                  *                       'THEN'
000204                                  *                       'BEGIN'
000205  0173  C840 0000        P                LDR     4,READOR
000206  0175  C900 0010        X                CMR     4,<IOWORD+16
000207  0177  0981 0074                         BNE     L15
000208        0179                      L14     EQU     $
000209                                  *                       'IF' 'BITS' @1,12@ STATUS WAS 'NE'
000210                                  *                            'BITS' @1,12@ STATUS SHOULD BE
000211                                  *                       'THEN'
000212  0179  E2C0 0000        P                LLH     6,STATWA
000213  017B  6055                              SCR     6,5
000214  017C  604F                              SOR     6,15
000215  017D  C2C0 0000        P                LLH     4,STATUS
000216  017F  4055                              SCR     4,5
000217  0180  404F                              SOR     4,15
000218  0181  E954                              CMR     6,=$R4
000219  0182  0908                              BE      >L16
000220                                  *                            PUT NEW TEXT ("ASCII CODE ERROR$");
000221  0183  FBC0 FE7F                         LAB     7,D+3
000222  0185  9BC0 FEAB                         LAB     1,T17
000223  0187  9F87                              STB     1,$B7
000224  0188  D3C0 0000        P                LNJ     5,VKPNT
000225        018A                      L16     EQU     $
000226                                  *                       'IF' READ OR PUNCH 'EQ' READ @0@
000227                                  *                       'THEN'
000228                                  *                       'BEGIN'
000229  018A  E840 0000        P                LDR     6,READOR
000230  018C  E900 0007        X                CMR     6,<IOWORD+7
000231  018E  09A2                              BNE     >L18
000232                                  *                            'IF' 'BITS' @1,11@ STATUS WAS 'EQ' 1
000233                                  *                            'THEN'
000234  018F  82C0 0000        P                LB      STATWA,2'0800'
        0191  0800
000235  0192  0588                             BBF     >L19
000236  0193  FBC0 FE6F                         LAB     7,D+3
000237  0195  9BC0 FEA5                         LAB     1,T20
000238  0197  9F87                              STB     1,$B7
000239  0198  D3C0 0000        P                LNJ     5,VKPNT
000240                                  *                            PUT NEW TEXT ("READ REG ERROR$");
000241        019A                      L19     EQU     $
000242                                  *                            'IF' 'BITS' @1,10@ STATUS WAS 'EQ' 1
000243                                  *                            'THEN'
000244                                  *                            'BEGIN'
000245  019A  82C0 0000        P                LB      STATWA,2'0400'
        019C  0400
000246  019D  0593                             BBF     >L18
000247                                  *                                 'IF' RANGE 'NE' 0
000248                                  *                                 'THEN'
000249  019E  E840 FE62                         LDR     6,D+1
000250  01A0  6909                              BEZ     6,>L22
000251                                  *                                      PUT NEW TEXT ("STROBE ERROR$")
000252  01A1  FBC0 FE61                         LAB     7,D+3
000253  01A3  9BC0 FEA0                         LAB     1,T23
000254  01A5  9F87                              STB     1,$B7
000255  01A6  D3C0 0000        P                LNJ     5,VKPNT
000256                                  *                                 'ELSE'
000257  01A8  0F88                              B       >L18
000258        01A9                      L22     EQU     $
000259                                  *                                     PUT NEW TEXT ("LIGHT/DARK ERROR$");
000260  01A9  FBC0 FE59                         LAB     7,D+3
000261  01AB  9BC0 FEA0                         LAB     1,T25
000262  01AD  9F87                              STB     1,$B7
000263  01AE  D3C0 0000        P                LNJ     5,VKPNT
000264                                  *                            'END';
000265                                  *                       'END';
000266        01B0                      L18     EQU     $
000267                                  *                       'IF' READ OR PUNCH 'EQ' PUNCH @0@
000268                                  *                       'THEN'
000269                                  *                       'BEGIN'
000270  01B0  E840 0000        P                LDR     6,READOR
000271  01B2  E900 0010        X                CMR     6,<IOWORD+16
000272  01B4  0997                              BNE     >L26
000273                                  *                            'IF' 'BITS' @1,11@ STATUS WAS 'EQ' 1
000274                                  *                            'THEN'
000275  01B5  82C0 0000        P                LB      STATWA,2'0800'
        01B7  0800
000276  01B8  0588                             BBF     >L27
000277                                  *                            PUT NEW TEXT ("PUNCH ECHO ERROR$");
000278  01B9  FBC0 FE49                         LAB     7,D+3
000279  01BB  9BC0 FE9A                         LAB     1,T28
000280  01BD  9F87                              STB     1,$B7
000281  01BE  D3C0 0000        P                LNJ     5,VKPNT
000282        01C0                      L27     EQU     $
000283                                  *                            'IF' 'BITS' @1,10@ STATUS WAS 'EQ' 1
000284                                  *                            'THEN'
000285  01C0  82C0 0000        P                LB      STATWA,2'0400'
        01C2  0400
000286  01C3  0588                             BBF     >L26
000287  01C4  FBC0 FE3E                         LAB     7,D+3
000288  01C6  9BC0 FE99                         LAB     1,T30
000289  01C8  9F87                              STB     1,$B7
000290  01C9  D3C0 0000        P                LNJ     5,VKPNT
000291                                  *                            PUT NEW TEXT ("ODD PUNCH ECHO$");
000292                                  *                       'END';
000293        01CB                      L26     EQU     $
```

```
000294                                  *                       'IF' 'BITS' @1,9@ STATUS WAS 'EQ' 1
000295                                  *                       'THEN'
000296                                  *                       'BEGIN'
000297                                  *                           PUT NEW TEXT
000298  01CB  82C0 0000       P         *           LB          STATWA,Z'0200'
        01CD  0200
000299  01CE  058A                      *           BBF         >L31
000300                                  *                           ("JAM, RECORD MAINTENANCE PANEL$");
000301  01CF  FBC0 FE33                 *           LAB         7,D+3
000302  01D1  9BC0 FE97                 *           LAB         1,T32
000303  01D3  9F87                      *           STB         1,$B7
000304  01D4  D3C0 0000       P         *           LNJ         5,VKPNT
000305  01D6  D3C0 0000       P         *           LNJ         5,JWR
000306                                  *                           WAIT FOR RETURN;
000307                                  *                       'END';
000308        01D8                      *           L31         EQU         $
000309                                  *                       'IF' 'BITS' @1,8@ STATUS WAS 'NE'
000310                                  *                            'BITS' @1,8@ STATUS SHOULD BE
000311                                  *                       'THEN'
000312  01D8  E2C0 0000       P         *           LLH         6,STATWA
000313  01DA  E570 0001                 *           AND         6,=Z'0001'
000314  01DC  C2C0 0000       P         *           LLH         4,STATUS
000315  01DE  C570 0001                 *           AND         4,=Z'0001'
000316  01E0  E954                      *           CMR         6,=$R4
000317  01E1  0901 005E                 *           BE          L33
000318  01E3  FBC0 FE1F                 *           LAB         7,D+3
000319  01E5  9BC0 FE93                 *           LAB         1,T34
000320  01E7  9F87                      *           STB         1,$B7
000321  01E8  D3C0 0000       P         *           LNJ         5,VKPNT
000322                                  *                           PUT NEW TEXT ("HOPPER EMPTY$");
000323                                  *                       'END'
000324                                  *                       'ELSE'
000325                                  *                       'BEGIN'
000326  01EA  0F81 0055                 *           B           L33
000327        01EC                      *           L15         EQU         $
000328                                  *                       'IF' 'BITS' @1,12@ STATUS WAS 'NE'
000329                                  *                            'BITS' @1,12@ STATUS SHOULD BE
000330                                  *                       'THEN'
000331  01EC  E2C0 0000       P         *           LLH         6,STATWA
000332  01EE  6055                      *           SCR         6,5
000333  01EF  604F                      *           SOR         6,15
000334  01F0  C2C0 0000       P         *           LLH         4,STATUS
000335  01F2  4055                      *           SCR         4,5
000336  01F3  404F                      *           SOR         4,15
000337  01F4  E954                      *           CMR         6,=$R4
000338  01F5  0908                      *           BE          >L35
000339                                  *                           PUT NEW TEXT ("MARK SENSE$");
000340  01F6  FBC0 FE0C                 *           LAB         7,D+3
000341  01F8  9BC0 FE88                 *           LAB         1,T36
000342  01FA  9F87                      *           STB         1,$B7
000343  01FB  D3C0 0000       P         *           LNJ         5,VKPNT
000344        01FD                      *           L35         EQU         $
000345                                  *                       'IF' 'BITS' @1,11@ STATUS WAS 'NE'
000346                                  *                            'BITS' @1,11@ STATUS SHOULD BE
000347                                  *                       'THEN'
000348  01FD  E2C0 0000       P         *           LLH         6,STATWA
000349  01FF  6054                      *           SCR         6,4
000350  0200  604F                      *           SOR         6,15
000351  0201  C2C0 0000       P         *           LLH         4,STATUS
000352  0203  4054                      *           SCR         4,4
000353  0204  404F                      *           SOR         4,15
000354  0205  E954                      *           CMR         6,=$R4
000355  0206  0908                      *           BE          >L37
000356                                  *                           PUT NEW TEXT ("40 COL$");
000357  0207  FBC0 FDFB                 *           LAB         7,D+3
000358  0209  9BC0 FE7E                 *           LAB         1,T38
000359  020B  9F87                      *           STB         1,$B7
000360  020C  D3C0 0000       P         *           LNJ         5,VKPNT
000361        020E                      *           L37         EQU         $
000362                                  *                       'IF' 'BITS' @1,10@ STATUS WAS 'EQ' 1
000363                                  *                       'THEN'
000364  020E  82C0 0000       P         *           LB          STATWA,Z'0400'
        0210  0400
000365  0211  0588                      *           BBF         >L39
000366                                  *                           PUT NEW TEXT ("51 COL$");
000367  0212  FBC0 FDF0                 *           LAB         7,D+3
000368  0214  9BC0 FE78                 *           LAB         1,T40
000369  0216  9F87                      *           STB         1,$B7
000370  0217  D3C0 0000       P         *           LNJ         5,VKPNT
000371        0219                      *           L39         EQU         $
000372                                  *                       'IF' 'BITS' @1,9@ STATUS WAS 'EQ' 1
000373                                  *                       'THEN'
000374  0219  82C0 0000       P         *           LB          STATWA,Z'0200'
        021B  0200
000375  021C  0588                      *           BBF         >L41
000376  021D  FBC0 FDE5                 *           LAB         7,D+3
000377  021F  9BC0 FE72                 *           LAB         1,T42
000378  0221  9F87                      *           STB         1,$B7
000379  0222  D3C0 0000       P         *           LNJ         5,VKPNT
000380                                  *                           PUT NEW TEXT ("EXT CLOCK TRK$");
000381        0224                      *           L41         EQU         $
000382                                  *                       'IF' 'BITS' @1,8@ STATUS WAS 'EQ' 1
000383                                  *                       'THEN'
000384  0224  82C0 0000       P         *           LB          STATWA,Z'0100'
        0226  0100
000385  0227  0588                      *           BBF         >L43
000386                                  *                           PUT NEW TEXT ("READ CHECK$");
000387  0228  FBC0 FDDA                 *           LAB         7,D+3
000388  022A  9BC0 FE6F                 *           LAB         1,T44
000389  022C  9F87                      *           STB         1,$B7
000390  022D  D3C0 0000       P         *           LNJ         5,VKPNT
000391        022F                      *           L43         EQU         $
000392                                  *                       'IF' 'BITS' @1,7@ STATUS WAS 'NE'
000393                                  *                            'BITS' @1,7@ STATUS SHOULD BE
000394                                  *                       'THEN'
000395  022F  E840 0000       P         *           LDR         6,STATWA
000396  0231  E2D6                      *           LLH         6,=$R6
000397  0232  6047                      *           SOR         6,7
000398  0233  C840 0000       P         *           LDR         4,STATUS
000399  0235  C2D4                      *           LLH         4,=$R4
000400  0236  4047                      *           SOR         4,7
000401  0237  E954                      *           CMR         6,=$R4
000402  0238  0908                      *           BE          >L33
```

```
000403                                  *                       PUT NEW TEXT ("ASCII CODE$");
000404  0239  FBC0 FDC9                          LAB     7,D+3
000405  023B  9BC0 FE65                          LAB     1,T46
000406  023D  9F87                               STB     1,$B7
000407  023E  D3C0 0000      P                   LNJ     5,VKPNT
000408                                  *
000409        0240                      L33     EQU     $
000410                                  *                       'IF' 'BITS' @1,3@ STATUS WAS 'EQ' 1
000411                                  *                       'THEN'
000412  0240  82C0 0000      P                   LB      STATWA,Z'0008'
      0242  0008
000413  0243  0588                               BBF     >L47
000414                                  *                       PUT NEW TEXT ("CORRECTED MEM ERR$");
000415  0244  FBC0 FDBE                          LAB     7,D+3
000416  0246  9BC0 FE61                          LAB     1,T48
000417  0248  9F87                               STB     1,$B7
000418  0249  D3C0 0000      P                   LNJ     5,VKPNT
000419        024B                      L47     EQU     $
000420                                  *                       'IF' 'BITS' @1,2@ STATUS WAS 'EQ'1
000421                                  *                       'THEN'
000422  024B  82C0 0000      P                   LB      STATWA,Z'0004'
      024D  0004
000423  024E  0588                               BBF     >L49
000424                                  *                       PUT NEW TEXT ("NON EXT RESOURCE$");
000425  024F  FBC0 FDB3                          LAB     7,D+3
000426  0251  9BC0 FE60                          LAB     1,T50
000427  0253  9F87                               STB     1,$B7
000428  0254  D3C0 0000      P                   LNJ     5,VKPNT
000429        0256                      L49     EQU     $
000430                                  *                       'IF' 'BITS' @1,1@ STATUS WAS 'EQ'1
000431                                  *                       'THEN'
000432  0256  82C0 0000      P                   LB      STATWA,Z'0002'
      0258  0002
000433  0259  0588                               BBF     >L51
000434                                  *                       PUT NEW TEXT ("BUS PARITY$");
000435  025A  FBC0 FDA8                          LAB     7,D+3
000436  025C  9BC0 FE5F                          LAB     1,T52
000437  025E  9F87                               STB     1,$B7
000438  025F  D3C0 0000      P                   LNJ     5,VKPNT
000439        0261                      L51     EQU     $
000440                                  *                       'IF' 'BITS' @1,0@ STATUS WAS 'EQ' 1
000441                                  *                       'THEN'
000442  0261  E840 0000      P                   LDR     6,STATWA
000443  0263  6B08                               BEVN    6,>L9
000444  0264  FBC0 FD9E                          LAB     7,D+3
000445  0266  9BC0 FE5C                          LAB     1,T54
000446  0268  9F87                               STB     1,$B7
000447  0269  D3C0 0000      P                   LNJ     5,VKPNT
000448                                  *                       PUT NEW TEXT ("UNCOR MEM ERR$");
000449                                  *           'END';
000450                                  *       'END';
000451                                  *       'END';          (OF SEGMENT)
000452                                  *       'FINISH'
000453  026B  83C8 FD94                  L9      JMP     *D
000454        026D                      L1      EQU     $
000455  026D  83C0 0000      P                   JMP     ZK$EX
000456                                          XLOC    ZK$EX
000457                                          CTRL LINK ZK$EX
000458                                          XLOC    VKPT
000459                                          XLOC    VKPNT
000460                                          XLOC    VKPH
000461                                          XLOC    ID
000462                                          XLOC    STATWA
000463                                          XLOC    STATUS
000464                                          XLOC    PUESDO
000465                                          XLOC    SETTOA
000466                                          XLOC    LABEL1
000467                                          XLOC    LABEL2
000468                                          XLOC    ERRORF
000469                                          XLOC    READOR
000470                                          XLOC    IOWORD
000471                                          XLOC    JIO
000472                                          XLOC    JPN
000473                                          XLOC    JWR
000474  026F  00CB                              END     ZVKJSE,ZVKJSE
0000  ERR COUNT
L
```

```
E   ZVKJTC.LIST                    05/19/78  1425.3R W 05/19/78  1423.9      1034730000
000001                                        TITLE     ZVKJTC,'REV 00'
000002          000C                          XDEF      JGT
000003          0007                          XDEF      ZVKJTC
000004          0008                ZKCOM     COMM      8
000005  0000    0000                D         RESV      7,0
000006          0007                ZVKJTC    EQU       $
000007  0007 8F00 0001      K               SAVE      <ZKCOM+1,Z'0011'
        0009 0011
000008                              *       'BEGIN'
000009  000A 0F81 0096                      B         L1
000010                              *         'INTEGER' N;
000011  000C DFC0 FFF3                JGT       STB       5,D
000012                              *           N := 0;
000013  000E 8740 FFF2                          CL        D+1
000014                              *           READ OR PUNCH ;=FALSE;
000015  0010 8740 0000      P                   CL        READOR
000016                              *           RESET RANDOM DATA (CLEAR);
000017  0012 FBC0 FFEF                          LAB       7,D+2
000018  0014 9BC0 0000      P                   LAB       1,SETTOA
000019  0016 9F87                               STB       1,$B7
000020  0017 D3C0 0000      P                   LNJ       5,VKRRD
000021  0019 FBC0 FFE8                          LAB       7,D+2
000022  001B 9BC0 0000      P                   LAB       1,BUFF1
000023  001D 9F87                               STB       1,$B7
000024  001E 6C54                               LDV       6,84
000025  001F EF40 FFE3                          STR       6,D+3
000026  0021 4C50                               LDV       4,80
000027  0022 CF40 FFE1                          STR       4,D+4
000028                              *           FILL BUFFER (BUFFER 1,080,'HEX'(FFF));
000029  0024 D870 0FFF                          LDR       5,=4095
000030  0026 DF40 FFDE                          STR       5,D+5
000031  0028 D3C0 0000      P                   LNJ       5,JFB
000032  002A FBC0 FFD7                          LAB       7,D+2
000033  002C 9BC0 0000      P                   LAB       1,BUFF2
000034  002E 9F87                               STB       1,$B7
000035  002F 6C50                               LDV       6,80
000036  0030 EF40 FFD2                          STR       6,D+3
000037  0032 EF40 FFD1                          STR       6,D+4
000038                              *           FILL BUFFER (BUFFER 2,080,'HEX'(000));
000039  0034 8740 FFD0                          CL        D+5
000040  0036 D3C0 0000      P                   LNJ       5,JFB
000041  0038 6C01                   L3        LDV       6,1
000042  0039 EF40 0000      P                   STR       6,CARDSR
000043                              *           'FOR' CARD COUNT ;= 1 'STEP' 1 'UNTIL' 99
000044                              *           'DO'
000045                              *           'BEGIN'
000046  003B E840 0000      P       L4          LDR       6,CARDSR
000047  003D 6E9D                               ADV       6,-99
000048  003E 6A01 0059                          BGZ       6,L5
000049                              *                       'IF' CARD COUNT 'LT' 81
000050                              *                       'THEN'
000051                              *                       'BEGIN'
000052  0040 E840 0000      P                   LDR       6,CARDSR
000053  0042 6D51                               CMV       6,81
000054  0043 089F                               BAGE      >L6
000055                              *                                 'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000056                              *                                 'THEN'
000057                              *                                 'BEGIN'
000058  0044 6B0D                               BEVN      6,>L7
000059  0045 6EFD                               ADV       6,-3
000060                              *                                       BUFFER2 @CARDCOUNT-3@:=0;
000061  0046 9856                               LDR       1,=$R6
000062  0047 8710 0000      X                   CL        <BUFF2,$R1
000063  0049 9840 0000      P                   LDR       1,CARDSR
000064  004B 1EFF                               ADV       1,-1
000065                              *                                       BUFFER2 @CARDCOUNT-1@:=CARDCOUNT;
000066                              *                                 'END'
000067  004C E840 0000      P                   LDR       6,CARDSR
000068  004E EF10 0000      X                   STR       6,<BUFF2,$R1
000069                              *                                 'ELSE'
000070                              *                                 'BEGIN'
000071  0050 0FAA                               B         >L8
000072  0051 9840 0000      P       L7          LDR       1,CARDSR
000073  0053 9240 FFAD                          SUB       1,D+1
000074                              *                                       BUFFER1 @CARDCOUNT-N@:=0;
000075  0055 8710 0000      X                   CL        <BUFF1,$R1
000076  0057 9840 0000      P                   LDR       1,CARDSR
000077  0059 1EFF                               ADV       1,-1
000078                              *                                       BUFFER1 @CARDCOUNT-1@:=CARDCOUNT;
000079  005A E840 0000      P                   LDR       6,CARDSR
000080  005C EF10 0000      X                   STR       6,<BUFF1,$R1
000081                              *                                       N := 3;
000082  005E 6C03                               LDV       6,3
000083  005F EF40 FFA1                          STR       6,D+1
000084                              *                                 'END';
000085                              *                       'END'
000086                              *                       'ELSE'
000087  0061 0F99                               B         >L8
000088          0062                L6        EQU       $
000089                              *                                 'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000090                              *                                 'THEN'
000091  0062 E840 0000      P                   LDR       6,CARDSR
000092  0064 6B0C                               BEVN      6,>L9
000093                              *                                       FILL WITH RANDOM DATA (BUFFER 2@0@,80)
000094  0065 FBC0 FF9C                          LAB       7,D+2
000095  0067 9BC0 0000      P                   LAB       1,BUFF2
000096  0069 9F87                               STB       1,$B7
000097                              *                                 'ELSE'
000098  006A 4C50                               LDV       4,80
000099  006B CF40 FF97                          STR       4,D+3
000100  006D D3C0 0000      P                   LNJ       5,VKFRD
000101  006F 0F8B                               B         >L8
000102  0070 FBC0 FF91              L9          LAB       7,D+2
000103  0072 9BC0 0000      P                   LAB       1,BUFF1
000104  0074 9F87                               STB       1,$B7
000105                              *                                       FILL WITH RANDOM DATA (BUFFER 1@0@,80);
000106  0075 6C50                               LDV       6,80
000107  0076 EF40 FF8C                          STR       6,D+3
000108  0078 D3C0 0000      P                   LNJ       5,VKFRD
000109          007A                L8        EQU       $
000110                              *                       GENERATE CHECKSUMS (READ OR PUNCH);
000111  007A FBC0 FF87                          LAB       7,D+2
```

```
000112  007C  E840 0000      P              LDR       6,READOR
000113  007E  EF07                          STR       6,$B7
000114  007F  D3C0 0000      P              LNJ       5,JGC
000115                              *                    'IF' CARDCOUNT 'EQ' 1
000116                              *                    'THEN'
000117  0081  E840 0000      P              LDR       6,CARDSR
000118  0083  6D01                          CMV       6,1
000119  0084  0990                          BNE       >L11
000120  0085  FBC0 FF7C                     LAB       7,D+2
000121  0087  9BC0 0000      P              LAB       1,BUFF1
000122  0089  9F87                          STB       1,$B7
000123  008A  4C54                          LDV       4,84
000124  008B  CF40 FF77                     STR       4,D+3
000125  008D  5C50                          LDV       5,80
000126  008E  DF40 FF75                     STR       5,D+4
000127                              *                              FILL BUFFER (BUFFER 1,080,0);
000128  0090  8740 FF74                     CL        D+5
000129  0092  D3C0 0000      P              LNJ       5,JFB
000130                              *            'END';
000131  0094  8AC0 0000      P      L11     INC       CARDSR
000132  0096  0F81 FFA4                     B         L4
000133        0098                  L5      EQU       $
000134                              *          GENERATE CHECKSUMS (READ OR PUNCH);
000135                              *      'END';
000136  0098  FBC0 FF69                       LAB       7,D+2
000137  009A  E840 0000      P                LDR       6,READOR
000138  009C  EF07                            STR       6,$B7
000139  009D  D3C0 0000      P                LNJ       5,JGC
000140                              *      'END';      (OF SEGMENT)
000141                              *      'FINISH'
000142  009F  83C8 FF60                       JMP       *D
000143        00A1                  L1      EQU       $
000144  00A1  83C0 0000      P                JMP       ZK$EX
000145                                       XLOC      ZK$EX
000146                                       CTRL LINK ZK$EX
000147                                       XLOC      VKRRD
000148                                       XLOC      VKFRD
000149                                       XLOC      SETTOA
000150                                       XLOC      CARDSR
000151                                       XLOC      READOR
000152                                       XLOC      BUFF1
000153                                       XLOC      BUFF2
000154                                       XLOC      JGC
000155                                       XLOC      JFB
000156  00A3  0007                           END       ZVKJTC,ZVKJTC
0000  ERR COUNT
ee@L
```

```
E    ZVKJTD.LIST                    05/19/78  1425.3R W 05/19/78  1423.9       5419800000
     000001                                     TITLE     ZVKJID,'REV 00'
     000002          0004                       XDEF      (TIME1,D+4)
     000003          0000                       XDEF      (BASE,D)
     000004          0001                       XDEF      (SUM,D+1)
     000005          0002                       XDEF      (TOTAL,D+2)
     000006          0003                       XDEF      (RESPON,D+3)
     000007          000F                       XDEF      JTD
     000008          000A                       XDEF      ZVKJTD
     000009          0008               ZKCOM   COMM      8
     000010  0000    0000               D       RESV      10,0
     000011          000A               ZVKJTD  EQU       $
     000012  000A    8F00 0001    K             SAVE      <ZKCOM+1,Z'0011'
             000C    0011
     000013                                 *       'INTEGER' BASE;  (BASE OF TIMING DELAY)
     000014                                 *       'INTEGER' SUM;   (SUM OF ERROR TIMMING)
     000015                                 *       'INTEGER' TOTAL; (TOTAL OF RANDOM DELAYS)
     000016                                 *       'INTEGER' RESPONSE; (RESPONSE OF DELAY)
     000017                                 *       'INTEGER' TIME1,TIME2; (OLD DELAY TIMES)
     000018                                 **;
     000019                                 ** RANDOM OR FIXED TIMING DELAY;
     000020                                 **;
     000021                                 *       'PROCEDURE' TIME DELAY;
     000022                                 *       'BEGIN'
     000023  000D    0F81 002F                        B         L1
     000024  000F    DFC0 FFF6          JTD           STB       5,D+6
     000025                                 *             TIME1 := TIME2; (GET DELAY FOR CARD BEING READ)
     000026  0011    E840 FFF3                        LDR       6,D+5
     000027  0013    EF40 FFF0                        STR       6,D+4
     000028                                 *             TIME2 := ZHRTCI; (GET DELAY FOR PREVIOUS READ)
     000029  0015    C840 0000    P                   LDR       4,ZHRTCI
     000030  0017    CF40 FFED                        STR       4,D+5
     000031                                 *          'IF' BASE 'EQ' 0
     000032                                 *          'THEN'
     000033                                 *          'BEGIN'
     000034  0019    D840 FFE6                        LDR       5,D
     000035  001B    5989                             BNEZ      5,>L3
     000036                                 *             ZHRTCC := RESPONSE + 100;
     000037  001C    F840 FFE6                        LDR       7,D+3
     000038  001E    7E64                             ADV       7,100
     000039  001F    FF40 0000    P                   STR       7,ZHRTCC
     000040                                 *             ZHRTCI := ZHRTCC;
     000041                                 *          'END'
     000042  0021    FF40 0000    P                   STR       7,ZHRTCI
     000043                                 *          'ELSE'
     000044                                 *          'BEGIN'
     000045  0023    0F92                             B         >L4
     000046  0024    E840 FFDB          L3            LDR       6,D
     000047  0026    E570 3ECD                        AND       6,=16077
     000048                                 *             BASE := (BASE 'MASK' 'HEX' (3ECD) + 'HEX' (5555)) / 2;
     000049  0028    EA70 5555                        ADD       6,=21845
     000050  002A    6061                             SAR       6,1
     000051  002B    EF40 FFD4                        STR       6,D
     000052                                 *             ZHRTCC := BASE 'MASK' 'HEX' (07C0) / 64 + 103;(NEXT DELAY)
     000053  002D    E570 07C0                        AND       6,=1984
     000054  002F    6066                             SAR       6,6
     000055  0030    6E67                             ADV       6,103
     000056  0031    EF40 0000    P                   STR       6,ZHRTCC
     000057                                 *             ZHRTCI := ZHRTCC;
     000058  0033    EF40 0000    P                   STR       6,ZHRTCI
     000059                                 *          'END';
     000060          0035               L4            EQU       $
     000061  0035    0004                             DC        4
     000062                                 *             CLOCK ON;
     000063                                 *       KEEP WAITING:
     000064          0036                   L5        EQU       $
     000065                                 *          'IF' ZHRTCC 'GT' 100
     000066                                 *          'THEN'
     000067                                 *             'GO TO' KEEP WAITING;
     000068  0036    E840 0000    P                   LDR       6,ZHRTCC
     000069  0038    6D64                             CMV       6,100
     000070  0039    0A7D                             BAG       >L5
     000071  003A    0005                             DC        5
     000072                                 *             CLOCK OFF;
     000073                                 *       'END';   (OF PROCEDURE)
     000074                                 *       'END';   (OF SEGMENT)
     000075                                 *       'FINISH'
     000076  003B    83C8 FFCA                        JMP       *D+6
     000077          003D               L1            EQU       $
     000078  003D    83C0 0000    P                   JMP       ZK$EX
     000079                                           XLOC      ZK$EX
     000080                                           CTRL LINK ZK$EX
     000081                                           XLOC      ZHRTCI
     000082                                           XLOC      ZHRTCC
     000083  003F    000A                             END       ZVKJTD,ZVKJTD
0000  ERR COUNT
eL
```

```
E   ZVKJWR.LIST                                  05/19/78  1425.3R W 05/19/78  1424.0       4645800000
    000001                                                  TITLE     ZVKJWR,'REV 00'
    000002          0025                                    XDEF      JWR
    000003          0021                                    XDEF      ZVKJWR
    000004          0008                          ZKCOM     COMM      8
    000005  0000    0000                          D         RESV      4,0
    000006  0004    0023                          T5        DC        35
    000007  0005    5459 5045 2027                          TEXT      'TYPE ''CARRIAGE RET'
            0008    4341 5252 4941
                    4745 2052 4554
    000008  000E    5552 4E27 2C20                          TEXT      'URN'', WHEN READY$'
            0011    5748 454E 2052
                    4541 4459 2400
    000009  0017    0012                          T8        DC        18
    000010  0018    2A2A 2A2A 2A20                          TEXT      '***** BREAK *****$'
            001B    4252 4541 4B20
                    2A2A 2A2A 2A24
    000011          0021                          ZVKJWR    EQU       $
    000012  0021    8F00 0001       K                       SAVE      <ZKCOM+1,Z'0011'
            0023    0011
    000013                                        *         'BEGIN'
    000014  0024    0FA8                                    B         >L1
    000015                                        *         'INTEGER' TEMP CHARACTER;
    000016  0025    DFC0 FFDA                     JWR       STB       5,D
    000017                                        *             'IF' ZVDTTY 'EQ' 0
    000018                                        *                 'THEN'
    000019  0027    E840 0000       P                       LDR       6,ZV$TTY
    000020  0029    6985                                    BNEZ      6,>L3
    000021                                        *             TEMP CHARACTER := 'HEX' (0D00);
    000022                                        *         WAIT;
    000023  002A    C870 0D00                               LDR       4,=3328
    000024  002C    CF40 FFD4                               STR       4,D+1
    000025  002E    FBC0 FFD3                     L3        LAB       7,D+2
    000026  0030    9BC0 FFD3                               LAB       1,T5
    000027  0032    9F87                                    STB       1,$B7
    000028  0033    D3C0 0000       P                       LNJ       5,VKPNT
    000029                                        *             PUT NEW TEXT ("TYPE 'CARRIAGE RETURN', WHEN READY$");
    000030  0035    FBC0 FFCC                               LAB       7,D+2
    000031  0037    9BC0 FFC9                               LAB       1,D+1
    000032  0039    9F87                                    STB       1,$B7
    000033  003A    D3C0 0000       P                       LNJ       5,VKGC
    000034                                        *             GET CHAR (TEMP CHARACTER);
    000035                                        *             'IF' TEMP CHARACTER 'NE' 'HEX' (0D)
    000036                                        *             'THEN'
    000037                                        *             'BEGIN'
    000038  003C    E840 FFC4                               LDR       6,D+1
    000039  003E    6D0D                                    CMV       6,13
    000040  003F    090B                                    BE        >L6
    000041                                        *                     'IF'
    000042                                        *                         TEMP CHARACTER 'EQ' 0
    000043                                        *                     'THEN'
    000044                                        *                     'BEGIN'
    000045  0040    69EE                                    BNEZ      6,>L3
    000046                                        *                         PUT NEW TEXT ("***** BREAK *****$");
    000047  0041    FBC0 FFC0                               LAB       7,D+2
    000048  0043    9BC0 FFD3                               LAB       1,T8
    000049  0045    9F87                                    STB       1,$B7
    000050  0046    D3C0 0000       P                       LNJ       5,VKPNT
    000051                                        *                     'GO TO' ASK NEXT;
    000052  0048    83C0 0000       P                       JMP       ASKNEX
    000053                                        *                     'END';
    000054                                        *                 'GO TO' WAIT;
    000055                                        *             'END';
    000056                                        *         'END';
    000057                                        *         'END';
    000058                                        *         'FINISH'
    000059  004A    83C8 FFB5                     L6        JMP       *D
    000060          004C                          L1        EQU       $
    000061  004C    83C0 0000       P                       JMP       ZK$EX
    000062                                                  XLOC      ZK$EX
    000063                                                  CTRL LINK ZK$EX
    000064                                                  XLOC      VKPNT
    000065                                                  XLOC      VKGC
    000066                                                  XLOC      ZV$TTY
    000067                                                  CTRL LINK ZV$TT
    000068                                                  XLOC      ASKNEX
    000069  004E    0021                                    END       ZVKJWR,ZVKJWR
0000  ERR COUNT
@L
```

```
E   ZVKJLB.LIST                              05/19/78  1427.1R W 05/19/78  1422.3      24819300O0
000001                                                  TITLE     ZVKJLB,'REV 00'
000002          0097                                    XDEF      KJLB
000003          0092                                    XDEF      ZVKJLB
000004          0008                            ZKCOM   COMM      8
000005  0000    0000                            D       RESV      6,0
000006  0006    000C                            T3      DC        12
000007  0007    454C 4543 5452                          TEXT      'ELECTRONICS$'
        000A    4F4E 4943 5324
000008  000D    0012                            T4      DC        18
000009  000E    4649 524D 5741                          TEXT      'FIRMWARE REVISION$'
        0011    5245 2052 4556
                4953 494F 4E24
000010  0017    001E                            T7      DC        30
000011  0018    2C20 4153 4349                          TEXT      ', ASCII NOT SET ON'
        001B    4920 4E4F 5420
                5345 5420 4F4E
000012  0021    2049 4E49 5449                          TEXT      ' INITIALIZE$'
        0024    414C 495A 4524
000013  0027    0022                            T9      DC        34
000014  0028    2C20 4249 4E41                          TEXT      ', BINARY NOT SET O'
        002B    5259 204E 4F54
                2053 4554 204F
000015  0031    4E20 434F 4E46                          TEXT      'N CONFIGURATION$'
        0034    4947 5552 4154
                494F 4E24
000016  0039    0013                            T11     DC        19
000017  003A    2C20 5441 534B                          TEXT      ', TASK NOT CLEARED'
        003D    204E 4F54 2043
                4C45 4152 4544
000018  0043    2400                                    TEXT      '$'
000019  0044    0013                            T13     DC        19
000020  0045    2C20 5441 534B                          TEXT      ', TASK NOT CLEARED'
        0048    204E 4F54 2043
                4C45 4152 4544
000021  004E    2400                                    TEXT      '$'
000022  004F    0019                            T17     DC        25
000023  0050    4341 4E4E 4F54                          TEXT      'CANNOT SET PROGRAM'
        0053    2053 4554 2050
                524F 4752 414D
000024  0059    2045 4A45 4354                          TEXT      ' EJECT$'
        005C    2400
000025  005D    0015                            T19     DC        21
000026  005E    4341 4E4E 4F54                          TEXT      'CANNOT SET HOLLERI'
        0061    2053 4554 2048
                4F4C 4C45 5249
000027  0067    5448 2400                               TEXT      'TH$'
000028  0069    001B                            T21     DC        27
000029  006A    4341 4E4E 4F54                          TEXT      'CANNOT SET PUNCH F'
        006D    2053 4554 2050
                554E 4348 2046
000030  0073    4545 4420 5245                          TEXT      'EED READ$'
        0076    4144 2400
000031  0078    0015                            T23     DC        21
000032  0079    4341 4E4E 4F54                          TEXT      'CANNOT SET TEST MO'
        007C    2053 4554 2054
                4553 5420 4D4F
000033  0082    4445 2400                               TEXT      'DE$'
000034  0084    001A                            T25     DC        26
000035  0085    4341 4E4E 4F54                          TEXT      'CANNOT SET EJECT O'
        0088    2053 4554 2045
                4A45 4354 204F
000036  008E    4E20 4552 524F                          TEXT      'N ERROR$'
        0091    5224
000037          0092                            ZVKJLB  EQU       $
000038  0092    8F00 0001       K                       SAVE      <ZKCOM+1,Z'0011'
        0094    0011
000039                                          *       'BEGIN'
000040  0095    0F81 017F                               B         L1
000041                                          *           'INTEGER'TEMP;
000042  0097    DFC0 FF68                       KJLB    STB       5,D
000043                                          *           ERROR LABEL := 'LITERAL' (B1);
000044  0099    E870 4231                               LDR       6,=16945
000045  009B    EF40 0000       P                       STR       6,LABEL1
000046                                          *           PUT NEW TEXT ("ELECTRONICS$");
000047  009D    FBC0 FF64                               LAB       7,D+2
000048  009F    93C0 FF66                               LAB       1,T3
000049  00A1    9F87                                    STB       1,$B7
000050  00A2    D3C0 0000       P                       LNJ       5,VKPNT
000051                                          *           CHECK STATE (SHOULD BE OFF);
000052  00A4    FBC0 FF5D                               LAB       7,D+2
000053  00A6    8707                                    CL        $B7
000054  00A7    D3C0 0000       P                       LNJ       5,JCS
000055                                          *           TEMP := 'HEX' (FFFF);
000056  00A9    6CFF                                    LDV       6,-1
000057  00AA    EF40 FF56                               STR       6,D+1
000058  00AC    FBC0 FF55                               LAB       7,D+2
000059  00AE    9BC0 FF52                               LAB       1,D+1
000060  00B0    9F87                                    STB       1,$B7
000061                                          *           INOUT (TEMP, OUTPUT CONFIGURATION @0@); (OUTPUT GARBAGE)
000062  00B1    0F00 000B       X       N1      NOP       <IOWORD+11
000063  00B3    C840 FFFE                               LDR       4,N1+1
000064  00B5    CF40 FF4D                               STR       4,D+3
000065  00B7    D3C0 0000       P                       LNJ       5,JIO
000066  00B9    FBC0 FF48                               LAB       7,D+2
000067  00BB    9BC0 FF45                               LAB       1,D+1
000068  00BD    9F87                                    STB       1,$B7
000069                                          *           INOUT (TEMP, OUTPUT TASK          @0@); (OUTPUT GARBAGE)
000070  00BE    0F00 0005       X       N2      NOP       <IOWORD+5
000071  00C0    E840 FFFE                               LDR       6,N2+1
000072  00C2    EF40 FF40                               STR       6,D+3
000073  00C4    D3C0 0000       P                       LNJ       5,JIO
000074  00C6    FBC0 FF3B                               LAB       7,D+2
000075  00C8    9BC0 0000       P                       LAB       1,BINARY
000076  00CA    9F87                                    STB       1,$B7
000077                                          *           INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000078  00CB    ABC0 0000       P                       LAB       2,IOWORD
000079  00CD    AFC0 FF35                               STB       2,D+3
000080  00CF    D3C0 0000       P                       LNJ       5,JIO
000081                                          *           PUT NEW TEXT ("FIRMWARE REVISION$");
000082  00D1    FBC0 FF30                               LAB       7,D+2
000083  00D3    9BC0 FF39                               LAB       1,T4
000084  00D5    9F87                                    STB       1,$B7
```

```
000085  00D6  D3C0 0000       P                LNJ       5,VKPNT
000086  00D8  FBC0 FF29                        LAB       7,D+2
000087  00DA  9BC0 FF26                        LAB       1,D+1
000088  00DC  9F87                             STB       1,$B7
000089                              *              INOUT (TEMP,INPUT FIRMWARE REV @0@);
000090  00DD  0F00 0003       X     N3         NOP       <IOWORD+3
000091  00DF  E840 FFFE                        LDR       6,N3+1
000092  00E1  EF40 FF21                        STR       6,D+3
000093  00E3  D3C0 0000       P                LNJ       5,JIO
000094                              *              TEMP := 'BITS' @8,8@ TEMP;   (RIGHT JUSTIFY)
000095  00E5  E2C0 FF1B                        LLH       6,D+1
000096  00E7  EF40 FF19                        STR       6,D+1
000097                              *              PUT HEX     (TEMP                );
000098  00E9  FBC0 FF18                        LAB       7,D+2
000099  00EB  EF07                             STR       6,$B7
000100  00EC  D3C0 0000       P                LNJ       5,VKPH
000101                              *              'IF' ID 'EQ' 'HEX' (2008)
000102                              *              'THEN'
000103                              *              'BEGIN'
000104  00EE  E840 0000       P                LDR       6,ID
000105  00F0  E970 2008                        CMR       6,=8200
000106  00F2  0981 0056                        BNE       L5
000107  00F4  FBC0 FF0D                        LAB       7,D+2
000108  00F6  9BC0 FF0A                        LAB       1,D+1
000109  00F8  9F87                             STB       1,$B7
000110                              *                    INOUT (TEMP,INPUT CONFIGURATION @0@);
000111  00F9  0F00 000A       X     N4         NOP       <IOWORD+10
000112  00FB  C840 FFFE                        LDR       4,N4+1
000113  00FD  CF40 FF05                        STR       4,D+3
000114  00FF  D3C0 0000       P                LNJ       5,JIO
000115                              *                    'IF' TEMP 'NE' 0
000116                              *                    'THEN'
000117                              *                    'BEGIN'
000118  0101  E840 FEFF                        LDR       6,D+1
000119  0103  6901 0012                        BEZ       6,L6
000120                              *                         LABEL 2 := 'LITERAL' (AM);
000121  0105  C870 414D                        LDR       4,=16717
000122  0107  CF40 0000       P                STR       4,LABEL2
000123  0109  D840 0000       P                LDR       $R5,SETTOA
000124  010B  C840 FEF5                        LDR       $R4,D+1
000125                              *                         LOAD R4 R5 (TEMP, CLEAR);
000126  010D  D3C0 0000       P                LNJ       5,JRE
000127                              *                         REPORT ERROR;
000128                              *                         PUT TEXT   (", ASCII NOT SET ON INITIALIZE$");     (MDC)
000129  010F  FBC0 FEF2                        LAB       7,D+2
000130  0111  9BC0 FF05                        LAB       1,T7
000131  0113  9F87                             STB       1,$B7
000132  0114  D3C0 0000       P                LNJ       5,VKPT
000133                              *                    'END';
000134                              **;
000135                              **           CONFIGURATION TEST  - CARDREADER;
000136                              **;
000137  0116  FBC0 FEEB                        L6        LAB       7,D+2
000138  0118  9BC0 0000       P                LAB       1,BINARY
000139  011A  9F87                             STB       1,$B7
000140                              *                    INOUT (BINARY MODE, OUTPUT CONFIGURATION @0@);
000141  011B  0F00 000B       X     N5         NOP       <IOWORD+11
000142  011D  E840 FFFE                        LDR       6,N5+1
000143  011F  EF40 FEE3                        STR       6,D+3
000144  0121  D3C0 0000       P                LNJ       5,JIO
000145  0123  FBC0 FEDE                        LAB       7,D+2
000146  0125  9BC0 FEDB                        LAB       1,D+1
000147  0127  9F87                             STB       1,$B7
000148                              *                    INOUT (TEMP,INPUT CONFIGURATION @0@);
000149  0128  0F00 000A       X     N6         NOP       <IOWORD+10
000150  012A  E840 FFFE                        LDR       6,N6+1
000151  012C  EF40 FED6                        STR       6,D+3
000152  012E  D3C0 0000       P                LNJ       5,JIO
000153                              *                    'IF' TEMP 'NE' 'HEX' (8000)
000154                              *                    'THEN'
000155                              *                    'BEGIN'
000156  0130  E840 FED0                        LDR       6,D+1
000157  0132  E970 8000                        CMR       6,=-32768
000158  0134  0901 00D1                        BE        L8
000159                              *                         LABEL 2 := 'LITERAL' (BM);
000160  0136  C870 424D                        LDR       4,=16973
000161  0138  CF40 0000       P                STR       4,LABEL2
000162  013A  D840 0000       P                LDR       $R5,SETTOA
000163  013C  C840 FEC4                        LDR       $R4,D+1
000164                              *                         LOAD R4 R5 (TEMP, CLEAR);
000165  013E  D3C0 0000       P                LNJ       5,JRE
000166                              *                         REPORT ERROR;
000167                              *                         PUT TEXT   (", BINARY NOT SET ON CONFIGURATION$");(MDC)
000168  0140  FBC0 FEC1                        LAB       7,D+2
000169  0142  9BC0 FEE4                        LAB       1,T9
000170  0144  9F87                             STB       1,$B7
000171  0145  D3C0 0000       P                LNJ       5,VKPT
000172                              *                    'END';
000173                              *                 'END'
000174                              *                 'ELSE'
000175                              *                 'BEGIN'
000176                              **;
000177                              **           TASK WORD TESTS - PUNCH;
000178                              **;
000179  0147  0F81 00BE                        B         L8
000180        0149                  L5        EQU       $
000181                              *                    ERROR LABEL := 'LITERAL' (B2);
000182  0149  E870 4232                        LDR       6,=16946
000183  014B  EF40 0000       P                STR       6,LABEL1
000184  014D  FBC0 FEB4                        LAB       7,D+2
000185  014F  9BC0 FEB1                        LAB       1,D+1
000186  0151  9F87                             STB       1,$B7
000187                              *                    INOUT (TEMP , INPUT TASK @0@);(GARBAGE BEEN CLEARED)
000188  0152  0F00 0004       X     N7         NOP       <IOWORD+4
000189  0154  C840 FFFE                        LDR       4,N7+1
000190  0156  CF40 FEAC                        STR       4,D+3
000191  0158  D3C0 0000       P                LNJ       5,JIO
000192                              *                    'IF' TEMP 'NE' 0
000193                              *                    'THEN'
000194                              *                    'BEGIN'
000195  015A  E840 FEA6                        LDR       6,D+1
000196  015C  6901 0012                        BEZ       6,L10
000197                              *                         LABEL 2 := 'LITERAL' (TC);
```

```
000198  015E  C870 5443          LDR     4,=21571
000199  0160  CF40 0000  P       STR     4,LABEL2
000200  0162  D840 0000  P       LDR     $R5,SETTOA
000201  0164  C840 FE9C          LDR     $R4,D+1
000202                      *                 LOAD R4 R5 (TEMP, CLEAR);
000203  0166  D3C0 0000  P       LNJ     5,JRE
000204                      *                 REPORT ERROR;
000205  0168  FBC0 FE99          LAB     7,D+2
000206  016A  9BC0 FECE          LAB     1,T11
000207  016C  9F87               STB     1,$B7
000208  016D  D3C0 0000  P       LNJ     5,VKPT
000209                      *                 PUT TEXT (", TASK NOT CLEARED$");
000210                      *               'END';
000211  016F  FBC0 FE92          L10     LAB     7,D+2
000212  0171  9BC0 0000  P       LAB     1,SETTOA
000213  0173  9F87               STB     1,$B7
000214                      *             INOUT (CLEAR, OUTPUT TASK @0@);
000215  0174  0F00 0005  X       N8      NOP     <IOWORD+5
000216  0176  E840 FFFE          LDR     6,N8+1
000217  0178  EF40 FE8A          STR     6,D+3
000218  017A  D3C0 0000  P       LNJ     5,JIO
000219  017C  FBC0 FE85          LAB     7,D+2
000220  017E  9BC0 FE82          LAB     1,D+1
000221  0180  9F87               STB     1,$B7
000222                      *             INOUT  (TEMP,INPUT TASK @0@);
000223  0181  0F00 0004  X       N9      NOP     <IOWORD+4
000224  0183  E840 FFFE          LDR     6,N9+1
000225  0185  EF40 FE7D          STR     6,D+3
000226  0187  D3C0 0000  P       LNJ     5,JIO
000227                      *             'IF' TEMP 'NE' 0
000228                      *             'THEN'
000229                      *             'BEGIN'
000230  0189  E840 FE77          LDR     6,D+1
000231  018B  6901 0012          BEZ     6,L12
000232                      *                 LABEL 2 := 'LITERAL' (TS);
000233  018D  C870 5453          LDR     4,=21587
000234  018F  CF40 0000  P       STR     4,LABEL2
000235  0191  D840 0000  P       LDR     $R5,SETTOA
000236  0193  C840 FE6D          LDR     $R4,D+1
000237                      *                 LOAD R4 R5 (TEMP, CLEAR);
000238  0195  D3C0 0000  P       LNJ     5,JRE
000239                      *                 REPORT ERROR;
000240  0197  FBC0 FE6A          LAB     7,D+2
000241  0199  9BC0 FEAA          LAB     1,T13
000242  019B  9F87               STB     1,$B7
000243  019C  D3C0 0000  P       LNJ     5,VKPT
000244                      *                 PUT TEXT (", TASK NOT CLEARED$");
000245                      *               'END';
000246        019E                L12     EQU     $
000247                      *             TEMP := 'HEX' (5E00);
000248  019E  E870 5E00          LDR     6,=24064
000249  01A0  EF40 FE60          STR     6,D+1
000250  01A2  FBC0 FE5F          LAB     7,D+2
000251  01A4  9BC0 FE5C          LAB     1,D+1
000252  01A6  9F87               STB     1,$B7
000253                      *             INOUT (TEMP,OUTPUT TASK @0@);   (SET TASK WORD)
000254  01A7  0F00 0005  X       N10     NOP     <IOWORD+5
000255  01A9  C840 FFFE          LDR     4,N10+1
000256  01AB  CF40 FE57          STR     4,D+3
000257  01AD  D3C0 0000  P       LNJ     5,JIO
000258  01AF  FBC0 FE52          LAB     7,D+2
000259  01B1  9BC0 FE4F          LAB     1,D+1
000260  01B3  9F87               STB     1,$B7
000261                      *             INOUT (TEMP,INPUT TASK @0@);
000262  01B4  0F00 0004  X       N11     NOP     <IOWORD+4
000263  01B6  E840 FFFE          LDR     6,N11+1
000264  01B8  EF40 FE4A          STR     6,D+3
000265  01BA  D3C0 0000  P       LNJ     5,JIO
000266  01BC  82C0 FE44          LB      D+1,Z'1E00'
        01BE  1E00
000267  01BF  0586               BBF     >L14
000268                      *             'IF' 'BITS' @4,9@ TEMP 'EQ' 0 'OR' 'BITS' @1,14@ TEMP 'EQ' 0
000269                      *             'THEN'
000270                      *           'BEGIN'
000271  01C0  82C0 FE40          LB      D+1,Z'4000'
        01C2  4000
000272  01C3  0501 000B          BBT     L15
000273        01C5                L14     EQU     $
000274                      *                 LABEL 2 := 'LITERAL' (TW);
000275  01C5  E870 5457          LDR     6,=21591
000276  01C7  EF40 0000  P       STR     6,LABEL2
000277  01C9  D840 0000  P       LDR     $R5,SETTOA
000278  01CB  C840 FE35          LDR     $R4,D+1
000279                      *                 LOAD R4 R5 (TEMP, CLEAR);
000280  01CD  D3C0 0000  P       LNJ     5,JRE
000281                      *                 REPORT ERROR;
000282                      *           'END';
000283        01CF                L15     EQU     $
000284                      *             'IF' 'BITS' @1,9@ TEMP 'NE' 1
000285                      *             'THEN'
000286  01CF  82C0 FE31          LB      D+1,Z'0200'
        01D1  0200
000287  01D2  0508               BBT     >L16
000288                      *                 PUT NEW TEXT ("CANNOT SET PROGRAM EJECT$");
000289  01D3  FBC0 FE2E          LAB     7,D+2
000290  01D5  9BC0 FE79          LAB     1,T17
000291  01D7  9F87               STB     1,$B7
000292  01D8  D3C0 0000  P       LNJ     5,VKPNT
000293        01DA                L16     EQU     $
000294                      *             'IF' 'BITS' @1,10@ TEMP 'NE' 1
000295                      *             'THEN'
000296  01DA  82C0 FE26          LB      D+1,Z'0400'
        01DC  0400
000297  01DD  0508               BBT     >L18
000298                      *                 PUT NEW TEXT ("CANNOT SET HOLLERITH$");
000299  01DE  FBC0 FE23          LAB     7,D+2
000300  01E0  9BC0 FE7C          LAB     1,T19
000301  01E2  9F87               STB     1,$B7
000302  01E3  D3C0 0000  P       LNJ     5,VKPNT
000303        01E5                L18     EQU     $
000304                      *             'IF' 'BITS' @1,11@ TEMP 'NE' 1
000305                      *             'THEN'
000306  01E5  82C0 FE1B          LB      D+1,Z'0800'
```

```
          01E7  0800
000307    01E8  0508                        *          BBT       >L20
000308                                                  PUT NEW TEXT("CANNOT SET PUNCH FEED READ$");
000309    01E9  FBC0 FE18                   *          LAB       7,D+2
000310    01EB  9BC0 FE7D                   *          LAB       1,T21
000311    01ED  9F87                        *          STB       1,$B7
000312    01EE  D3C0 0000       P           *          LNJ       5,VKPNT
000313          01F0                        L20        EQU       $
000314                                    *          'IF' 'BITS' @1,12@ TEMP 'NE' 1
000315                                    *          'THEN'
000316    01F0  82C0 FE10                   *          LB        D+1,Z'1000'
          01F2  1000
000317    01F3  0508                        *          BBT       >L22
000318                                                  PUT NEW TEXT ("CANNOT SET TEST MODE$");
000319    01F4  FBC0 FE0D                   *          LAB       7,D+2
000320    01F6  9BC0 FE81                   *          LAB       1,T23
000321    01F8  9F87                        *          STB       1,$B7
000322    01F9  D3C0 0000       P           *          LNJ       5,VKPNT
000323          01FB                        L22        EQU       $
000324                                    *          'IF' 'BITS' @1,14@ TEMP 'NE' 1
000325                                    *          'THEN'
000326    01FB  82C0 FE05                   *          LB        D+1,Z'4000'
          01FD  4000
000327    01FE  0508                        *          BBT       >L8
000328                                                  PUT NEW TEXT ("CANNOT SET EJECT ON ERROR$");
000329    01FF  FBC0 FE02                   *          LAB       7,D+2
000330    0201  9BC0 FE82                   *          LAB       1,T25
000331    0203  9F87                        *          STB       1,$B7
000332    0204  D3C0 0000       P           *          LNJ       5,VKPNT
000333                                    *               'END';
000334    0206  FBC0 FDFB                   L8         LAB       7,D+2
000335    0208  9BC0 0000       P           *          LAB       1,SETTOA
000336    020A  9F87                        *          STB       1,$B7
000337                                    *               INOUT (CLEAR, OUTPUT TASK @0@);
000338                                    *          'END';
000339    020B  0F00 0005       X           N12        NOP       <IOWORD+5
000340    020D  E840 FFFE                   *          LDR       6,N12+1
000341    020F  EF40 FDF3                   *          STR       6,D+3
000342    0211  D3C0 0000       P           *          LNJ       5,JIO
000343                                    *          'END';
000344                                    *          'FINISH'
000345    0213  83C8 FDEC                   *          JMP       *D
000346          0215                        L1         EQU       $
000347    0215  83C0 0000       P           *          JMP       ZK$EX
000348                                                  XLOC      ZK$EX
000349                                                  CTRL LINK ZK$EX
000350                                                  XLOC      VKPT
000351                                                  XLOC      VKPNT
000352                                                  XLOC      VKPH
000353                                                  XLOC      ID
000354                                                  XLOC      SETTOA
000355                                                  XLOC      BINARY
000356                                                  XLOC      LABEL1
000357                                                  XLOC      LABEL2
000358                                                  XLOC      IOWORD
000359                                                  XLOC      JRE
000360                                                  XLOC      JIO
000361                                                  XLOC      JCS
000362    0217  0092                        *          END       ZVKJLB,ZVKJLB
0000  ERR COUNT
@@L
```

```
E   ZVKJLC.LIST                                  05/19/78  1427.1R W 05/19/78  1422.4      184/700000
000001                                                      TITLE     ZVKJLC,'REV 00'
000002          00C2                                        XDEF      KJLC
000003          006C                                        XDEF      ZVKJLC
000004          0008                                ZKCOM   COMM      8
000005  0000    0000                                D       RESV      12,0
000006  000C    000C                                T10     DC        12
000007  000D    5055 5420 4F4E                              TEXT      'PUT ON-LINE$'
        0010    2D4C 494E 4524
000008  0013    0029                                T12     DC        41
000009  0014    4445 5649 4345                              TEXT      'DEVICE CONTROLS',Z'0D0A','L'
        0017    2043 4F4E 5452
        001A    4F4C 530D 0A4C
000010  001D    4F41 4420 494E                              TEXT      'OAD INPUT, PUT ON-LINE$'
        0020    5055 542C 2050
                5554 204F 4E2D
                4C49 4E45 2400
000011  0029    000C                                T13     DC        12
000012  002A    5354 4F50 2044                              TEXT      'STOP DEVICE$'
        002D    4556 4943 4524
000013  0030    0015                                T16     DC        21
000014  0031    534C 4944 4520                              TEXT      'SLIDE OUTPUT STACK'
        0034    4F55 5450 5554
                2053 5441 434B
000015  003A    4552 2400                                   TEXT      'ER$'
000016  003C    0010                                T17     DC        16
000017  003D    5245 4C45 4153                              TEXT      'RELEASE STACKER$'
        0040    4520 5354 4143
                4B45 5224
000018  0045    0010                                T19     DC        16
000019  0046    5445 5354 2049                              TEXT      'TEST INTERLOCK5$'
        0049    4E54 4552 4C4F
                434B 5324
000020  004E    0009                                T22     DC        9
000021  004F    4F50 454E 204C                              TEXT      'OPEN LID$'
        0052    4944 2400
000022  0054    000A                                T23     DC        10
000023  0055    434C 4F53 4520                              TEXT      'CLOSE LID$'
        0058    4C49 4424
000024  005A    000F                                T24     DC        15
000025  005B    4F50 454E 2052                              TEXT      'OPEN REAR DOOR$'
        005E    4541 5220 444F
                4F52 2400
000026  0063    0010                                T25     DC        16
000027  0064    434C 4F53 4520                              TEXT      'CLOSE REAR DOOR$'
        0067    5245 4152 2044
                4F4F 5224
000028          006C                                ZVKJLC  EQU       $
000029  006C    8F00 0001           K                       SAVE      <ZKCOM+1,Z'0011'
        006E    0011
000030  006F    0F81 00E5                                   B         L1
000031                                      *       'PROCEDURE' WAIT 10 SECONDS ('VALUE''INTEGER' STATE SHOULD BE);
000032                                      *       'BEGIN'
000033  0071    DFC0 FF8E                           P2      STB       5,D
000034  0073    EF40 FF8D                                   STR       6,D+1
000035                                      *                 ZHRTCC := HERTZ * 10; (SET CLOCK FOR 10 SECONDS)
000036  0075    C840 0000           P                       LDR       4,HERTZ
000037  0077    4F0A                                        MLV       4,10
000038  0078    CF40 0000           P                       STR       4,ZHRTCC
000039                                      *                 ZHRTCI := HERTZ * 10;
000040  007A    D840 0000           P                       LDR       5,HERTZ
000041  007C    5F0A                                        MLV       5,10
000042  007D    DF40 0000           P                       STR       5,ZHRTCI
000043                                      *                 ZHRTCL := 20;          (SET LEVEL)
000044  007F    7C14                                        LDV       7,20
000045  0080    FF40 0000           P                       STR       7,ZHRTCL
000046                                      *                 CLOCK SEMAPHORE := 0;
000047  0082    8740 0000           P                       CL        CKSEM
000048  0084    0004                                        DC        4
000049                                      *                 CLOCK ON;
000050                                      *       WAIT;
000051  0085    FBC0 FF7C                           L3      LAB       7,D+2
000052  0087    9BC0 0000           P                       LAB       1,STATWA
000053  0089    9F87                                        STB       1,$B7
000054                                      *                 INOUT (STATUS WAS,INPUT STATUS WORD @0@);
000055  008A    0F00 000C           X       N1              NOP       <IOWORD+12
000056  008C    E840 FFFE                                   LDR       6,N1+1
000057  008E    EF40 FF74                                   STR       6,D+3
000058  0090    D3C0 0000           P                       LNJ       5,JIO
000059  0092    D3C0 0000           P                       LNJ       5,VKGB
000060                                      *                 'IF' GET BREAK 'EQ' TRUE
000061                                      *                 'THEN'
000062                                      *                       'GO TO' ASK NEXT;
000063  0094    6D01                                        CMV       6,1
000064  0095    0901 0000           P                       BE        ASKNEX
000065                                      *                 'IF' 'BITS' @1,15@ STATUS WAS 'NE' STATE SHOULD BE
000066                                      *                 'THEN'
000067                                      *                 'BEGIN'
000068  0097    E2C0 0000           P                       LLH       6,STATWA
000069  0099    6047                                        SOK       6,7
000070  009A    E940 FF66                                   CMR       6,D+1
000071  009C    0911                                        BE        >L6
000072                                      *                 'IF' CLOCK SEMAPHORE 'NE' 0
000073                                      *                 'THEN'
000074                                      *                 'BEGIN'
000075  009D    C840 0000           P                       LDR       4,CKSEM
000076  009F    4966                                        BEZ       4,>L3
000077                                      *                       STATUS SHOULD BE := STATE SHOULD BE*'HEX'(8000);
000078  00A0    D840 FF60                                   LDR       5,D+1
000079  00A2    DB70 8000                                   MUL       5,=-32768
000080  00A4    DF40 0000           P                       STR       5,STATUS
000081                                      *                       STATUS SHOULD BE :=STATUS SHOULD BE'UNION''HEX'(4000);
000082  00A6    8940 0000           P                       LBT       STATUS,Z'4000'
        00A8    4000
000083  00A9    D3C0 0000           P                       LNJ       5,JSE
000084                                      *                       CHECK STATUS FOR ERRORS;
000085                                      *                       'GO TO' EXIT;
000086                                      *                 'END'
000087  00AB    0F81 0001                                   B         L6
000088                                      *                 'ELSE'
000089                                      *                       'GO TO' WAIT;
000090                                      *             'END';
000091                                      *       EXIT;
```

```
000092          00AD                    L6      EQU     $
000093  00AD     0005                            DC      5
000094                                  *               CLOCK OFF;
000095                                  *       'END';  (OF 10 SEC PROCEDURE)
000096                                  **;
000097                                  ** PROCEDURE TO START DEVICE;
000098                                  **;
000099                                  *       'PROCEDURE' START DEVICE;
000100  00AE    83C8 FF51                       JMP     *D
000101                                  *       'BEGIN'
000102  00B0    DFC0 FF54               P9      STB     5,D+5
000103                                  *               LABEL 1 := 'LITERAL' (C1);
000104  00B2    E870 4331                       LDR     6,=17201
000105  00B4    EF40 0000       P               STR     6,LABEL1
000106                                  *               PUT NEW TEXT ("PUT ON-LINE$");
000107  00B6    FBC0 FF4F                       LAB     7,D+6
000108  00B8    9BC0 FF53                       LAB     1,T10
000109  00BA    9F87                            STB     1,$B7
000110  00BB    D3C0 0000       P               LNJ     5,VKPNT
000111                                  *               WAIT 10 SECONDS (SHOULD BE ON);
000112  00BD    6C01                            LDV     6,1
000113  00BE    D3C0 FFB2                       LNJ     5,P2
000114                                  *       'END';
000115                                  **;
000116                                  ** LETTER C - DEVICE SAFETY FEATURES;
000117                                  **;
000118                                  *       'PROCEDURE' LETTER CP;
000119  00C0    83C8 FF44                       JMP     *D+5
000120                                  *       'BEGIN'
000121                                  *               'INTEGER' RESPONSE;
000122  00C2    DFC0 FF44               KJLC    STB     5,D+7
000123                                  *               LABEL 1 := 'LITERAL' (C1);
000124  00C4    E870 4331                       LDR     6,=17201
000125  00C6    EF40 0000       P               STR     6,LABEL1
000126  00C8    FBC0 FF40                       LAB     7,D+9
000127  00CA    9BC0 0000       P               LAB     1,BINARY
000128  00CC    9F87                            STB     1,$B7
000129                                  *               INOUT (INITIALIZE, OUTPUT CONTROL@0@);
000130  00CD    ABC0 0000       P               LAB     2,IOWORD
000131  00CF    AFC0 FF3A                       STB     2,D+10
000132  00D1    D3C0 0000       P               LNJ     5,JIO
000133                                  *               PUT NEW TEXT ("DEVICE CONTROLS!C L!LOAD INPUT, PUT ON-LINE$");
000134  00D3    FBC0 FF35                       LAB     7,D+9
000135  00D5    9BC0 FF3D                       LAB     1,T12
000136  00D7    9F87                            STB     1,$B7
000137  00D8    D3C0 0000       P               LNJ     5,VKPNT
000138  00DA    D3C0 0000       P               LNJ     5,JWR
000139                                  *               WAIT FOR RETURN;
000140                                  *               WAIT 10 SECONDS (SHOULD BE ON);
000141  00DC    6C01                            LDV     6,1
000142  00DD    D3C0 FF93                       LNJ     5,P2
000143                                  *               LABEL 1 := 'LITERAL' (C2);
000144  00DF    E870 4332                       LDR     6,=17202
000145  00E1    EF40 0000       P               STR     6,LABEL1
000146                                  *               PUT NEW TEXT ("STOP DEVICE$");
000147  00E3    FBC0 FF25                       LAB     7,D+9
000148  00E5    9BC0 FF43                       LAB     1,T13
000149  00E7    9F87                            STB     1,$B7
000150  00E8    D3C0 0000       P               LNJ     5,VKPNT
000151                                  *               WAIT 10 SECONDS (SHOULD BE OFF);
000152  00EA    6C00                            LDV     6,0
000153  00EB    D3C0 FF85                       LNJ     5,P2
000154                                  *               'IF' ID 'EQ' 'HEX' (2008)
000155                                  *               'THEN'
000156                                  *                       'GO TO' END LETTER C;
000157  00ED    E840 0000       P               LDR     6,ID
000158  00EF    E970 2008                       CMR     6,=8200
000159  00F1    0901 0061                       BE      L15
000160  00F3    D3C0 FFBC                       LNJ     5,P9
000161                                  *               START DEVICE;
000162                                  *               LABEL 1 := 'LITERAL' (C3);
000163  00F5    E870 4333                       LDR     6,=17203
000164  00F7    EF40 0000       P               STR     6,LABEL1
000165                                  *               PUT NEW TEXT ("SLIDE OUTPUT STACKER$");
000166  00F9    FBC0 FF0F                       LAB     7,D+9
000167  00FB    9BC0 FF34                       LAB     1,T16
000168  00FD    9F87                            STB     1,$B7
000169  00FE    D3C0 0000       P               LNJ     5,VKPNT
000170                                  *               WAIT 10 SECONDS (SHOULD BE OFF);
000171  0100    6C00                            LDV     6,0
000172  0101    D3C0 FF6F                       LNJ     5,P2
000173                                  *               PUT NEW TEXT ("RELEASE STACKER$");
000174  0103    FBC0 FF05                       LAB     7,D+9
000175  0105    9BC0 FF36                       LAB     1,T17
000176  0107    9F87                            STB     1,$B7
000177  0108    D3C0 0000       P               LNJ     5,VKPNT
000178  010A    D3C0 FFA5                       LNJ     5,P9
000179                                  *               START DEVICE;
000180                                  *       ASK AGAIN;
000181          010C                    L18     EQU     $
000182                                  *               ASK NEW QUESTION ("TEST INTERLOCKS$");
000183  010C    FBC0 FEFC                       LAB     7,D+9
000184  010E    9BC0 FF36                       LAB     1,T19
000185  0110    9F87                            STB     1,$B7
000186  0111    D3C0 0000       P               LNJ     5,VKANQ
000187                                  *               RESPONSE := 'HEX' (4E0D);
000188  0113    E870 4E0D                       LDR     6,=19981
000189  0115    EF40 FEF2                       STR     6,D+8
000190  0117    FBC0 FEF1                       LAB     7,D+9
000191  0119    9BC0 FEEE                       LAB     1,D+8
000192  011B    9F87                            STB     1,$B7
000193  011C    D3C0 0000       P               LNJ     5,VKGC
000194                                  *               GET CHAR (RESPONSE);
000195                                  *               'IF' RESPONSE 'NE' 'LITERAL' (Y)
000196                                  *               'THEN'
000197                                  *                'BEGIN'
000198  011E    E840 FEE9                       LDR     6,D+8
000199  0120    6D59                            CMV     6,89
000200  0121    0904                            BE      >L20
000201                                  *                       'IF' RESPONSE 'NE' 'LITERAL' (N)
000202                                  *                       'THEN'
000203                                  *                               'GO TO' ASK AGAIN;
000204  0122    6D4E                            CMV     6,78
```

```
000205  0123  09E9                        BNE       >L18
000206                              *               'GO TO' END LETTER C;
000207  0124  0FAF                        B         >L15
000208                              *     'END';
000209        0125                  L20   EQU       $
000210                              *     LABEL 1 := 'LITERAL' (C4);
000211  0125  E870 4334                    LDR       6,=17204
000212  0127  EF40 0000        P           STR       6,LABEL1
000213                              *     PUT NEW TEXT ("OPEN LID$");
000214  0129  FBC0 FEDF                    LAB       7,D+9
000215  012B  9BC0 FF22                    LAB       1,T22
000216  012D  9F87                         STB       1,$B7
000217  012E  D3C0 0000        P           LNJ       5,VKPNT
000218                              *     WAIT 10 SECONDS (SHOULD BE OFF);
000219  0130  6C00                        LDV       6,0
000220  0131  D3C0 FF3F                    LNJ       5,P2
000221                              *     PUT NEW TEXT ("CLOSE LID$");
000222  0133  FBC0 FED5                    LAB       7,D+9
000223  0135  9BC0 FF1E                    LAB       1,T23
000224  0137  9F87                         STB       1,$B7
000225  0138  D3C0 0000        P           LNJ       5,VKPNT
000226  013A  D3C0 FF75                    LNJ       5,P9
000227                              *     START DEVICE;
000228                              *     LABEL 1 := 'LITERAL' (C5);
000229  013C  E870 4335                    LDR       6,=17205
000230  013E  EF40 0000        P           STR       6,LABEL1
000231  0140  FBC0 FEC8                    LAB       7,D+9
000232  0142  9BC0 FF17                    LAB       1,T24
000233  0144  9F87                         STB       1,$B7
000234  0145  D3C0 0000        P           LNJ       5,VKPNT
000235                              *     PUT NEW TEXT ("OPEN REAR DOOR$");
000236                              *     WAIT 10 SECONDS (SHOULD BE OFF);
000237  0147  6C00                        LDV       6,0
000238  0148  D3C0 FF28                    LNJ       5,P2
000239                              *     PUT NEW TEXT ("CLOSE REAR DOOR$");
000240  014A  FBC0 FEBE                    LAB       7,D+9
000241  014C  9BC0 FF16                    LAB       1,T25
000242  014E  9F87                         STB       1,$B7
000243  014F  D3C0 0000        P           LNJ       5,VKPNT
000244  0151  D3C0 FF5E                    LNJ       5,P9
000245                              *     START DEVICE;
000246                              *   END LETTER C;
000247                              *   'END';      (OF LETTER C)
000248                              *   'END';      (OF SEGMENT)
000249                              *   'FINISH'
000250  0153  83C8 FEB3            L15     JMP       *D+7
000251        0155                  L1      EQU       $
000252  0155  83C0 0000        P           JMP       ZK$EX
000253                                    XLOC      ZK$EX
000254                                    CTRL LINK ZK$EX
000255                                    XLOC      ZHRTC1
000256                                    XLOC      ZHRTCL
000257                                    XLOC      ZHRTCC
000258                                    XLOC      VKPNT
000259                                    XLOC      VKANQ
000260                                    XLOC      VKGC
000261                                    XLOC      VKGB
000262                                    XLOC      ID
000263                                    XLOC      STATWA
000264                                    XLOC      STATUS
000265                                    XLOC      HERTZ
000266                                    XLOC      BINARY
000267                                    XLOC      LABEL1
000268                                    XLOC      CKSEM
000269                                    XLOC      IOWORD
000270                                    XLOC      JIO
000271                                    XLOC      JWR
000272                                    XLOC      JSE
000273                                    XLOC      ASKNEX
000274  0157  006C                        END       ZVKJLC,ZVKJLC
0000  ERR COUNT
eL
```

```
E   ZVKJLF.LIST                                    05/19/78  1427.2R W 05/19/78  1422.4      2928150000
000001                                                        TITLE     ZVKJLF,'REV 00'
000002          005C                                          XDEF      KJLF
000003          0057                                          XDEF      ZVKJLF
000004          0008                                ZKCOM     COMM      8
000005  0000    0000                                D         RESV      11,0
000006  000B    0015                                T8        DC        21
000007  000C    5055 4E43 4820                                TEXT      'PUNCH 400 CPM - MA'
        000F    3430 3020 4350
                4D20 2D20 4D41
000008  0015    5820 2400                                     TEXT      'X $'
000009  0017    000E                                T9        DC        14
000010  0018    2043 4152 4453                                TEXT      ' CARDS, TOTAL$'
        001B    2C20 544F 5441
                4C24
000011  001F    000C                                T11       DC        12
000012  0020    5245 4144 5920                                TEXT      'READY PUNCH$'
        0023    5055 4E43 4824
000013  0026    0021                                T30       DC        33
000014  0027    5241 5445 2053                                TEXT      'RATE SHOULD BE 400'
        002A    484F 554C 4420
                4245 2034 3030
000015  0030    2043 504D 2C20                                TEXT      ' CPM, PUNCHED $'
        0033    5055 4E43 4845
                4420 2400
000016  0038    0022                                T33       DC        34
000017  0039    5354 4154 5553                                TEXT      'STATUS ERROR CARDS'
        003C    2045 5252 4F52
                2043 4152 4453
000018  0042    204F 4646 5345                                TEXT      ' OFFSET STACKED$'
        0045    5420 5354 4143
                4B45 4424
000019  004A    000E                                T36       DC        14
000020  004B    2C20 4552 524F                                TEXT      ', ERROR EJECT$'
        004E    5220 454A 4543
                5424
000021  0052    0007                                T38       DC        7
000022  0053    5645 5249 4659                                TEXT      'VERIFY$'
        0056    2400
000023          0057                                ZVKJLF    EQU       $
000024  0057    8F00 0001       K                             SAVE      <ZKCOM+1,Z'0011'
        0059    0011
000025                                              *         'BEGIN'
000026  005A    0F81 019A                                     B         L1
000027                                              *             'INTEGER' TEMP,I, COL 8;
000028  005C    DFC0 FFA3                           KJLF      STB       5,D
000029                                              *             'IF' ID 'EQ' 'HEX' (2008)
000030                                              *             'THEN'
000031                                              *                 'GO TO' END LETTER F;
000032  005E    E840 0000       P                             LDR       6,ID
000033  0060    E970 2008                                     CMR       6,=8200
000034  0062    0901 0190                                     BE        L39
000035                                              *         LABEL 1 := 'LITERAL' (F1);
000036                                              *     'COMMENT' CLEAR BUFFER 1 AND 2;
000037  0064    C870 4631                                     LDR       4,=17969
000038  0066    CF40 0000       P                             STR       4,LABEL1
000039  0068    FBC0 FF9B                                     LAB       7,D+4
000040  006A    9BC0 0000       P                             LAB       1,BUFF1
000041  006C    9F87                                          STB       1,$B7
000042  006D    5C54                                          LDV       5,84
000043  006E    DF40 FF96                                     STR       5,D+5
000044  0070    F870 00A4                                     LDR       7,=164
000045  0072    FF40 FF93                                     STR       7,D+6
000046                                              *             FILL BUFFER (BUFFER 1,164,0);
000047  0074    8740 FF92                                     CL        D+7
000048  0076    D3C0 0000       P                             LNJ       5,JFB
000049                                              *             R OR V MODE := 'LITERAL' (R);
000050  0078    6C52                                          LDV       6,82
000051  0079    EF40 0000       P                             STR       6,RORVMD
000052                                              *             READ OR PUNCH := PUNCH @0@;
000053  007B    C800 0010       X                             LDR       4,<IOWORD+16
000054  007D    CF40 0000       P                             STR       4,READOR
000055  007F    FBC0 FF84                                     LAB       7,D+4
000056  0081    9BC0 0000       P                             LAB       1,BINARY
000057  0083    9F87                                          STB       1,$B7
000058                                              *             INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000059  0084    ABC0 0000       P                             LAB       2,IOWORD
000060  0086    AFC0 FF7E                                     STB       2,D+5
000061  0088    D3C0 0000       P                             LNJ       5,JIO
000062  008A    FBC0 FF79                                     LAB       7,D+4
000063  008C    9BC0 0000       P                             LAB       1,SETTOH
000064  008E    9F87                                          STB       1,$B7
000065                                              *             INOUT (SET TO HOLLERITH, OUTPUT TASK @0@);
000066  008F    0F00 0005       X               N1            NOP       <IOWORD+5
000067  0091    E840 FFFE                                     LDR       6,N1+1
000068  0093    EF40 FF71                                     STR       6,D+5
000069  0095    D3C0 0000       P                             LNJ       5,JIO
000070                                              *             STATUS SHOULD BE := 'HEX' (8000);
000071  0097    E870 8000                                     LDR       6,=-32768
000072  0099    EF40 0000       P                             STR       6,STATUS
000073                                              *             LOW ADDRESS := ZVDLR;
000074  009B    C840 0000       P                             LDR       4,ZV$LR
000075  009D    CF40 0000       P                             STR       4,LOWADD
000076                                              *             ZHRTCC := 0;
000077  009F    8740 0000       P                             CL        ZHRTCC
000078                                              *             ZHRTCI := 0;
000079  00A1    8740 0000       P                             CL        ZHRTCI
000080                                              *             'IF' MAXIMUM CARDS 'LT' 999
000081                                              *             'THEN'
000082                                              *                 TEMP := MAXIMUM CARDS
000083  00A3    D840 0000       P                             LDR       5,MAXIMU
000084  00A5    D970 03E7                                     CMR       5,=999
000085  00A7    0884                                          BAGE      >L5
000086                                              *             'ELSE'
000087  00A8    DF40 FF58                                     STR       5,D+1
000088  00AA    0F85                                          B         >L6
000089          00AB                                L5        EQU       $
000090                                              *                 TEMP := 999;
000091                                              *         ASK AGAIN;
000092  00AB    E870 03E7                                     LDR       6,=999
000093  00AD    EF40 FF53                                     STR       6,D+1
000094          00AF                                L6        EQU       $
000095                                              *             PUT NEW TEXT ("PUNCH 400 CPM - MAX $");
```

```
000096  00AF  FBC0 FF54                    LAB     7,D+4
000097  00B1  9BC0 FF59                    LAB     1,T8
000098  00B3  9F87                         STB     1,$B7
000099  00B4  D3C0 0000      P             LNJ     5,VKPNT
000100  00B6  FBC0 FF4D                    LAB     7,D+4
000101  00B8  E840 FF48                    LDR     6,D+1
000102  00BA  EF07                         STR     6,$B7
000103  00BB  D3C0 0000      P             LNJ     5,JPN
000104                        *               PUT NUMBER (TEMP);
000105  00BD  FBC0 FF46                    LAB     7,D+4
000106  00BF  9BC0 FF57                    LAB     1,T9
000107  00C1  9F87                         STB     1,$B7
000108  00C2  D3C0 0000      P             LNJ     5,VKAQ
000109                        *               ASK QUESTION (" CARDS, TOTAL$");
000110                        *               GET DECIMAL (CARDS SPECIFIED);
000111  00C4  FBC0 FF3F                    LAB     7,D+4
000112  00C6  9BC0 0000      P             LAB     1,CARDSS
000113  00C8  9F87                         STB     1,$B7
000114  00C9  D3C0 0000      P             LNJ     5,VKGD
000115                        *               'IF' CARDS SPECIFIED 'GT' TEMP
000116                        *               'THEN'
000117                        *                   'GO TO' ASK AGAIN;
000118  00CB  E840 0000      P             LDR     6,CARDSS
000119  00CD  E940 FF33                    CMR     6,D+1
000120  00CF  0A60                         BAG     >L6
000121                        *               PUT NEW TEXT ("READY PUNCH$");
000122  00D0  FBC0 FF33                    LAB     7,D+4
000123  00D2  9BC0 FF4C                    LAB     1,T11
000124  00D4  9F87                         STB     1,$B7
000125  00D5  D3C0 0000      P             LNJ     5,VKPNT
000126  00D7  D3C0 0000      P             LNJ     5,JWR
000127                        *               WAIT FOR RETURN;
000128                        *               CHECK STATE (SHOULD BE ON);
000129  00D9  FBC0 FF2A                    LAB     7,D+4
000130  00DB  6C01                         LDV     6,1
000131  00DC  EF07                         STR     6,$B7
000132  00DD  D3C0 0000      P             LNJ     5,JCS
000133  00DF  D3C0 0000      P             LNJ     5,JSE
000134                        *               CHECK STATUS FOR ERRORS;
000135                        *               LABEL 1 := 'LITERAL' (F2);
000136  00E1  E870 4632                    LDR     6,=17970
000137  00E3  EF40 0000      P             STR     6,LABEL1
000138                        *               BUFFER 1 @7@ := 'HEX' (100);
000139  00E5  C870 0100                    LDR     4,=256
000140  00E7  CF40 0007      P             STR     4,BUFF1+7
000141  00E9  FBC0 FF1A                    LAB     7,D+4
000142  00EB  9BC0 0000      P             LAB     1,BUFF1
000143  00ED  9F87                         STB     1,$B7
000144  00EE  0F00 0010      X     N2      NOP     <IOWORD+16
000145  00F0  D840 FFFE                    LDR     5,N2+1
000146  00F2  DF40 FF12                    STR     5,D+5
000147                        *               IOLD (BUFFER 1@0@,PUNCH @0@+160);  (PUNCH NO 1 CARD)
000148  00F4  F870 00A0                    LDR     7,=160
000149  00F6  FF40 FF0F                    STR     7,D+6
000150  00F8  D3C0 0000      P             LNJ     5,JIL
000151  00FA  D3C0 0000      P             LNJ     5,JSE
000152                        *               CHECK STATUS FOR ERRORS;
000153                        *               CHECK STATE (SHOULD BE ON);
000154  00FC  FBC0 FF07                    LAB     7,D+4
000155  00FE  6C01                         LDV     6,1
000156  00FF  EF07                         STR     6,$B7
000157  0100  D3C0 0000      P             LNJ     5,JCS
000158                        *               ERROR FLAG := 0;
000159  0102  8740 0000      P       CL          ERRORF
000160                        *               CLOCK SEMAPHORE := 0;
000161  0104  8740 0000      P       CL          CKSEM
000162                        *               CARD COUNT := 1;
000163  0106  6C01                         LDV     6,1
000164  0107  EF40 0000      P             STR     6,CARDSR
000165                        *               COL 8 := 'HEX' (80);
000166  0109  C870 0080                    LDR     4,=128
000167  010B  CF40 FEF7                    STR     4,D+3
000168                        *               TEMP := 'HEX'(1000);   (COLUMN VALUE - NO PUNCH)
000169  010D  D870 1000                    LDR     5,=4096
000170  010F  DF40 FEF1                    STR     5,D+1
000171  0111  0004                         DC      4
000172                        *               CLOCK ON;
000173                        *           'COMMENT' ENTER THE 100'S LOOP;
000174  0112  E840 FEEE      L12           LDR     6,D+1
000175  0114  EF40 FEED                    STR     6,D+2
000176  0116  D3C0 0012                    LNJ     5,P13
000177  0118  E870 0100      L14           LDR     6,=256
000178  011A  EF40 FEE7                    STR     6,D+2
000179  011C  D3C0 000C                    LNJ     5,P13
000180  011E  E840 FEE3      L15           LDR     6,D+2
000181  0120  6061                         SAR     6,1
000182  0121  EF40 FEE0                    STR     6,D+2
000183                        *               'FOR' I := TEMP, 'HEX' (100),   I/2 'WHILE' I 'GE' 1
000184                        *               'DO'
000185                        *               'BEGIN'
000186  0123  6D01                         CMV     6,1
000187  0124  0801 006D                    BAL     L16
000188  0126  D3C0 0002                    LNJ     5,P13
000189  0128  0FF6                         B       >L15
000190  0129  DFC0 FEDA      P13           STB     5,D+4
000191                        *               'INTEGER' J;
000192                        *           'COMMENT' ENTER THE 10'S LOOP;
000193  012B  E840 FED5      L17           LDR     6,D+1
000194  012D  EF40 FED7                    STR     6,D+5
000195  012F  D3C0 0012                    LNJ     5,P18
000196  0131  E870 0100      L19           LDR     6,=256
000197  0133  EF40 FED1                    STR     6,D+5
000198  0135  D3C0 000C                    LNJ     5,P18
000199  0137  E840 FECD      L20           LDR     6,D+5
000200  0139  6061                         SAR     6,1
000201  013A  EF40 FECA                    STR     6,D+5
000202                        *                   'FOR' J:=TEMP,'HEX'(100), J/2 'WHILE' J 'GE' 1
000203                        *                   'DO'
000204                        *                   'BEGIN'
000205  013C  6D01                         CMV     6,1
000206  013D  0801 0052                    BAL     L21
000207  013F  D3C0 0002                    LNJ     5,P18
000208  0141  0FF6                         B       >L20
```

```
000209  0142  DFC0 FEC3          P18     STB     5,D+6
000210                            *                       'INTEGER' K;
000211                            *             'COMMENT' ENTER THE 1'S LOOP;
000212  0144  E840 FEBE          L22     LDR     6,D+3
000213  0146  EF40 FEC0                  STR     6,D+7
000214  0148  D3C0 000B                  LNJ     5,P23
000215  014A  E840 FEBC          L24     LDR     6,D+7
000216  014C  6061                      SAR     6,1
000217  014D  EF40 FEB9                  STR     6,D+7
000218                            *                       'FOR' K := COL 8, K/2 'WHILE' K 'GE' 1
000219                            *                           'DO'
000220                            *                           'BEGIN'
000221  014F  6D01                      CMV     6,1
000222  0150  0838                      BAL     >L25
000223  0151  D3C0 0002                  LNJ     5,P23
000224  0153  0FF7                      B       >L24
000225  0154  DFC0 FEB3          P23     STB     5,D+8
000226                            *                               'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000227                            *                               'THEN'
000228                            *                               'BEGIN'
000229  0156  E840 0000       P          LDR     6,CARDSR
000230  0158  6B0E                      BEVN    6,>L26
000231                            *                                   BUFFER 2@5@  := I; (COLUMN  6 = I)
000232  0159  C840 FEA8                  LDR     4,D+2
000233  015B  CF40 0005       P          STR     4,BUFF2+5
000234                            *                                   BUFFER 2@6@  := J; (COLUMN  7 = J)
000235  015D  E840 FEA7                  LDR     6,D+5
000236  015F  EF40 0006       P          STR     6,BUFF2+6
000237                            *                                   BUFFER 2@7@  := K; (COLUMN  8 = K)
000238                            *                               'END'
000239  0161  D840 FEA5                  LDR     5,D+7
000240  0163  DF40 0007       P          STR     5,BUFF2+7
000241                            *                               'ELSE'
000242                            *                               'BEGIN'
000243  0165  0F8D                      B       >L27
000244        0166                L26     EQU     $
000245                            *                                   BUFFER 1@5@  := I;
000246  0166  E840 FE9B                  LDR     6,D+2
000247  0168  EF40 0005       P          STR     6,BUFF1+5
000248                            *                                   BUFFER 1@6@  := J;
000249  016A  C840 FE9A                  LDR     4,D+5
000250  016C  CF40 0006       P          STR     4,BUFF1+6
000251                            *                                   BUFFER 1@7@  := K;
000252  016E  D840 FE98                  LDR     5,D+7
000253  0170  DF40 0007       P          STR     5,BUFF1+7
000254                            *                               'END';
000255        0172                L27     EQU     $               GENERATE CHECKSUMS (READ OR PUNCH);
000256                            *
000257                            *     'COMMENT' WHILE PUNCHING ONE CARD GENERATE CHECKSUMS ON PREVIOUS CARD;
000258  0172  FBC0 FE96                  LAB     7,D+9
000259  0174  E840 0000       P          LDR     6,READOR
000260  0176  EF07                      STR     6,$B7
000261  0177  D3C0 0000       P          LNJ     5,JGC
000262  0179  D3C0 0000       P          LNJ     5,JSE
000263                            *                               CHECK STATUS FOR ERRORS;
000264                            *                               CARD COUNT := CARD COUNT +1;
000265  017B  8AC0 0000       P          INC     CARDSR
000266                            *                               'IF' CARD COUNT 'GE' CARDS SPECIFIED
000267  017D  E840 0000       P          LDR     6,CARDSR
000268  017F  E940 0000       P          CMR     6,CARDSS
000269  0181  0891                      BAGE    >L16
000270                            *                               'OR' GET BREAK 'EQ' TRUE
000271  0182  D3C0 0000       P          LNJ     5,VKGB
000272                            *                               'THEN'
000273                            *                                   'GO TO' STOP PUNCHING;
000274  0184  6D01                      CMV     6,1
000275  0185  090D                      BE      >L16
000276                            *                       'END';      (OF 1'S LOOP)
000277  0186  83C8 FE81                  JMP     *D+8
000278        0188                L25     EQU     $
000279                            *                           COL 8 := 'HEX' (200);  (SET FOR 0 PUNCH)
000280  0188  E870 0200                  LDR     6,=512
000281  018A  EF40 FE78                  STR     6,D+3
000282                            *                       TEMP := 'HEX' (200);  (SET ROW 0 PUNCH)
000283  018C  EF40 FE74                  STR     6,D+1
000284                            *                       'END';      (OF 10'S LOOP)
000285  018E  83C8 FE77                  JMP     *D+6
000286                            *             'END';                (OF 100'S LOOP)
000287                            *           STOP PUNCHING;
000288  0190  83C8 FE73          L21     JMP     *D+4
000289        0192                L16     EQU     $
000290  0192  0005                      DC      5
000291                            *               CLOCK OFF;
000292                            *               READ OR PUNCH := FALSE; (DON'T PUNCH, JUST CHK LAST CARD)
000293  0193  8740 0000       P          CL      READOR
000294                            *               GENERATE CHECKSUMS (READ OR PUNCH);
000295  0195  FBC0 FE6E                  LAB     7,D+4
000296  0197  E840 0000       P          LDR     6,READOR
000297  0199  EF07                      STR     6,$B7
000298  019A  D3C0 0000       P          LNJ     5,JGC
000299  019C  FBC0 FE67                  LAB     7,D+4
000300  019E  9BC0 FE87                  LAB     1,T30
000301  01A0  9F87                      STB     1,$B7
000302  01A1  D3C0 0000       P          LNJ     5,VKPNT
000303                            *               PUT NEW TEXT ("RATE SHOULD BE 400 CPM, PUNCHED $");
000304                            *               CALCULATE ELAPSED TIME (CARD COUNT);
000305  01A3  FBC0 FE60                  LAB     7,D+4
000306  01A5  E840 0000       P          LDR     6,CARDSR
000307  01A7  EF07                      STR     6,$B7
000308  01A8  D3C0 0000       P          LNJ     5,JET
000309                            *             'IF' ERROR FLAG 'NE' 0
000310                            *               'THEN'
000311                            *             'BEGIN'
000312                            *           WRONG RESPONSE;
000313  01AA  E840 0000       P          LDR     6,ERRORF
000314  01AC  6926                      BEZ     6,>L31
000315                            *                   ASK NEW QUESTION
000316  01AD  FBC0 FE56          L32     LAB     7,D+4
000317  01AF  9BC0 FE88                  LAB     1,T33
000318  01B1  9F87                      STB     1,$B7
000319  01B2  D3C0 0000       P          LNJ     5,VKANQ
000320                            *                       ("STATUS ERROR CARDS OFFSET STACKED$");
000321                            *                       TEMP := 'HEX' (590D);
```

```
000322  01B4  E870 590D                    LDR     6,=22797
000323  01B6  EF40 FE4A                    STR     6,D+1
000324                          *                   GET CHAR (TEMP);
000325  01B8  FBC0 FE4B                    LAB     7,D+4
000326  01BA  9BC0 FE46                    LAB     1,D+1
000327  01BC  9F87                         STB     1,$B7
000328  01BD  D3C0 0000       P            LNJ     5,VKGC
000329                          *                       'IF' TEMP 'NE' 'LITERAL' (Y)
000330                          *                       'THEN'
000331                          *                       'BEGIN'
000332  01BF  E840 FE41                    LDR     6,D+1
000333  01C1  6D59                         CMV     6,89
000334  01C2  0910                         BE      >L31
000335                          *                               'IF' TEMP 'NE' 'LITERAL' (N)
000336                          *                               'THEN'
000337                          *                                   'GO TO' WRONG RESPONSE;
000338  01C3  6D4E                         CMV     6,78
000339  01C4  09E9                         BNE     >L32
000340                          *                               LABEL 2 := 'LITERAL' (OS);
000341  01C5  C870 4F53                    LDR     4,=20307
000342  01C7  CF40 0000       P            STR     4,LABEL2
000343  01C9  D3C0 0000       P            LNJ     5,JRE
000344                          *                               REPORT ERROR;
000345  01CB  FBC0 FE38                    LAB     7,D+4
000346  01CD  9BC0 FE7C                    LAB     1,T36
000347  01CF  9F87                         STB     1,$B7
000348  01D0  D3C0 0000       P            LNJ     5,VKPT
000349                          *                               PUT TEXT (", ERROR EJECT$");
000350                          *                           'END';
000351                          *               'END';
000352        01D2              L31         EQU     $
000353                          *           CARDS SPECIFIED := CARD COUNT;
000354                          *       VERIFY:
000355  01D2  E840 0000       P            LDR     6,CARDSR
000356  01D4  EF40 0000       P            STR     6,CARDSS
000357        01D6              L37         EQU     $
000358                          *           R OR V MODE := 'LITERAL' (V);
000359  01D6  6C56                         LDV     6,86
000360  01D7  EF40 0000       P            STR     6,RORVMD
000361                          *           ASK NEW QUESTION ("VERIFY$");
000362  01D9  FBC0 FE2A                    LAB     7,D+4
000363  01DB  9BC0 FE76                    LAB     1,T38
000364  01DD  9F87                         STB     1,$B7
000365  01DE  D3C0 0000       P            LNJ     5,VKANQ
000366                          *               TEMP := 'HEX' (590D);
000367  01E0  E870 590D                    LDR     6,=22797
000368  01E2  EF40 FE1E                    STR     6,D+1
000369                          *           GET CHAR (TEMP);
000370  01E4  FBC0 FE1F                    LAB     7,D+4
000371  01E6  9BC0 FE1A                    LAB     1,D+1
000372  01E8  9F87                         STB     1,$B7
000373  01E9  D3C0 0000       P            LNJ     5,VKGC
000374                          *           'IF' TEMP 'NE' 'LITERAL' (N)
000375                          *           'THEN'
000376                          *           'BEGIN'
000377  01EB  E840 FE15                    LDR     6,D+1
000378  01ED  6D4E                         CMV     6,78
000379  01EE  0905                         BE      >L39
000380                          *                   'IF' TEMP 'NE' 'LITERAL' (Y)
000381                          *                   'THEN'
000382                          *                       'GO TO' VERIFY;
000383  01EF  6D59                         CMV     6,89
000384  01F0  09E6                         BNE     >L37
000385                          *                   LETTER RP;
000386  01F1  D3C0 0000       P            LNJ     5,KJLR
000387                          *           'END';
000388                          *       END LETTER F;
000389                          *       'END';  (OF PROCEDURE)
000390                          *       'END';  (OF SEGMENT)
000391                          *       'FINISH'
000392  01F3  83C8 FE0C                    JMP     *D
000393        01F5              L1          EQU     $
000394  01F5  83C0 0000       P            JMP     ZK$EX
000395                                     XLOC    ZK$EX
000396                                     CTRL LINK ZK$EX
000397                                     XLOC    ZV$LR
000398                                     CTRL LINK ZV$LR
000399                                     XLOC    ZHRTCI
000400                                     XLOC    ZHRTCC
000401                                     XLOC    VKPT
000402                                     XLOC    VKPNT
000403                                     XLOC    VKAQ
000404                                     XLOC    VKANQ
000405                                     XLOC    VKGC
000406                                     XLOC    VKGD
000407                                     XLOC    VKGB
000408                                     XLOC    ID
000409                                     XLOC    STATUS
000410                                     XLOC    SETTOH
000411                                     XLOC    BINARY
000412                                     XLOC    LABEL1
000413                                     XLOC    LABEL2
000414                                     XLOC    MAXIMU
000415                                     XLOC    CARDSS
000416                                     XLOC    CARDSR
000417                                     XLOC    LOWADD
000418                                     XLOC    RORVMD
000419                                     XLOC    ERRORF
000420                                     XLOC    CKSEM
000421                                     XLOC    READOR
000422                                     XLOC    BUFF1
000423                                     XLOC    BUFF2
000424                                     XLOC    IOWORD
000425                                     XLOC    JGC
000426                                     XLOC    JRE
000427                                     XLOC    JIO
000428                                     XLOC    JIL
000429                                     XLOC    JPN
000430                                     XLOC    JWR
000431                                     XLOC    JSE
000432                                     XLOC    JCS
000433                                     XLOC    JET
000434                                     XLOC    JFB
```

```
    000435                                          XLOC    KJLR
    000436  01F7  0057                              END     ZVKJLF,ZVKJLF
0000  ERR COUNT
L
```

```
E   ZVKJLI.LIST                          05/19/78  1427.2R W 05/19/78  1422.4      4343220000
000001                                              TITLE     ZVKJLI,'REV 00'
000002          0062                                XDEF      KJLI
000003          005D                                XDEF      ZVKJLI
000004          0008                        ZKCOM   COMM      8
000005  0000    0000                        D       RESV      9,0
000006  0009    000A                        T3      DC        10
000007  000A    5255 5054 2054                      TEXT      'RUPT TEST$'
        000D    4553 5424
000008  000F    001F                        T6      DC        31
000009  0010    2C20 494E 5445                      TEXT      ', INTERRUPT REGIST'
        0013    5252 5550 5420
                5245 4749 5354
000010  0019    4552 2049 4E43                      TEXT      'ER INCORRECT$'
        001C    4F52 5245 4354
                2400
000011  0020    001F                        T9      DC        31
000012  0021    2C20 494E 5445                      TEXT      ', INTERRUPT REGIST'
        0024    5252 5550 5420
                5245 4749 5354
000013  002A    4552 2049 4E43                      TEXT      'ER INCORRECT$'
        002D    4F52 5245 4354
                2400
000014  0031    001C                        T28     DC        28
000015  0032    2C20 4449 444E                      TEXT      ', DIDN''T RUPT - SH'
        0035    2754 2052 5550
                5420 2D20 5348
000016  003B    4F55 4C44 2048                      TEXT      'OULD HAVE$'
        003E    4156 4524
000017  0040    0016                        T31     DC        22
000018  0041    2C20 5255 5054                      TEXT      ', RUPT WHEN SHOULD'
        0044    2057 4845 4E20
                5348 4F55 4C44
000019  004A    4E5E 5424                           TEXT      'NJT$'
000020  004C    001F                        T35     DC        31
000021  004D    2C20 5045 4E44                      TEXT      ', PENDING RUPT WHE'
        0050    494E 4720 5255
                5054 2057 4845
000022  0056    4E20 4350 2027                      TEXT      'N CP ''NE'' 63$'
        0059    4E45 2720 3633
                2400
000023          005D                        ZVKJLI  EQU       $
000024  005D    8F00 0001           K               SAVE      <ZKCOM+1,Z'0011'
        005F    0011
000025                                      *       'BEGIN'
000026  0060    0F81 02AE                           B         L1
000027                                      *       'INTEGER' TEMP,
000028                                      *                 I;
000029  0062    DFC0 FF9D                   KJLI    STB       5,D
000030                                      *         LABEL1 := 'LITERAL' (I1);
000031  0064    E870 4931                           LDR       6,=18737
000032  0066    EF40 0000           P               STR       6,LABEL1
000033                                      *         PUT NEW TEXT ("RUPT TEST$");
000034  0068    FBC0 FF9A                           LAB       7,D+3
000035  006A    9BC0 FF9E                           LAB       1,T3
000036  006C    9F87                                STB       1,$B7
000037  006D    D3C0 0000           P               LNJ       5,VKPNT
000038                                      *         CHECK STATE (SHOULD BE OFF);
000039  006F    FBC0 FF93                           LAB       7,D+3
000040  0071    8707                                CL        $B7
000041  0072    D3C0 0000           P               LNJ       5,JCS
000042                                      *         STATUS SHOULD BE := 0;
000043  0074    8740 0000           P               CL        STATUS
000044                                      *           'IF' ID 'EQ' 'HEX' (2008)
000045                                      *           'THEN'
000046  0076    E840 0000           P               LDR       6,ID
000047  0078    E970 2008                           CMR       6,=8200
000048  007A    0983                                BNE       >L4
000049  007B    D3C0 0000           P               LNJ       5,JSE
000050                                      *                 CHECK STATUS ERRORS;
000051                                      **;
000052                                      **    INTERRUPT CONTROL TEST;
000053                                      **;
000054          007D                        L4      EQU       $
000055                                      *         TEMP := 'HEX' (AAAA);
000056  007D    E870 AAAA                           LDR       6,=-21846
000057  007F    EF40 FF81                           STR       6,D+1
000058  0081    FBC0 FF81                           LAB       7,D+3
000059  0083    9BC0 FF7D                           LAB       1,D+1
000060  0085    9F87                                STB       1,$B7
000061                                      *                 INOUT (TEMP,OUTPUT INTERRUPT @0@);
000062  0086    0F00 0002           X       N1      NOP       <IOWORD+2
000063  0088    C840 FFFE                           LDR       4,N1+1
000064  008A    CF40 FF79                           STR       4,D+4
000065  008C    D3C0 0000           P               LNJ       5,JIO
000066  008E    FBC0 FF74                           LAB       7,D+3
000067  0090    9BC0 FF70                           LAB       1,D+1
000068  0092    9F87                                STB       1,$B7
000069                                      *                 INOUT (TEMP, INPUT INTERRUPT @0@);
000070  0093    0F00 0001           X       N2      NOP       <IOWORD+1
000071  0095    E840 FFFE                           LDR       6,N2+1
000072  0097    EF40 FF6C                           STR       6,D+4
000073  0099    D3C0 0000           P               LNJ       5,JIO
000074                                      *           'IF' TEMP 'NE' 'HEX' (AAAA)
000075                                      *           'THEN'
000076                                      *           'BEGIN'
000077  009B    E840 FF65                           LDR       6,D+1
000078  009D    E970 AAAA                           CMR       6,=-21846
000079  009F    0901 0014                           BE        L5
000080                                      *                 LABEL 2 := 'LITERAL' (1A);
000081  00A1    C870 4941                           LDR       4,=18753
000082  00A3    CF40 0000           P               STR       4,LABEL2
000083  00A5    D840 0000           P               LDR       $R5,SETTOA
000084  00A7    C840 FF59                           LDR       $R4,D+1
000085                                      *                 LOAD R4 R5 (TEMP, CLEAR);
000086  00A9    D3C0 0000           P               LNJ       5,JRE
000087                                      *                 REPORT ERROR;
000088                                      *                 PUT TEXT (", INTERRUPT REGISTER INCORRECT$");
000089  00AB    FBC0 FF57                           LAB       7,D+3
000090  00AD    9BC0 FF61                           LAB       1,T6
000091  00AF    9F87                                STB       1,$B7
000092  00B0    D3C0 0000           P               LNJ       5,VKPT
000093                                      *                 'GO TO' FRESH START;
```

```
000094  00B2  83C0 0000      P                 JMP       FRESHS
000095                                  *           'END' ;
000096        00B4                      L5      EQU       $
000097                                  *         TEMP := 'HEX' (5555);
000098  00B4  E870 5555                         LDR       6,=21845
000099  00B6  EF40 FF4A                         STR       6,D+1
000100  00B8  FBC0 FF4A                         LAB       7,D+3
000101  00BA  9BC0 FF46                         LAB       1,D+1
000102  00BC  9F87                              STB       1,$B7
000103                                  *         INOUT (TEMP,OUTPUT INTERRUPT @0@);
000104  00BD  0F00 0002      X           N3     NOP       <IOWORD+2
000105  00BF  C840 FFFE                         LDR       4,N3+1
000106  00C1  CF40 FF42                         STR       4,D+4
000107  00C3  D3C0 0000      P                 LNJ       5,JIO
000108  00C5  FBC0 FF3D                         LAB       7,D+3
000109  00C7  9BC0 FF39                         LAB       1,D+1
000110  00C9  9F87                              STB       1,$B7
000111                                  *         INOUT(TEMP, INPUT INTERRUPT @0@);
000112  00CA  0F00 0001      X           N4     NOP       <IOWORD+1
000113  00CC  E840 FFFE                         LDR       6,N4+1
000114  00CE  EF40 FF35                         STR       6,D+4
000115  00D0  D3C0 0000      P                 LNJ       5,JIO
000116                                  *           'IF' TEMP 'NE' 'HEX' (5555)
000117                                  *         'THEN'
000118                                  *         'BEGIN'
000119  00D2  E840 FF2E                         LDR       6,D+1
000120  00D4  E970 5555                         CMR       6,=21845
000121  00D6  0901 0014                         BE        L8
000122                                  *                   LABEL2 := 'LITERAL' (15);
000123  00D8  C870 4935                         LDR       4,=18741
000124  00DA  CF40 0000      P                 STR       4,LABEL2
000125  00DC  D840 0000      P                 LDR       $R5,SETTOA
000126  00DE  C840 FF22                         LDR       $R4,D+1
000127                                  *                   LOAD R4 R5 (TEMP, CLEAR);
000128  00E0  D3C0 0000      P                 LNJ       5,JRE
000129                                  *                   REPORT ERROR;
000130                                  *                   PUT TEXT (', INTERRUPT REGISTER INCORRECT$");
000131  00E2  FBC0 FF20                         LAB       7,D+3
000132  00E4  9BC0 FF3B                         LAB       1,T9
000133  00E6  9F87                              STB       1,$B7
000134  00E7  D3C0 0000      P                 LNJ       5,VKPT
000135                                  *                   'GO TO' FRESH START;
000136  00E9  83C0 0000      P                 JMP       FRESHS
000137                                  *           'END';
000138  00EB  FBC0 FF17                 L8      LAB       7,D+3
000139  00ED  9BC0 0000      P                 LAB       1,BINARY
000140  00EF  9F87                              STB       1,$B7
000141                                  *                   INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000142                                          **;
000143                                          ** TIME THE BINC INSTRUCTION, TO OBTAIN SYSTEM SPEED;
000144                                          **;
000145  00F0  ABC0 0000      P                 LAB       2,IOWORD
000146  00F2  AFC0 FF11                         STB       2,D+4
000147  00F4  D3C0 0000      P                 LNJ       5,JIO
000148  00F6  9FC0 0000      P                 STB       $B1,TEMPCH
000149  00F8  0F80           T                 B         >+$A
000150  00F9  83C0 0000      P                 JMP       TIMER
000151  00F9                                  ORG       $-1-$AF
000152  00FA                        $E        ORG       $+1
000153  00FA                        $B        ORG       $-1+$AF
000154  00FB                        $C        ORG       $+1
000155  00FB  82C0 FFFD             $A        LB        -$E,=X'40'
        00FD  0040
000156  00FE  0581 0000      T                 BBF       +$D
000157  0100  ABC0 FFF9                       LAB       $B2,-$C
000158  0102  9802                            LDR       $R1,$B2
000159  0103  9B92                            LAB       $B1,$B2,$R1
000160  0104  0F81 0000      T                 B         +$F
000161  0106  9BC8 FFF3             $D        LAB       $B1,*-$B
000162  0108  9FC0 0005      P      $F         STB       $B1,ISAC+4+$AF
000163  010A  9CC0 0000      P                 LDB       $B1,TEMPCH
000164                                  *                   ADDRESS OF LABEL (ISACP,TIMER);
000165                                  *                   ZHRTCC := 100;
000166  010C  6C64                             LDV       6,100
000167  010D  EF40 0000      P                 STR       6,ZHRTCC
000168                                  *                   ZHRTCI := 100;
000169  010F  EF40 0000      P                 STR       6,ZHRTCI
000170                                  *                   ZHRTCL :=  20;
000171                                          **;
000172                                          **  COUNT IN R6,R7 UNTIL INTERRUPTED;
000173                                          **;
000174  0111  4C14                             LDV       4,20
000175  0112  CF40 0000      P                 STR       4,ZHRTCL
000176  0114  0004                             DC        4
000177                                  *                   CLOCK ON;
000178                                  *                   'CODE''BEGIN']
000179                                  *                             CL =$R6
000180                                  *                             CL =$R7
000181  0115  8756                             CL        =$R6
000182  0116  8757                             CL        =$R7
000183                                  *                   ]CBJ
000184                                  *         LOOP      BINC $R7,$
000185                                  *                   BINC $R6,<LOOP
000186  0117  7781 FFFF             LOOP        BINC      $R7,$
000187  0119  6780 0117                         BINC      $R6,<LOOP
000188                                  *                   ]CBJ
000189                                  *                   XDEF STOPTI,RUPTI
000190                                  *         STOPTI    NOP >$-1
000191                                  *                 ]'END';
000192        011B                        XDEF            STOPTI,RUPTI
              0242
000193  011B  0F7F                 STOPTI  NOP       >$-1
000194  011C  0005                         DC        5
000195                                  *                   CLOCK OFF;
000196                                  *                   'CODE''BEGIN']
000197                                  *                   DIV $R7,=10000 NUMBER OF BINCS / 83 USEC AT 60 HZ
000198                                  *                   STR $R7,]SPEED
000199  011D  F370 2710                         DIV       $R7,=10000
000200  011F  FF40 0000      P                 STR       $R7,SPEED
000201  0121  9FC0 0000      P                 STB       $B1,TEMPCH
000202  0123  0F80           T                 B         >+$A
000203  0124  83C0 0000      P                 JMP       CPRUPT
000204  0126                                  ORG       $-1-$AF
```

```
000205  0125                            $E      ORG     $+1
000206  0125                            $B      ORG     $-1+$AF
000207  0126                            $C      ORG     $+1
000208  0126  82C0 FFFD                $A      LB      -$E,=X'40'
        0128  0040
000209  0129  0581 0000      T                 BBF     +$D
000210  012B  ABC0 FFF9                        LAB     $B2,-$C
000211  012D  9802                            LDR     $R1,$B2
000212  012E  9B92                            LAB     $B1,$B2.$R1
000213  012F  0F81 0000      T                 B       +$F
000214  0131  9BC8 FFF3      $D                LAB     $B1,*-$B
000215  0133  9FC0 0005      P   $F            STB     $B1,ISAP+4+$AF
000216  0135  9CC0 0000      P                 LDB     $B1,TEMPCH
000217        0137                      L12     EQU     $
000218                                  *               'FOR' I:= 0 'STEP' 1 'UNTIL' 63
000219  0137  8740 FECA                        CL      D+2
000220                                  *               'DO'
000221  0139  E840 FEC8                L13     LDR     6,D+2
000222  013B  6EC1                            ADV     6,-63
000223  013C  6A14                            BGZ     6,>L14
000224  013D  9840 FEC4                        LDR     1,D+2
000225  013F  9B40 0000      P                 MUL     1,ZV$AF
000226  0141  0F00 0000      X   N5            NOP     <ZHISAZ
000227  0143  9A40 FFFE                        ADD     1,N5+1
000228                                  *               @'LOCATION' (ZHISAZ)+I*ZVDAF@ := 'LOCATION'(ISARD@0@);
000229  0145  9F40 FEBD                        STR     1,D+3
000230  0147  0F00 0002      X   N6            NOP     <ISAD+2
000231  0149  E840 FFFE                        LDR     6,N6+1
000232  014B  EF48 FEB7                        STR     6,*D+3
000233  014D  8AC0 FEB4                        INC     D+2
000234  014F  0FEA                            B       >L13
000235        0150                      L14     EQU     $
000236  0150  9FC0 0000      P                 STB     $B1,TEMPCH
000237  0152  0F80            I                 B       >+$A
000238  0153  83C0 0000      P                 JMP     DEVIH
000239  0153                                    ORG     $-1-$AF
000240  0154                            $E      ORG     $+1
000241  0154                            $B      ORG     $-1+$AF
000242  0155                            $C      ORG     $+1
000243  0155  82C0 FFFD                $A      LB      -$E,=X'40'
        0157  0040
000244  0158  0581 0000      T                 BBF     +$D
000245  015A  ABC0 FFF9                        LAB     $B2,-$C
000246  015C  9802                            LDR     $R1,$B2
000247  015D  9B92                            LAB     $B1,$B2.$R1
000248  015E  0F81 0000      T                 B       +$F
000249  0160  9BC8 FFF3      $D                LAB     $B1,*-$B
000250  0162  9FC0 0005      P   $F            STB     $B1,ISAD+4+$AF
000251  0164  9CC0 0000      P                 LDB     $B1,TEMPCH
000252  0166  6C3F                      L16     LDV     6,63
000253  0167  EF40 0000      P                 STR     6,CPLEVE
000254                                  *               'FOR' CPLEVEL:=63 'STEP' -1 'UNTIL' 0
000255                                  *               'DO'
000256                                  *               'BEGIN'
000257  0169  E840 0000      P   L17           LDR     6,CPLEVE
000258  016B  6801 0139                        BLZ     6,L18
000259                                  *               'INTEGER' TEMP;
000260                                  **  RUPT B;
000261                                  *               ZHIAFB := 0;                    (CLEAR ACTIVITY FLAGS)
000262  016D  8740 0000      P                 CL      ZHIAFB
000263  016F  9BC0 0000      P                 LAB     1,ZHIAFB
000264  0171  9FC0 FE92                        STB     1,D+4
000265                                  *               @'LOCATION' (ZHIAFB)+1@ := 0;
000266  0173  1C01                            LDV     1,1
000267  0174  8718 0004                        CL      *<D+4.$R1
000268  0176  0F00 0000      X   N7            NOP     <ZHIAFB
000269  0178  A840 FFFE                        LDR     2,N7+1
000270  017A  AA40 0000      P                 ADD     2,ZHIAFB
000271                                  *               @'LOCATION' (ZHIAFB)+2+ZHIAFD@ := 0;
000272  017C  AF40 FE87                        STR     2,D+4
000273  017E  3C02                            LDV     3,2
000274  017F  8738 0004                        CL      *<D+4.$R3
000275  0181  0F00 0000      X   N8            NOP     <ZHIAFB
000276  0183  A840 FFFE                        LDR     2,N8+1
000277  0185  AA40 0000      P                 ADD     2,ZHIAFB
000278                                  *               @'LOCATION' (ZHIAFB)+3+ZHIAFD@ := 0;
000279  0187  AF40 FE7C                        STR     2,D+4
000280  0189  1C03                            LDV     1,3
000281  018A  8718 0004                        CL      *<D+4.$R1
000282  018C  E840 0000      P                 LDR     6,CPLEVE
000283  018E  E470 8080                        OR      6,=-32640
000284  0190  EF40 0000      P                 STR     6,LEVTYP
000285  0192  8E40 0000      P                 LEV     LEVTYP
000286                                  *               LEV ('HEX'(8080),CPLEVEL);      (RUN AT NEW CP LEVEL)
000287                                  **  FIRST TIME - SET IV TO POINT TO PROGRAM ISA;
000288                                  **  OTHER TIMES - SET IV TO POINT TO DEVICE ISA;
000289                                  *               'IF' CP LEVEL 'EQ' 63
000290                                  *               'THEN'
000291  0194  E840 0000      P                 LDR     6,CPLEVE
000292  0196  6D3F                            CMV     6,63
000293  0197  0991                            BNE     >L19
000294  0198  9840 0000      P                 LDR     1,ZV$AF
000295  019A  1F3F                            MLV     1,63
000296  019B  0F00 0000      X   N9            NOP     <ZHISAZ
000297  019D  9A40 FFFE                        ADD     1,N9+1
000298                                  *                       @'LOCATION' (ZHISAZ)+63*ZVDAF@ :='LOCATION'(ISAID @0@)
000299                                  *               'ELSE'
000300                                  **  RUPT C;
000301  019F  9F40 FE64                        STR     1,D+4
000302  01A1  0F00 0002      X   N10           NOP     <ISAP+2
000303  01A3  C840 FFFE                        LDR     4,N10+1
000304  01A5  CF48 FE5E                        STR     4,*D+4
000305  01A7  0F92                            B       >L20
000306        01A8                      L19     EQU     $
000307                                  *                       @'LOCATION'(ZHISAZ)+(CPLEVEL+1)*ZVDAF@;=
000308  01A8  9840 0000      P                 LDR     1,CPLEVE
000309  01AA  1E01                            ADV     1,1
000310  01AB  9B40 0000      P                 MUL     1,ZV$AF
000311  01AD  0F00 0000      X   N11           NOP     <ZHISAZ
000312  01AF  9A40 FFFE                        ADD     1,N11+1
000313                                  *                               'LOCATION' (ISARD @0@);
000314                                  **  RUPT E;
000315                                  ** DEVICE LEVEL LOOP;
```

```
000316  01B1  9F40 FE53            STR     1,D+5
000317  01B3  0F00 0002    X       N12     NOP     <ISAD+2
000318  01B5  E840 FFFE            LDR     6,N12+1
000319  01B7  EF48 FE4D            STR     6,*D+5
000320        01B9          L20     EQU     $
000321  01B9  6C3F        L21     LDV     6,63
000322  01BA  EF40 0000    P       STR     6,DLVL
000323                        *               'FOR' DEVICE LEVEL:=63 'STEP' -1 'UNTIL' 0
000324                        *               'DO'
000325                        *               'BEGIN'
000326  01BC  E840 0000    P       L22     LDR     6,DLVL
000327  01BE  6801 00E2            BLZ     6,L23
000328                        *                       'INTEGER' CPID;
000329                        **  RUPT F;
000330                        *                       DEVICE SEMAPHORE := 0;  (CLEAR IV HANDLER FLAG)
000331  01C0  8740 0000    P               CL      DVSEM
000332  01C2  C840 0000    P               LDR     4,SREG
000333  01C4  C570 03C0                    AND     4,=960
000334  01C6  C440 0000    P               OR      4,DLVL
000335  01C8  CF40 FE3B                    STR     4,D+4
000336                        *                       CPID := SREG 'MASK' 'HEX' (03C0) 'UNION' DEVICELEVEL;
000337  01CA  FBC0 FE3A                    LAB     7,D+5
000338  01CC  9BC0 FE37                    LAB     1,D+4
000339  01CE  9F87                         STB     1,$B7
000340                        *                       INOUT (CPID, OUTPUT INTERRUPT @0@);
000341                        **;
000342                        **  NOW AN ATTEMPT IS MADE TO GET AN INTERRUPT VIA STOP I/O;
000343                        **;
000344  01CF  0F00 0002    X       N13     NOP     <IOWORD+2
000345  01D1  D840 FFFE            LDR     5,N13+1
000346  01D3  DF40 FE32            STR     5,D+6
000347  01D5  D3C0 0000    P       LNJ     5,JIO
000348                        *               TEMP := 'HEX' (4000);
000349  01D7  E870 4000            LDR     6,=16384
000350  01D9  EF40 FE29            STR     6,D+3
000351  01DB  FBC0 FE29            LAB     7,D+5
000352  01DD  9BC0 FE25            LAB     1,D+3
000353  01DF  9F87                 STB     1,$B7
000354                        *               INOUT (TEMP, OUTPUT CONTROL @0@);
000355  01E0  ABC0 0000    P       LAB     2,IOWORD
000356  01E2  AFC0 FE23            STB     2,D+6
000357  01E4  D3C0 0000    P       LNJ     5,JIO
000358                        *                       'CODE''BEGIN']
000359  01E6  F840 0000    P       LDR     $R7,SPEED
000360  01E8  7701 FFFF            BDEC    $R7,$
000361                        *                               ]CB]
000362                        *                               NOP $]
000363  01EA  0F01 FFFF            NOP     $
000364                        *               'IF' DEVICE LEVEL 'EQ' 0
000365  01EC  E840 0000    P       LDR     6,DLVL
000366  01EE  6901 001C            BEZ     6,L24
000367                        *                       'OR' DEVICE LEVEL 'GE' CP LEVEL
000368                        *                       'THEN'
000369                        *                               'GO TO' RUPT G;
000370  01F0  E940 0000    P       CMR     6,CPLEVE
000371  01F2  0881 0018            BAGE    L24
000372                        *                       'IF' DEVICE SEMAPHORE 'NE' 0
000373                        *                       'THEN'
000374                        *                               'GO TO' RUPT N;
000375  01F4  C840 0000    P       LDR     4,DVSEM
000376  01F6  4981 00A6            BNEZ    4,L27
000377                        *                       LABEL 2 := 'LITERAL' (R1);
000378  01F8  D870 5231            LDR     5,=21041
000379  01FA  DF40 0000    P       STR     5,LABEL2
000380  01FC  D840 0000    P       LDR     $R5,DLVL
000381  01FE  C840 0000    P       LDR     $R4,CPLEVE
000382                        *                       LOAD R4 R5 (CP LEVEL, DEVICE LEVEL);
000383  0200  D3C0 0000    P       LNJ     5,JRE
000384                        *                       REPORT ERROR;
000385                        *                       PUT TEXT ("', DIDN'T RUPT - SHOULD HAVE$");
000386  0202  FBC0 FE02            LAB     7,D+5
000387  0204  9BC0 FE2C            LAB     1,T28
000388  0206  9F87                 STB     1,$B7
000389  0207  D3C0 0000    P       LNJ     5,VKPT
000390                        *                       'GO TO' FRESH START;
000391                        *       RUPT G;
000392  0209  83C0 0000    P       JMP     FRESHS
000393        020B          L24     EQU     $
000394                        *               'IF' DEVICE SEMAPHORE 'NE' 0
000395                        *               'THEN'
000396                        *               'BEGIN'
000397  020B  E840 0000    P       LDR     6,DVSEM
000398  020D  6901 0014            BEZ     6,L30
000399                        *                       LABEL 2 := 'LITERAL' (R2);
000400  020F  C870 5232            LDR     4,=21042
000401  0211  CF40 0000    P       STR     4,LABEL2
000402  0213  D840 0000    P       LDR     $R5,DLVL
000403  0215  C840 0000    P       LDR     $R4,CPLEVE
000404                        *                       LOAD R4 R5 (CP LEVEL, DEVICE LEVEL);
000405  0217  D3C0 0000    P       LNJ     5,JRE
000406                        *                       REPORT ERROR;
000407                        *                       PUT TEXT ("', RUPT WHEN SHOULDN'T$");
000408  0219  FBC0 FDEB            LAB     7,D+5
000409  021B  9BC0 FE24            LAB     1,T31
000410  021D  9F87                 STB     1,$B7
000411  021E  D3C0 0000    P       LNJ     5,VKPT
000412                        *                       'GO TO' FRESH START;
000413  0220  83C0 0000    P       JMP     FRESHS
000414                        *               'END';
000415                        **  RUPT H;
000416        0222          L30     EQU     $
000417                        *               'IF' DEVICE LEVEL 'EQ' 0
000418  0222  E840 0000    P       LDR     6,DLVL
000419  0224  6903                BEZ     6,>L32
000420                        *                       'OR' DEVICE LEVEL 'EQ' 63
000421                        *                       'THEN'
000422                        *                       'BEGIN'
000423                        **  RUPT M - INITIALIZE TO CLEAR PENDING RUPT;
000424  0225  6D3F                CMV     6,63
000425  0226  098E                BNE     >L33
000426  0227  FBC0 FDDD    L32     LAB     7,D+5
000427  0229  9BC0 0000    P       LAB     1,BINARY
000428  022B  9F87                STB     1,$B7
```

```
000429                                  *                                       INOUT (INITIALIZE,OUTPUT CONTROL @0@);
000430  022C  ABC0 0000        P               LAB     2,IOWORD
000431  022E  AFC0 FDD7                        STB     2,D+6
000432  0230  D3C0 0000        P               LNJ     5,JIO
000433                                  *                                       'GO TO' RUPT N;
000434  0232  0F81 006A                        B       L27
000435                                  *                               'END';
000436        0234                      L33     EQU     $               'CODE''BEGIN']
000437                                  *
000438  0234  9FC0 0000        P               STB     $B1,TEMPCH
000439  0236  9B80 0242                        LAB     $B1,<RUPTI
000440                                  *                                       ]CB]
000441  0238  9FC0 0005        P               STB     $B1,ISAD+4+$AF
000442  023A  9CC0 0000        P               LDB     $B1,TEMPCH
000443  023C  E870 803F                        LDR     6,=-32705
000444  023E  EF40 0000        P               STR     6,LEVTYP
000445                                  *                                       LEV ('HEX' (8000),63);(MOVE CP - LET RUPT HAPPEN)
000446  0240  8E40 0000        P               LEV     LEVTYP
000447                                  *                                       'CODE''BEGIN']
000448                                  *       RUPTI                           NOP $
000449                                  *                                       LDR $R7,=255
000450  0242  0F01 FFFF                RUPTI   NOP     $
000451  0244  F870 00FF                        LDR     $R7,=255
000452                                  *                                       ]CB]
000453                                  *                                       BDEC $R7,$
000454                                  *                                       STS  =$R7
000455  0246  7701 FFFF                        BDEC    $R7,$
000456  0248  8C57                             STS     =$R7
000457                                  *                                       ]CB]
000458  0249  FF40 FDB9                        STR     $R7,D+3
000459  024B  8800 0003        X               LBF     <ZHIAFB+3,=Z'0001'
        024D  0001
000460                                  *                                       'IF' 'BITS' @7,0@ TEMP 'NE' DEVICE LEVEL
000461                                  *                                       'THEN'
000462                                  *                                       'BEGIN'
000463  024E  E840 FDB4                        LDR     6,D+3
000464  0250  E570 007F                        AND     6,=Z'007F'
000465  0252  E940 0000        P               CMR     6,DLVL
000466  0254  0901 0014                        BE      L34
000467                                  *                                       LABEL 2:= 'LITERAL' (R3);
000468  0256  C870 5233                        LDR     4,=21043
000469  0258  CF40 0000        P               STR     4,LABEL2
000470  025A  D840 0000        P               LDR     $R5,DLVL
000471  025C  C840 FDA6                        LDR     $R4,D+3
000472                                  *                                       LOAD R4 R5 (TEMP, DEVICE LEVEL);
000473  025E  D3C0 0000        P               LNJ     5,JRE
000474                                  *                                       REPORT ERROR;
000475                                  *                                       PUT TEXT ("' PENDING RUPT WHEN CP 'NE' 63$");
000476  0260  FBC0 FDA4                        LAB     7,D+5
000477  0262  9BC0 FDE9                        LAB     1,T35
000478  0264  9F87                             STB     1,$B7
000479  0265  D3C0 0000        P               LNJ     5,VKPT
000480                                  *                                       'GO TO' FRESH START;
000481  0267  83C0 0000        P               JMP     FRESHS
000482                                  *                               'END';
000483                                  **  RUPT J;
000484        0269                      L34     EQU     $
000485                                  *                                       'IF' DEVICE LEVEL 'NE' CP LEVEL
000486                                  *                                       'THEN'
000487  0269  E840 0000        P               LDR     6,DLVL
000488  026B  E940 0000        P               CMR     6,CPLEVE
000489  026D  0901 0009                        BE      L36
000490  026F  C840 0000        P               LDR     4,CPLEVE
000491  0271  C470 8080                        OR      4,=-32640
000492  0273  CF40 0000        P               STR     4,LEVTYP
000493  0275  8E40 0000        P               LEV     LEVTYP
000494                                  *                                       LEV ('HEX'(8080),CP LEVEL); (GO TO OLD CP LEVEL)
000495                                  **  RUPT K - RESTORE POINTER TO HANDLER;
000496        0277                      L36     EQU     $
000497  0277  9FC0 0000        P               STB     $B1,TEMPCH
000498  0279  0F80              T               B       >+$A
000499  027A  83C0 0000        P               JMP     DEVIH
000500  027A                                    ORG     $-1-$AF
000501  027B                    $E              ORG     $+1
000502  027B                    $B              ORG     $-1+$AF
000503  027C                    $C              ORG     $+1
000504  027C  82C0 FFFD         $A              LB      -$E,=X'40'
        027E  0040
000505  027F  0581 0000        T               BBF     +$D
000506  0281  ABC0 FFF9                        LAB     $B2,-$C
000507  0283  9802                             LDR     $R1,$B2
000508  0284  9B92                             LAB     $B1,$B2,$R1
000509  0285  0F81 0000        T               B       +$F
000510  0287  9BC8 FFF3         $D              LAB     $B1,*-$3
000511  0289  9FC0 0005        P   $F          STB     $B1,ISAD+4+$AF
000512  028B  9CC0 0000        P               LDB     $B1,TEMPCH
000513                                  *                                       @'LOCATION' (ZHISAZ)+CP LEVEL*ZVDAF@ :=
000514  028D  9840 0000        P               LDR     1,CPLEVE
000515  028F  9B40 0000        P               MUL     1,ZV$AF
000516  0291  0F00 0000        X   N14         NOP     <ZHISAZ
000517  0293  9A40 FFFE                        ADD     1,N14+1
000518                                  *                                       'LOCATION' (ISATD @0@);
000519                                  *       RUPT N;
000520  0295  9F40 FD6F                        STR     1,D+5
000521  0297  0F00 0002        X   N15         NOP     <ISAP+2
000522  0299  E840 FFFE                        LDR     6,N15+1
000523  029B  EF48 FD69                        STR     6,*D+5
000524                                  *                               'END';  (OF DEVICE RUPT LOOP)
000525  029D  88C0 0000        P   L27         DEC     DLVL
000526  029F  0F81 FF1C                        B       L22
000527                                  *                               'END';          (OF CP RUPT  LOOP)
000528                                  ** RESET INTERRUPTS;
000529  02A1  88C0 0000        P   L23         DEC     CPLEVE
000530  02A3  0F81 FEC5                        B       L17
000531        02A5                      L18     EQU     $
000532                                  *               ZHIAFB :=0;                     (CLEAR IV FLAGS)
000533  02A5  8740 0000        P               CL      ZHIAFB
000534  02A7  9BC0 0000        P               LAB     1,ZHIAFB
000535  02A9  9FC0 FD59                        STB     1,D+3
000536                                  *               @'LOCATION' (ZHIAFB)+1@ := 0;
000537  02AB  1C01                             LDV     1,1
000538  02AC  8718 0003                        CL      *<D+3,$R1
000539  02AE  E870 8096                        LDR     6,=-32618
```

```
000540  02B0  EF40 0000      P                   STR     6,LEVTYP
000541                                *                 LEV ('HEX' (8080),22);       (RESTORE OLD LEVEL)
000542                                **  SET UP OLD HANDLERS;
000543  02B2  8E40 0000      P                   LEV     LEVTYP
000544  02B4  9840 0000      P                   LDR     1,ZV$AF
000545  02B6  1F14                               MLV     1,20
000546  02B7  0F00 0000      X       N16         NOP     <ZHISAZ
000547  02B9  9A40 FFFE                           ADD     1,N16+1
000548                                *                 @'LOCATION' (ZHISAZ)+ZVDAF*20@ := 'LOCATION' (ISACD @0@);
000549  02BB  9F40 FD47                           STR     1,D+3
000550  02BD  0F00 0002      X       N17         NOP     <ISAC+2
000551  02BF  E840 FFFE                           LDR     6,N17+1
000552  02C1  EF48 FD41                           STR     6,*D+3
000553  02C3  9840 0000      P                   LDR     1,ZV$AF
000554  02C5  1F15                               MLV     1,21
000555  02C6  0F00 0000      X       N18         NOP     <ZHISAZ
000556  02C8  9A40 FFFE                           ADD     1,N18+1
000557                                *                 @'LOCATION' (ZHISAZ)+ZVDAF*21@ := 'LOCATION' (ISARD @0@);
000558  02CA  9F40 FD38                           STR     1,D+3
000559  02CC  0F00 0002      X       N19         NOP     <ISAD+2
000560  02CE  E840 FFFE                           LDR     6,N19+1
000561  02D0  EF48 FD32                           STR     6,*D+3
000562  02D2  9840 0000      P                   LDR     1,ZV$AF
000563  02D4  1F16                               MLV     1,22
000564  02D5  0F00 0000      X       N20         NOP     <ZHISAZ
000565  02D7  9A40 FFFE                           ADD     1,N20+1
000566                                *                 @'LOCATION' (ZHISAZ)+ZVDAF*22@ := 'LOCATION' (ISATD @0@);
000567  02D9  9F40 FD29                           STR     1,D+3
000568  02DB  0F00 0002      X       N21         NOP     <ISAP+2
000569  02DD  E840 FFFE                           LDR     6,N21+1
000570  02DF  EF48 FD23                           STR     6,*D+3
000571  02E1  9FC0 0000      P                   STB     $B1,TEMPCH
000572  02E3  0F80            T                  B       >+$A
000573  02E4  83C0 0000      P                   JMP     LEV20
000574  02E4                                    ORG     $-1-$AF
000575  02E5                          $E        ORG     $+1
000576  02E5                          $B        ORG     $-1+$AF
000577  02E6                          $C        ORG     $+1
000578  02E6  82C0 FFFD               $A        LB      -$E,=X'40'
        02E8  0040
000579  02E9  0581 0000       T                  BBF     +$D
000580  02EB  ABC0 FFF9                          LAB     $B2,-$C
000581  02ED  9802                               LDR     $R1,$B2
000582  02EE  9B92                               LAB     $B1,$B2,$R1
000583  02EF  0F81 0000       T                  B       +$F
000584  02F1  9BC8 FFF3               $D          LAB     $B1,*-$B
000585  02F3  9FC0 0005      P        $F          STB     $B1,ISAC+4+$AF
000586  02F5  9CC0 0000      P                   LDB     $B1,TEMPCH
000587                                *                 ADDRESS OF LABEL (ISACP,LEV20);
000588  02F7  9FC0 0000      P                   STB     $B1,TEMPCH
000589  02F9  0F80            T                  B       >+$A
000590  02FA  83C0 0000      P                   JMP     LEV21
000591  02FA                                    ORG     $-1-$AF
000592  02FB                          $E        ORG     $+1
000593  02FB                          $B        ORG     $-1+$AF
000594  02FC                          $C        ORG     $+1
000595  02FC  82C0 FFFD               $A        LB      -$E,=X'40'
        02FE  0040
000596  02FF  0581 0000       T                  BBF     +$D
000597  0301  ABC0 FFF9                          LAB     $B2,-$C
000598  0303  9802                               LDR     $R1,$B2
000599  0304  9B92                               LAB     $B1,$B2,$R1
000600  0305  0F81 0000       T                  B       +$F
000601  0307  9BC8 FFF3               $D          LAB     $B1,*-$B
000602  0309  9FC0 0005      P        $F          STB     $B1,ISAD+4+$AF
000603  030B  9CC0 0000      P                   LDB     $B1,TEMPCH
000604                                *                 ADDRESS OF LABEL (ISARP,LEV21);
000605                                *       'END';                       (OF RUP) PROCEDURE)
000606                                *       'END';                       (OF SEGMENT)
000607                                *       'FINISH'
000608  030D  83C8 FCF2                           JMP     *D
000609        030F                    L1          EQU     $
000610  030F  83C0 0000      P                   JMP     ZK$EX
000611                                            XLOC    ZK$EX
000612                                            CTRL LINK ZK$EX
000613                                            XLOC    ZHISAZ
000614                                            XLOC    ZHRTCI
000615                                            XLOC    ZHRTCL
000616                                            XLOC    ZHRTCC
000617                                            XLOC    ZV$AF
000618                                            CTRL LINK ZV$AF
000619                                            XLOC    VKPT
000620                                            XLOC    VKPNT
000621                                            XLOC    ID
000622                                            XLOC    STATUS
000623                                            XLOC    SETTOA
000624                                            XLOC    BINARY
000625                                            XLOC    LABEL1
000626                                            XLOC    LABEL2
000627                                            XLOC    SPEED
000628                                            XLOC    DLVL
000629                                            XLOC    DVSEM
000630                                            XLOC    CPLEVE
000631                                            XLOC    LEVTYP
000632                                            XLOC    TEMPCH
000633                                            XLOC    SREG
000634                                            XLOC    ZHIAFB
000635                                            XLOC    ISAC
000636                                            XLOC    ISAD
000637                                            XLOC    ISAP
000638                                            XLOC    IOWORD
000639                                            XLOC    JRE
000640                                            XLOC    JIO
000641                                            XLOC    JSE
000642                                            XLOC    JCS
000643                                            XLOC    FRESHS
000644                                            XLOC    TIMER
000645                                            XLOC    CPRUPT
000646                                            XLOC    DEVIH
000647                                            XLOC    LEV20
000648                                            XLOC    LEV21
000649  0311  005D                                END     ZVKJLI,ZVKJLI
0000  ERR COUNT
```

@L

```
E    ZVKJLK.LIST                              05/19/78  1427.3R W 05/19/78  1422.4      3155130000
000001                                                    TITLE     ZVKJLK,'REV 00'
000002          006C                                      XDEF      KJLK
000003          0067                                      XDEF      ZVKJLK
000004          0008                              ZKCOM   COMM      8
000005  0000    0000                              D       RESV      8,0
000006  0008    0010                              T5      DC        16
000007  0009    5245 4144 2F50                            TEXT      'READ/PUNCH TEST$'
        000C    554E 4348 2054
                4553 5424
000008  0011    0013                              T7      DC        19
000009  0012    5553 494E 4720                            TEXT      'USING 400 CPM DECK'
        0015    3430 3020 4350
                4D20 4445 434B
000010  001B    2400                                      TEXT      '$'
000011  001C    001E                              T9      DC        30
000012  001D    4F4E 4C59 2043                            TEXT      'ONLY COL 1-8 CAN H'
        0020    4F4C 2031 2D38
                2043 414E 2048
000013  0026    4156 4520 5055                            TEXT      'AVE PUNCHES$'
        0029    4E43 4845 5324
000014  002C    0007                              T10     DC        7
000015  002D    544F 5441 4C20                            TEXT      'TOTAL $'
        0030    2400
000016  0031    000B                              T13     DC        11
000017  0032    5245 4144 5920                            TEXT      'READY DECKS$'
        0035    4445 434B 2400
000018  0038    000D                              T18     DC        13
000019  0039    494C 4C45 4741                            TEXT      'ILLEGAL CARDS$'
        003C    4C20 4341 5244
                2400
000020  0040    000E                              T41     DC        14
000021  0041    5245 4144 2F50                            TEXT      'READ/PUNCHED $'
        0044    554E 4348 4544
                2024
000022  0048    0022                              T44     DC        34
000023  0049    5354 4154 5553                            TEXT      'STATUS ERROR CARDS'
        004C    2045 5252 4F52
                2043 4152 4453
000024  0052    204F 4646 5345                            TEXT      ' OFFSET STACKED$'
        0055    5420 5354 4143
                4B45 4424
000025  005A    000E                              T47     DC        14
000026  005B    2C20 4552 524F                            TEXT      ', ERROR EJECT$'
        005E    5220 454A 4543
                5424
000027  0062    0007                              T49     DC        7
000028  0063    5645 5249 4659                            TEXT      'VERIFY$'
        0066    2400
000029          0067                              ZVKJLK  EQU       $
000030  0067    8F00 0001       K                         SAVE      <ZKCOM+1,Z'0011'
        0069    0011
000031                                                    *       'BEGIN'
000032  006A    0F81 01CD                                 B         L1
000033                                                    *           'INTEGER' K,J;
000034  006C    DFC0 FF93                         KJLK    STB       5,D
000035                                                    *           'IF' ID 'NE' 'HEX' (208A)
000036                                                    *               'THEN'
000037                                                    *                   'GO TO' END LETTER K;
000038  006E    E840 0000       P                         LDR       6,ID
000039  0070    E970 208A                                 CMR       6,=8330
000040  0072    0981 01C3                                 BNE       L50
000041                                                    *           PUT NEW TEXT ("READ/PUNCH TEST$");
000042  0074    FBC0 FF8E                                 LAB       7,D+3
000043  0076    9BC0 FF91                                 LAB       1,T5
000044  0078    9F87                                      STB       1,$B7
000045  0079    D3C0 0000       P                         LNJ       5,VKPNT
000046                                                    *           LABEL 1 := 'LITERAL' (K1);
000047  007B    E870 4B31                                 LDR       6,=19249
000048  007D    EF40 0000       P                         STR       6,LABEL1
000049                                                    *           R OR V MODE := 'LITERAL' (R);
000050  007F    4C52                                      LDV       4,82
000051  0080    CF40 0000       P                         STR       4,RORVMD
000052                                                    *           READ OR PUNCH := PUNCH @0@;
000053  0082    D800 0010       X                         LDR       5,<IOWORD+16
000054  0084    DF40 0000       P                         STR       5,READOR
000055  0086    FBC0 FF7C                                 LAB       7,D+3
000056  0088    9BC0 0000       P                         LAB       1,BINARY
000057  008A    9F87                                      STB       1,$B7
000058                                                    *           INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000059  008B    ABC0 0000       P                         LAB       2,IOWORD
000060  008D    AFC0 FF76                                 STB       2,D+4
000061  008F    D3C0 0000       P                         LNJ       5,JIO
000062                                                    *           K := SET TO HOLLERITH 'UNION' 'HEX' (800);
000063  0091    E840 0000       P                         LDR       6,SETTOH
000064  0093    E470 0800                                 OR        6,=2048
000065  0095    EF40 FF6B                                 STR       6,D+1
000066  0097    FBC0 FF6B                                 LAB       7,D+3
000067  0099    9BC0 FF67                                 LAB       1,D+1
000068  009B    9F87                                      STB       1,$B7
000069                                                    *           INOUT (K, OUTPUT TASK @0@);
000070  009C    0F00 0005       X                 N1      NOP       <IOWORD+5
000071  009E    C840 FFFE                                 LDR       4,N1+1
000072  00A0    CF40 FF63                                 STR       4,D+4
000073  00A2    D3C0 0000       P                         LNJ       5,JIO
000074                                                    *           STATUS SHOULD BE := 'HEX'(8000);
000075  00A4    E870 8000                                 LDR       6,=-32768
000076  00A6    EF40 0000       P                         STR       6,STATUS
000077                                                    *           ZHRTCC := 0;
000078  00A8    8740 0000       P                         CL        ZHRTCC
000079                                                    *           ZHRTCI := 0;
000080  00AA    8740 0000       P                         CL        ZHRTCI
000081                                                    *           LOW ADDRESS := ZVDLR;
000082                                                    *       TRY AGAIN;
000083  00AC    C840 0000       P                         LDR       4,ZV$LR
000084  00AE    CF40 0000       P                         STR       4,LOWADD
000085                                                    **  CLEAR BUFFER  1 AND 2;
000086          00B0                              L6      EQU       $
000087                                                    *           ASK NEW QUESTION ("USING 400 CPM DECK$");
000088  00B0    FBC0 FF52                                 LAB       7,D+3
000089  00B2    9BC0 FF5E                                 LAB       1,T7
000090  00B4    9F87                                      STB       1,$B7
000091  00B5    D3C0 0000       P                         LNJ       5,VKANQ
```

```
000092                                  *           K := 'HEX' (590D);
000093  00B7  E870 590D                         LDR     6,=22797
000094  00B9  EF40 FF47                         STR     6,D+1
000095                                  *       GET CHAR (K);
000096  00BB  FBC0 FF47                         LAB     7,D+3
000097  00BD  9BC0 FF43                         LAB     1,D+1
000098  00BF  9F87                              STB     1,$B7
000099  00C0  D3C0 0000         P               LNJ     5,VKGC
000100                                  *       'IF' K 'EQ' 'LITERAL' (N)
000101                                  *       'THEN'
000102                                  *   'BEGIN'
000103  00C2  E840 FF3E                         LDR     6,D+1
000104  00C4  6D4E                              CMV     6,78
000105  00C5  0997                              BNE     >L8
000106                                  *       PUT NEW TEXT ("ONLY COL 1-8 CAN HAVE PUNCHES$");
000107  00C6  FBC0 FF3C                         LAB     7,D+3
000108  00C8  9BC0 FF53                         LAB     1,T9
000109  00CA  9F87                              STB     1,$B7
000110  00CB  D3C0 0000         P               LNJ     5,VKPNT
000111                                  *       ASK NEW QUESTION ("TOTAL $");
000112  00CD  FBC0 FF35                         LAB     7,D+3
000113  00CF  9BC0 FF5C                         LAB     1,T10
000114  00D1  9F87                              STB     1,$B7
000115  00D2  D3C0 0000         P               LNJ     5,VKANQ
000116                                  *       GET DECIMAL (CARDS SPECIFIED);
000117                                  *   'END'
000118  00D4  FBC0 FF2E                         LAB     7,D+3
000119  00D6  9BC0 0000         P               LAB     1,CARDSS
000120  00D8  9F87                              STB     1,$B7
000121  00D9  D3C0 0000         P               LNJ     5,VKGD
000122                                  *   'ELSE'
000123                                  *   'BEGIN'
000124  00DB  0F85                              B       >L11
000125        00DC                      L8      EQU     $
000126                                  *       'IF' K 'NE' 'LITERAL' (Y)
000127                                  *       'THEN'
000128                                  *           'GO TO' TRY AGAIN;
000129  00DC  E840 FF24                         LDR     6,D+1
000130  00DE  6D59                              CMV     6,89
000131  00DF  09D1                              BNE     >L6
000132                                  *   'END';
000133        00E0                      L11     EQU     $
000134                                  *       PUT NEW TEXT ("READY DECK$");
000135  00E0  FBC0 FF22                         LAB     7,D+3
000136  00E2  9BC0 FF4E                         LAB     1,T13
000137  00E4  9F87                              STB     1,$B7
000138  00E5  D3C0 0000         P               LNJ     5,VKPNT
000139  00E7  D3C0 0000         P               LNJ     5,JWR
000140                                  *       WAIT FOR RETURN;
000141                                  *       CHECK STATE (SHOULD BE ON);
000142  00E9  FBC0 FF19                         LAB     7,D+3
000143  00EB  6C01                              LDV     6,1
000144  00EC  EF07                              STR     6,$B7
000145  00ED  D3C0 0000         P               LNJ     5,JCS
000146                                  *       LABEL 1:= 'LITERAL' (K2);
000147  00EF  E870 4B32                         LDR     6,=19250
000148  00F1  EF40 0000         P               STR     6,LABEL1
000149  00F3  FBC0 FF0F                         LAB     7,D+3
000150  00F5  9BC0 0000         P               LAB     1,BUFF1
000151  00F7  9F87                              STB     1,$B7
000152  00F8  0F00 0007         X       N2      NOP     <IOWORD+7
000153  00FA  C840 FFFE                         LDR     4,N2+1
000154  00FC  CF40 FF07                         STR     4,D+4
000155                                  *       IOLD  (BUFFER1 @0@,READ @0@,171);
000156  00FE  DB70 00AB                         LDR     5,=171
000157  0100  DF40 FF04                         STR     5,D+5
000158  0102  D3C0 0000         P               LNJ     5,JIL
000159                                  *       CHECK STATE (SHOULD BE ON);
000160  0104  FBC0 FEFE                         LAB     7,D+3
000161  0106  6C01                              LDV     6,1
000162  0107  EF07                              STR     6,$B7
000163  0108  D3C0 0000         P               LNJ     5,JCS
000164  010A  D3C0 0000         P               LNJ     5,JSE
000165                                  *       CHECK STATUS FOR ERRORS;
000166  010C  6C08                      L14     LDV     6,8
000167  010D  EF40 FEF3                         STR     6,D+1
000168                                  *   'FOR' K:=8 'STEP' 1 'UNTIL' 79 'DO'
000169  010F  E840 FEF1                 L15     LDR     6,D+1
000170  0111  6EB1                              ADV     6,-79
000171  0112  6A12                              BGZ     6,>L16
000172                                  *           'IF' BUFFER 1 @K@ 'NE' 0
000173                                  *           'THEN'
000174                                  *           'BEGIN'
000175  0113  9840 FEED                         LDR     1,D+1
000176  0115  E810 0000         X               LDR     6,<BUFF1.$R1
000177  0117  690A                              BEZ     6,>L17
000178  0118  FBC0 FEEA                         LAB     7,D+3
000179  011A  9BC0 FF1D                         LAB     1,T18
000180  011C  9F87                              STB     1,$B7
000181  011D  D3C0 0000         P               LNJ     5,VKPNT
000182                                  *                   PUT NEW TEXT ("ILLEGAL CARD$");
000183                                  *                   'GO TO' TRY AGAIN;
000184  011F  0F81 FF90                         B       L6
000185                                  *           'END';
000186  0121  8AC0 FEDF                 L17     INC     D+1
000187  0123  0FEC                              B       >L15
000188        0124                      L16     EQU     $
000189  0124  6C01                      L19     LDV     6,1
000190  0125  EF40 FEDC                         STR     6,D+2
000191                                  *   'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
000192  0127  E840 FEDA                 L20     LDR     6,D+2
000193  0129  6EF8                              ADV     6,-8
000194  012A  6A19                              BGZ     6,>L21
000195  012B  8740 FED5                 L22     CL      D+1
000196                                  *   'FOR' K:=0 'STEP' 1 'UNTIL' 7 'DO'
000197  012D  E840 FED3                 L23     LDR     6,D+1
000198  012F  6EF9                              ADV     6,-7
000199  0130  6A10                              BGZ     6,>L24
000200  0131  E840 FED0                         LDR     6,D+2
000201  0133  6003                              SDL     6,3
000202  0134  EA40 FECC                         ADD     6,D+1
000203                                  *       BUFFER @J*8+K@ := BUFFER @K@;
000204  0136  9840 FECA                         LDR     1,D+1
```

```
000205  0138  A856                                LDR     2,=$R6
000206  0139  C810 0000      X                   LDR     4,<BUFF1.$R1
000207  013B  CF20 0000      X                   STR     4,<BUFF1.$R2
000208  013D  8AC0 FEC3                          INC     D+1
000209  013F  0FEE                              B       >L23
000210  0140  8AC0 FEC1              L24         INC     D+2
000211  0142  0FE5                              B       >L20
000212        0143                    L21       EQU     $
000213                              *               ERROR FLAG := 0;
000214  0143  8740 0000_     P                  CL      ERRORF
000215  0145  0004                              DC      4
000216                              *               CLOCK ON;
000217  0146  6C01                    L25       LDV     6,1
000218  0147  EF40 0000      P                  STR     6,CARDSR
000219                              *               'FOR' CARD COUNT := 1 'STEP' 1 'UNTIL' CARDS SPECIFIED-1 'DO'
000220                              *               'BEGIN'
000221  0149  E240 0000      P                  SUB     6,CARDSS
000222  014B  8256                              NEG     =$R6
000223  014C  EF40 FEB6                          STR     6,D+3
000224  014E  E840 0000      P     L26          LDR     6,CARDSR
000225  0150  E240 FEB2                          SUB     6,D+3
000226  0152  6A01 008A                          BGZ     6,L27
000227                              *               'IF' CARD COUNT 'EQ' 1
000228                              *               'THEN'
000229                              *               'BEGIN'
000230  0154  E840 0000      P                  LDR     6,CARDSR
000231  0156  6D01                              CMV     6,1
000232  0157  0998                              BNE     >L28
000233  0158  FBC0 FEAB                          LAB     7,D+4
000234  015A  9BC0 0000      P                  LAB     1,BUFF1
000235  015C  9F87                              STB     1,$B7
000236  015D  0F00 0010      X     N3           NOP     <IOWORD+16
000237  015F  C840 FFFE                          LDR     4,N3+1
000238  0161  CF40 FEA3                          STR     4,D+5
000239                              *                   IOLD (BUFFER 1 @0@, PUNCH @0@,171);
000240  0163  D870 00AB                          LDR     5,=171
000241  0165  DF40 FEA0                          STR     5,D+6
000242  0167  D3C0 0000      P                  LNJ     5,JIL
000243                              *                   READ OR PUNCH := PUNCH @0@;
000244  0169  E800 0010      X                  LDR     6,<IOWORD+16
000245  016B  EF40 0000      P                  STR     6,READOR
000246  016D  D3C0 0000      P                  LNJ     5,JSE
000247                              *                   CHECK STATUS FOR ERRORS;
000248                              *               'END';
000249        016F                    L28       EQU     $
000250                              *               'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000251                              *               'THEN'
000252  016F  E840 0000      P                  LDR     6,CARDSR
000253  0171  6B13                              BEVN    6,>L29
000254  0172  FBC0 FE91                          LAB     7,D+4
000255  0174  9BC0 0000      P                  LAB     1,BUFF2
000256  0176  9F87                              STB     1,$B7
000257  0177  0F00 0007      X     N4           NOP     <IOWORD+7
000258  0179  C840 FFFE                          LDR     4,N4+1
000259  017B  CF40 FE89                          STR     4,D+5
000260                              *                   IOLD (BUFFER 2 @0@,READ@0@,171)
000261                              *               'ELSE'
000262  017D  D870 00AB                          LDR     5,=171
000263  017F  DF40 FE86                          STR     5,D+6
000264  0181  D3C0 0000      P                  LNJ     5,JIL
000265  0183  0F92                              B       >L30
000266  0184  FBC0 FE7F              L29         LAB     7,D+4
000267  0186  9BC0 0000      P                  LAB     1,BUFF1
000268  0188  9F87                              STB     1,$B7
000269  0189  0F00 0007      X     N5           NOP     <IOWORD+7
000270  018B  E840 FFFE                          LDR     6,N5+1
000271  018D  EF40 FE77                          STR     6,D+5
000272  018F  C870 00AB                          LDR     4,=171
000273  0191  CF40 FE74                          STR     4,D+6
000274  0193  D3C0 0000      P                  LNJ     5,JIL
000275                              *                   IOLD (BUFFER 1 @0@,READ@0@,171);
000276        0195                    L30       EQU     $
000277                              *                   READ OR PUNCH := READ @0@;
000278  0195  E800 0007      X                  LDR     6,<IOWORD+7
000279  0197  EF40 0000      P                  STR     6,READOR
000280  0199  D3C0 0000      P                  LNJ     5,JSE
000281                              *                   CHECK STATUS FOR ERRORS;
000282  019B  6C01                    L31       LDV     6,1
000283  019C  EF40 FE65                          STR     6,D+2
000284                              *               'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
000285  019E  E840 FE63              L32         LDR     6,D+2
000286  01A0  6EF8                              ADV     6,-8
000287  01A1  6A29                              BGZ     6,>L33
000288  01A2  8740 FE5E              L34         CL      D+1
000289                              *               'FOR' K:=0 'STEP' 1 'UNTIL' 7      'DO'
000290  01A4  E840 FE5C              L35         LDR     6,D+1
000291  01A6  6EF9                              ADV     6,-7
000292  01A7  6A20                              BGZ     6,>L36
000293                              *                   'IF' 'BITS' @1,0@ CARD COUNT 'EQ' 1
000294                              *                   'THEN'
000295  01A8  E840 0000      P                  LDR     6,CARDSR
000296  01AA  6B0E                              BEVN    6,>L37
000297  01AB  C840 FE56                          LDR     4,D+2
000298  01AD  4003                              SOL     4,3
000299  01AE  CA40 FE52                          ADD     4,D+1
000300                              *                       BUFFER 2 @J*8+K@ := BUFFER 2 @K@
000301                              *                   'ELSE'
000302  01B0  9840 FE50                          LDR     1,D+1
000303  01B2  A854                              LDR     2,=$R4
000304  01B3  D810 0000      X                  LDR     5,<BUFF2.$R1
000305  01B5  DF20 0000      X                  STR     5,<BUFF2.$R2
000306  01B7  0F8D                              B       >L38
000307  01B8  E840 FE49              L37         LDR     6,D+2
000308  01BA  6003                              SOL     6,3
000309  01BB  EA40 FE45                          ADD     6,D+1
000310                              *                       BUFFER 1 @J*8+K@ := BUFFER 1 @K@;
000311  01BD  9840 FE43                          LDR     1,D+1
000312  01BF  A856                              LDR     2,=$R6
000313  01C0  C810 0000      X                  LDR     4,<BUFF1.$R1
000314  01C2  CF20 0000      X                  STR     4,<BUFF1.$R2
000315  01C4  8AC0 FE3C              L38         INC     D+1
000316  01C6  0FDE                              B       >L35
000317  01C7  8AC0 FE3A              L36         INC     D+2
```

```
000318  01C9  0FD5                        B       >L32
000319        01CA                  L33   EQU     $
000320                              *               READ OR PUNCH := PUNCH @0@;
000321  01CA  E800 0010       X           LDR     6,<IOWORD+16
000322  01CC  EF40 0000       P           STR     6,READOR
000323                              *               GENERATE CHECK SUMS (PUNCH @0@);
000324  01CE  FBC0 FE35                   LAB     7,D+4
000325  01D0  EF07                        STR     6,$B7
000326  01D1  D3C0 0000       P           LNJ     5,JGC
000327  01D3  D3C0 0000       P           LNJ     5,JSE
000328                              *               CHECK STATUS FOR ERRORS;
000329  01D5  D3C0 0000       P           LNJ     5,VKGB
000330                              *               'IF' GET BREAK 'EQ' TRUE
000331                              *               'THEN'
000332                              *                   'GO TO' STOP PUNCHING;
000333  01D7  6D01                        CMV     6,1
000334  01D8  0905                        BE      >L27
000335                              *           'END';
000336                              *       STOP PUNCHING;
000337  01D9  8AC0 0000       P           INC     CARDSR
000338  01DB  0F81 FF72                   B       L26
000339        01DD                  L27   EQU     $
000340  01DD  0005                        DC      5
000341                              *           CLOCK OFF;
000342                              *           READ OR PUNCH := FALSE;
000343  01DE  8740 0000       P           CL      READOR
000344                              *           GENERATE CHECKSUMS(READ OR PUNCH);
000345  01E0  FBC0 FE22                   LAB     7,D+3
000346  01E2  E840 0000       P           LDR     6,READOR
000347  01E4  EF07                        STR     6,$B7
000348  01E5  D3C0 0000       P           LNJ     5,JGC
000349  01E7  FBC0 FE1B                   LAB     7,D+3
000350  01E9  9BC0 FE56                   LAB     1,T41
000351  01EB  9F87                        STB     1,$B7
000352  01EC  D3C0 0000       P           LNJ     5,VKPNT
000353                              *           PUT NEW TEXT ("READ/PUNCHED $");
000354                              *           CALCULATE ELAPSED TIME (CARD COUNT);
000355  01EE  FBC0 FE14                   LAB     7,D+3
000356  01F0  E840 0000       P           LDR     6,CARDSR
000357  01F2  EF07                        STR     6,$B7
000358  01F3  D3C0 0000       P           LNJ     5,JET
000359                              *           'IF' ERROR FLAG 'NE' 0
000360                              *           'THEN'
000361                              *           'BEGIN'
000362  01F5  E840 0000       P           LDR     6,ERRORF
000363  01F7  6922                        BEZ     6,>L42
000364                              *               'INTEGER' TEMP;
000365                              *       WRONG RESPONSE:     ASK NEW QUESTION
000366                              *
000367  01F8  FBC0 FE0B               L43   LAB     7,D+4
000368  01FA  9BC0 FE4D                     LAB     1,T44
000369  01FC  9F87                          STB     1,$B7
000370  01FD  D3C0 0000       P             LNJ     5,VKANQ
000371                              *                   ("STATUS ERROR CARDS OFFSET STACKED$");
000372                              *                   GET CHAR (TEMP);
000373  01FF  FBC0 FE04                     LAB     7,D+4
000374  0201  9BC0 FE01                     LAB     1,D+3
000375  0203  9F87                          STB     1,$B7
000376  0204  D3C0 0000       P             LNJ     5,VKGC
000377                              *                   'IF' TEMP 'NE' 'LITERAL' (Y)
000378                              *                   'THEN'
000379                              *                   'BEGIN'
000380  0206  E840 FDFC                     LDR     6,D+3
000381  0208  6D59                          CMV     6,89
000382  0209  0910                          BE      >L42
000383                              *                           'IF' TEMP 'NE' 'LITERAL' (N)
000384                              *                           'THEN'
000385                              *                               'GO TO' WRONG RESPONSE;
000386  020A  6D4E                          CMV     6,78
000387  020B  09ED                          BNE     >L43
000388                              *                           LABEL 2 := 'LITERAL' (OS);
000389  020C  C870 4F53                     LDR     4,=20307
000390  020E  CF40 0000       P             STR     4,LABEL2
000391  0210  D3C0 0000       P             LNJ     5,JRE
000392                              *                           REPORT ERROR;
000393  0212  FBC0 FDF1                     LAB     7,D+4
000394  0214  9BC0 FE45                     LAB     1,T47
000395  0216  9F87                          STB     1,$B7
000396  0217  D3C0 0000       P             LNJ     5,VKPT
000397                              *                           PUT TEXT (", ERROR EJECT$");
000398                              *                   'END';
000399                              *               'END';
000400        0219                  L42   EQU     $
000401                              *           CARDS SPECIFIED := CARD COUNT;
000402                              *       VERIFY;
000403  0219  E840 0000       P           LDR     6,CARDSR
000404  021B  EF40 0000       P           STR     6,CARDSS
000405        021D                  L48   EQU     $
000406                              *           ROR V MODE := 'LITERAL' (V);
000407  021D  6C56                        LDV     6,86
000408  021E  EF40 0000       P           STR     6,RORVMD
000409                              *           ASK NEW QUESTION ("VERIFY$");
000410  0220  FBC0 FDE2                   LAB     7,D+3
000411  0222  9BC0 FE3F                   LAB     1,T49
000412  0224  9F87                        STB     1,$B7
000413  0225  D3C0 0000       P           LNJ     5,VKANQ
000414                              *           GET CHAR (K);
000415  0227  FBC0 FDDB                   LAB     7,D+3
000416  0229  9BC0 FDD7                   LAB     1,D+1
000417  022B  9F87                        STB     1,$B7
000418  022C  D3C0 0000       P           LNJ     5,VKGC
000419                              *           'IF' K 'NE' 'LITERAL'(N)
000420                              *           'THEN'
000421                              *           'BEGIN'
000422  022E  E840 FDD2                   LDR     6,D+1
000423  0230  6D4E                        CMV     6,78
000424  0231  0905                        BE      >L50
000425                              *               'IF' K 'NE' 'LITERAL' (Y)
000426                              *               'THEN'
000427                              *                   'GO TO' VERIFY;
000428  0232  6D59                        CMV     6,89
000429  0233  09EA                        BNE     >L48
000430                              *               LETTER RP;
```

```
000431  0234  D3C0 0000      P               LNJ     5,KJLR
000432                        *       'END';
000433                        *     END LETTER K;
000434                        *     'END';    (OF PROCEDURE)
000435                        *     'END';    (OF SEGMENT)
000436                        *     'FINISH'
000437  0236  83C8 FDC9              L50       JMP     *D
000438        0238                    L1        EQU     $
000439  0238  83C0 0000      P               JMP     ZK$EX
000440                                        XLOC    ZK$EX
000441                                        CTRL LINK ZK$EX
000442                                        XLOC    ZV$LR
000443                                        CTRL LINK ZV$LR
000444                                        XLOC    ZHRTC1
000445                                        XLOC    ZHRTCC
000446                                        XLOC    VKPT
000447                                        XLOC    VKPNT
000448                                        XLOC    VKANQ
000449                                        XLOC    VKGC
000450                                        XLOC    VKGD
000451                                        XLOC    VKGB
000452                                        XLOC    ID
000453                                        XLOC    STATUS
000454                                        XLOC    SETTOH
000455                                        XLOC    BINARY
000456                                        XLOC    LABEL1
000457                                        XLOC    LABEL2
000458                                        XLOC    CARDSS
000459                                        XLOC    CARDSR
000460                                        XLOC    LOWADD
000461                                        XLOC    RORVMQ
000462                                        XLOC    ERRORF
000463                                        XLOC    READOR
000464                                        XLOC    BUFF1
000465                                        XLOC    BUFF2
000466                                        XLOC    IOWORD
000467                                        XLOC    JGC
000468                                        XLOC    JRE
000469                                        XLOC    JIO
000470                                        XLOC    JIL
000471                                        XLOC    JWR
000472                                        XLOC    JSE
000473                                        XLOC    JCS
000474                                        XLOC    JET
000475                                        XLOC    KJLR
000476  023A  0067                           END     ZVKJLK,ZVKJLK
0000  ERR COUNT
eeL
```

```
E   ZVKJLL.LIST                          05/19/78  1427.3R W 05/19/78  1422.7      53520300O0
000001                                              TITLE     ZVKJLL,'REV 00'
000002          014D                                XDEF      KJLL
000003          0065                                XDEF      ZVKJLL
000004          0008                    ZKCOM       COMM      8
000005  0000    0000                    D           RESV      16,0
000006  0010    0017                    T9          DC        23
000007  0011    2C20 5752 4F4E                      TEXT      ', WRONG ''INVALID'' '
        0014    4720 2749 4E56
                414C 4944 2720
000008  001A    434F 4E2E 2400                      TEXT      'CON.$'
000009  001D    0015                    T22         DC        21
000010  001E    2C20 5752 4F4E                      TEXT      ', WRONG ''VALID'' CO'
        0021    4720 2756 414C
                4944 2720 434F
000011  0027    4445 2400                           TEXT      'DE$'
000012  0029    000A                    T24         DC        10
000013  002A    5445 5354 204D                      TEXT      'TEST MODE$'
        002D    4F44 4524
000014  002F    0010                    T30         DC        16
000015  0030    2C20 434F 554E                      TEXT      ', COUNTER WRONG$'
        0033    5445 5220 5752
                4F4E 4724
000016  0038    0010                    T39         DC        16
000017  0039    2C20 4441 5441                      TEXT      ', DATA LOOPBACK$'
        003C    204C 4F4F 5042
                4143 4B24
000018  0041    001F                    T53         DC        31
000019  0042    2C20 4A41 4D20                      TEXT      ', JAM ONE SHOT DID'
        0045    4F4E 4520 5348
                4F54 2044 4944
000020  004B    4E27 5420 5449                      TEXT      'N''T TIME-OUT$'
        004E    4D45 2D4F 5554
                2400
000021  0052    001C                    T56         DC        28
000022  0053    2C20 4A41 4D20                      TEXT      ', JAM ONE SHOT TIM'
        0056    4F4E 4520 5348
                4F54 2054 494D
000023  005C    4544 204F 5554                      TEXT      'ED OUT IN$'
        005F    2049 4E24
000024  0061    0006                    T57         DC        6
000025  0062    204D 5345 4324                      TEXT      ' MSEC$'
000026          0065                    ZVKJLL      EQU       $
000027  0065    8F00 0001       K                   SAVE      <ZKCOM+1,Z'0011'
        0067    0011
000028                                  *       'PROCEDURE' CONVERT BINARY TO ASCII ('VALUE''INTEGER' I ;
000029  0068    0F81 0374                           B         L1
000030                                  *                                          'VALUE' 'INTEGER' ASCII WORD);
000031                                  *       'BEGIN'
000032                                  *           'INTEGER' X, PARTWORD,K;
000033  006A    DFC0 FF95               P2          STB       5,D
000034  006C    8C87                                LDI       $87
000035  006D    8D40 FF93                           SDI       D+1
000036                                  *               'IF' 'BITS' @7,2@ I 'NE' 0
000037                                  *                   'THEN'
000038                                  *                   'BEGIN'
000039  006F    82D6                                LB        =$R6,Z'01FC'
        0070    01FC
000040  0071    0581 007C                           BBF       L3
000041                                  *                   X:= 0;
000042  0073    8740 FF8F                           CL        D+3
000043                                  *                   PARTWORD := 'BITS' @7,2@ I;
000044  0075    6007                                SOL       6,7
000045  0076    6049                                SOR       6,9
000046  0077    EF40 FF8C                           STR       6,D+4
000047          0079                    L4          EQU       $
000048                                  *                   'FOR' K:=0 'STEP' 1 'UNTIL' 6
000049  0079    8740 FF8B                           CL        D+5
000050                                  *                   'DO'
000051                                  *                   'BEGIN'
000052  007B    E840 FF89               L5          LDR       6,D+5
000053  007D    6EFA                                ADV       6,-6
000054  007E    6A12                                BGZ       6,>L6
000055                                  *                               X:= 'BITS' @1,0@ PARTWORD + X;
000056  007F    E840 FF84                           LDR       6,D+4
000057  0081    E570 0001                           AND       6,=Z'0001'
000058  0083    EA40 FF7F                           ADD       6,D+3
000059  0085    EF40 FF7D                           STR       6,D+3
000060                                  *                               PARTWORD := 'BITS' @7,1@ PARTWORD;
000061  0087    C840 FF7C                           LDR       4,D+4
000062  0089    C2D4                                LLH       4,=$R4
000063  008A    4041                                SOR       4,1
000064  008B    CF40 FF78                           STR       4,D+4
000065                                  *                   'END';
000066  008D    8AC0 FF77                           INC       D+5
000067  008F    0FEC                                B         >L5
000068                                  *               'IF' X 'NE' 1
000069          0090                    L6          EQU       $
000070                                  *               'THEN'
000071                                  *               'BEGIN'
000072  0090    E840 FF72                           LDR       6,D+3
000073  0092    6D01                                CMV       6,1
000074  0093    0901 002D                           BE        L7
000075                                  *                           'IF' 'BITS' @8,0@ ASCII WORD 'NE' 'HEX' (FF)
000076                                  *                           'THEN'
000077                                  *                           'BEGIN'
000078  0095    C840 FF6C                           LDR       4,D+2
000079  0097    C2D4                                LLH       4,=$R4
000080  0098    C970 00FF                           CMR       4,=255
000081  009A    0901 00B0                           BE        L20
000082                                  *                                   LABEL2 := 'LITERAL' (IC);
000083  009C    D870 4943                           LDR       5,=18755
000084  009E    DF40 0000       P                   STR       5,LABEL2
000085                                  *                                   K := 'HEX' (FF);
000086  00A0    F870 00FF                           LDR       7,=255
000087  00A2    FF40 FF62                           STR       7,D+5
000088  00A4    D840 FF5D                           LDR       $R5,D+2
000089  00A6    C840 FF5E                           LDR       $R4,D+5
000090                                  *                                   LOAD R4 R5 (K, ASCII WORD );
000091  00A8    D3C0 0000       P                   LNJ       5,JRE
000092                                  *                                   REPORT ERROR;
000093                                  *                                  PUT TEXT(", WRONG 'INVALID' CON.$");
000094  00AA    FBC0 FF5B                           LAB       7,D+6
```

```
000095  00AC  9BC0 FF63                    LAB     1,T9
000096  00AE  9F87                         STB     1,$B7
000097  00AF  D3C0 0000      P              LNJ     5,VKPT
000098  00B1  FBC0 FF54                    LAB     7,D+6
000099  00B3  E840 FF4E                    LDR     6,D+2
000100  00B5  EF07                         STR     6,$B7
000101  00B6  D3C0 0000      P              LNJ     5,VKPH
000102                              *                       PUT HEX (ASCII WORD);
000103                              *                      PUT HEX (1);
000104  00B8  FBC0 FF4D                    LAB     7,D+6
000105  00BA  E840 FF46                    LDR     6,D+1
000106  00BC  EF07                         STR     6,$B7
000107  00BD  D3C0 0000      P              LNJ     5,VKPH
000108                              *                        'GO TO' ASK NEXT;
000109  00BF  83C0 0000      P              JMP     ASKNEX
000110                              *                'END';
000111                              *                     'GO TO' END CBA;
000112                              *                'END';
000113        00C1                  L7      EQU     $
000114                              *                PARTWORD := 'BITS' @8,1@ I;
000115  00C1  E840 FF3F                    LDR     6,D+1
000116  00C3  6007                         SOL     6,7
000117  00C4  6048                         SOR     6,8
000118  00C5  EF40 FF3E                    STR     6,D+4
000119  00C7  8740 FF3D             L12    CL      D+5
000120  00C9  D3C0 000A                    LNJ     5,P13
000121  00CB  8AC0 FF39             L14    INC     D+5
000122                              *                'FOR' K:=0, K+1 'WHILE' PARTWORD 'NE' 1
000123                              *                'DO'
000124  00CD  E840 FF36                    LDR     6,D+4
000125  00CF  6D01                         CMV     6,1
000126  00D0  090D                         BE      >L15
000127  00D1  D3C0 0002                    LNJ     5,P13
000128  00D3  0FF8                         B       >L14
000129  00D4  DFC0 FF31             P13    STB     5,D+6
000130                              *                          PARTWORD := PARTWORD/2;
000131  00D6  E840 FF2D                    LDR     6,D+4
000132  00D8  6061                         SAR     6,1
000133  00D9  EF40 FF2A                    STR     6,D+4
000134  00DB  83C8 FF2A                    JMP     *D+6
000135  00DD  E840 FF27             L15    LDR     6,D+5
000136  00DF  6002                         SOL     6,2
000137  00E0  C2C0 FF20                    LLH     4,D+1
000138  00E2  C570 000E                    AND     4,=Z'000E'
000139  00E4  4004                         SOL     4,4
000140  00E5  CA56                         ADD     4,=$R6
000141  00E6  D840 FF1A                    LDR     5,D+1
000142  00E8  D570 0003                    AND     5,=Z'0003'
000143  00EA  CA55                         ADD     4,=$R5
000144  00EB  CF40 FF17                    STR     4,D+3
000145                              *                X := ('BITS' @3,9@ I)*32 +K*4 +'BITS' @2,0@ I;
000146                              *            'END'
000147                              *          'ELSE'
000148  00ED  0F8D                         B       >L16
000149        00EE                  L3      EQU     $
000150                              *                X := 'BITS' @3,9@I*32 + 'BITS' @2,0@ I;
000151  00EE  E2C0 FF12                    LLH     6,D+1
000152  00F0  E570 000E                    AND     6,=Z'000E'
000153  00F2  6004                         SOL     6,4
000154  00F3  C840 FF0D                    LDR     4,D+1
000155  00F5  C570 0003                    AND     4,=Z'0003'
000156  00F7  EA54                         ADD     6,=$R4
000157  00F8  EF40 FF0A                    STR     6,D+3
000158        00FA                  L16     EQU     $
000159                              *            'IF' 'BITS' @1,0@X'EQ' 0
000160                              *            'THEN'
000161                              *            'BEGIN'
000162  00FA  E840 FF08                    LDR     6,D+3
000163  00FC  6B8D                         BODD    6,>L17
000164  00FD  6061                         SAR     6,1
000165                              *                'IF' 'BITS' @8,8@ WORD CONVERSION @X/2@ 'NE'
000166                              *                     'BITS' @8,0@ ASCII WORD
000167                              *                'THEN'
000168                              *                     'GO TO' VALID BUT IN ERROR;
000169                              *            'END'
000170  00FE  C840 FF03                    LDR     4,D+2
000171  0100  C2D4                         LLH     4,=$R4
000172  0101  9856                         LDR     1,=$R6
000173  0102  D810 0000      X              LDR     5,<WORDCO.$R1
000174  0104  5048                         SOR     5,8
000175  0105  C955                         CMR     4,=$R5
000176  0106  0990                         BNE     >L19
000177                              *            'ELSE'
000178                              *            'BEGIN'
000179  0107  0F81 0043                    B       L20
000180  0109  E840 FEF9      *       L17    LDR     6,D+3
000181  010B  6061                         SAR     6,1
000182                              *                'IF' 'BITS' @8,0@ WORD CONVERSION @X/2@ 'NE'
000183                              *                     'BITS' @8,0@ ASCII WORD
000184                              *                'THEN'
000185                              *                'BEGIN'
000186                              *        VALID BUT IN ERROR;
000187  010C  C840 FEF5                    LDR     4,D+2
000188  010E  C2D4                         LLH     4,=$R4
000189  010F  9856                         LDR     1,=$R6
000190  0110  D810 0000      X              LDR     5,<WORDCO.$R1
000191  0112  D2D5                         LLH     5,=$R5
000192  0113  C955                         CMR     4,=$R5
000193  0114  0901 0036                    BE      L20
000194        0116                  L19     EQU     $
000195                              *                     LABEL2 := 'LITERAL' (WC);
000196  0116  E870 5743                    LDR     6,=22339
000197  0118  EF40 0000      P              STR     6,LABEL2
000198  011A  C840 FEE8                    LDR     4,D+3
000199  011C  4061                         SAR     4,1
000200  011D  9854                         LDR     1,=$R4
000201  011E  D810 0000      X              LDR     5,<WORDCO.$R1
000202  0120  DF40 FEE4                    STR     5,D+5
000203                              *                     K := WORDCONVERSION @X/2@;
000204  0122  D840 FEDF                    LDR     $R5,D+2
000205  0124  C840 FEE0                    LDR     $R4,D+5
000206                              *                     LOAD R4 R5 (K, ASCII WORD);
000207  0126  D3C0 0000      P              LNJ     5,JRE
```

```
000208                                  *                       REPORT ERROR;
000209                                  *                       PUT TEXT (", WRONG 'VALID' CODE$");
000210  0128  FBC0 FEDD                         LAB     7,D+6
000211  012A  9BC0 FEF2                         LAB     1,T22
000212  012C  9F87                              STB     1,$B7
000213  012D  D3C0 0000         P               LNJ     5,VKPT
000214  012F  E840 FED3                         LDR     6,D+3
000215  0131  6061                              SAR     6,1
000216                                  *                       PUT HEX (WORD CONVERSION @X/2@);
000217  0132  FBC0 FED3                         LAB     7,D+6
000218  0134  9840 FED1                         LDR     1,D+6
000219  0136  C810 0000         X               LDR     4,<WORDCO.$R1
000220  0138  CF07                              STR     4,$B7
000221  0139  D3C0 0000         P               LNJ     5,VKPH
000222  013B  FBC0 FEC4                         LAB     7,D+6
000223  013D  E840 FEC4                         LDR     6,D+2
000224  013F  EF07                              STR     6,$B7
000225  0140  D3C0 0000         P               LNJ     5,VKPH
000226                                  *                       PUT HEX (ASCII WORD);
000227                                  *                       PUT HEX (1);
000228  0142  FBC0 FEC3                         LAB     7,D+6
000229  0144  E840 FEBC                         LDR     6,D+1
000230  0146  EF07                              STR     6,$B7
000231  0147  D3C0 0000         P               LNJ     5,VKPH
000232                                  *               'GO TO' ASK NEXT;
000233  0149  83C0 0000         P               JMP     ASKNEX
000234                                  *               'END';
000235                                  *           'END';
000236                                  *       END CBA;
000237                                  *       'END';
000238                                  **;
000239                                  **  LETTER L - TEST MODE - DATA LOOPBACK;
000240                                  **;
000241                                  *       'PROCEDURE' LETTER LP;
000242  014B  83C8 FEB4                 L20     JMP     *D
000243                                  *       'BEGIN'
000244                                  *           'INTEGER' I,ASCII WORD;
000245  014D  DFC0 FEBA                 KJLL      STB     5,D+8
000246                                  *           PUT NEW TEXT ("TEST MODE$");
000247  014F  FBC0 FEBB                         LAB     7,D+11
000248  0151  9BC0 FED7                         LAB     1,T24
000249  0153  9F87                              STB     1,$B7
000250  0154  D3C0 0000         P               LNJ     5,VKPNT
000251                                  *           CHECK STATE (SHOULD BE OFF);
000252  0156  FBC0 FEB4                         LAB     7,D+11
000253  0158  8707                              CL      $B7
000254  0159  D3C0 0000         P               LNJ     5,JCS
000255                                  *           'IF' ID 'EQ' 'HEX' (2008)
000256                                  *           'THEN'
000257                                  *           'BEGIN'
000258                                  **;
000259                                  **    CARD READER ONLY;
000260                                  **;
000261  015B  E840 0000         P               LDR     6,ID
000262  015D  E970 2008                         CMR     6,=8200
000263  015F  0981 00BB                         BNE     L25
000264                                  *               LABEL1 := 'LITERAL' (L1);
000265  0161  C870 4C31                         LDR     4,=19505
000266  0163  CF40 0000         P               STR     4,LABEL1
000267  0165  FBC0 FEA5                         LAB     7,D+11
000268  0167  9BC0 0000         P               LAB     1,BINARY
000269  0169  9F87                              STB     1,$B7
000270                                  *               INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000271  016A  ABC0 0000         P               LAB     2,IOWORD
000272  016C  AFC0 FE9F                         STB     2,D+12
000273  016E  D3C0 0000         P               LNJ     5,JIO
000274  0170  FBC0 FE9A                         LAB     7,D+11
000275  0172  9BC0 0000         P               LAB     1,BINARY
000276  0174  9F87                              STB     1,$B7
000277                                  *               INOUT (BINARY MODE, OUTPUT CONFIGURATION @0@); (SET BINARY)
000278  0175  0F00 000B         X      N1       NOP     <IOWORD+11
000279  0177  E840 FFFE                         LDR     6,N1+1
000280  0179  EF40 FE92                         STR     6,D+12
000281  017B  D3C0 0000         P               LNJ     5,JIO
000282                                  *               BINCOUNT@2@:='HEX'(9E40);  (MICRO RESET DATA REGISTER)
000283  017D  E870 9E40                         LDR     6,=-25024
000284  017F  EF40 0002         P               STR     6,BINCOU+2
000285  0181  FBC0 FE89                         LAB     7,D+11
000286  0183  9BC0 0000         P               LAB     1,IOXI
000287  0185  9F87                              STB     1,$B7
000288  0186  0F00 000E         X      N2       NOP     <IOWORD+14
000289  0188  C840 FFFE                         LDR     4,N2+1
000290  018A  CF40 FE81                         STR     4,D+12
000291  018C  D3C0 0000         P               LNJ     5,JIO
000292                                  *               INOUT (WAIT LOOP ADDRESS, GET WAIT ADDRESS @0@);
000293        018E                      L26     EQU     $
000294                                  *               'FOR' I:= 1 'STEP' 1 'UNTIL' 'HEX' (0FFF)
000295  018E  6C01                              LDV     6,1
000296  018F  EF40 FE79                         STR     6,D+9
000297                                  *               'DO'
000298                                  *               'BEGIN'
000299  0191  E840 FE77                 L27     LDR     6,D+9
000300  0193  E270 0FFF                         SUB     6,=4095
000301  0195  6A01 0036                         BGZ     6,L28
000302                                  *                   'INTEGER' DAPCOUNT;
000303  0197  FBC0 FE74                         LAB     7,D+12
000304  0199  9BC0 0000         P               LAB     1,BINCOU
000305  019B  9F87                              STB     1,$B7
000306  019C  6C0D                              LDV     6,13
000307  019D  EF40 FE6F                         STR     6,D+13
000308  019F  D3C0 0000         P               LNJ     5,JMC
000309                                  *                   MICRO (BINCOUNT);
000310  01A1  FBC0 FE6A                         LAB     7,D+12
000311  01A3  9BC0 FE67                         LAB     1,D+11
000312  01A5  9F87                              STB     1,$B7
000313                                  *                   INOUT (DAPCOUNT, GET WAIT ADDRESS @0@);
000314  01A6  0F00 000E         X      N3       NOP     <IOWORD+14
000315  01A8  E840 FFFE                         LDR     6,N3+1
000316  01AA  EF40 FE62                         STR     6,D+13
000317  01AC  D3C0 0000         P               LNJ     5,JIO
000318                                  *                   BINCOUNT @2@ := 0; (DON'T RESET DATA REG)
000319  01AE  8740 0002         P               CL      BINCOU+2
000320                                  *                   'IF' DAPCOUNT 'NE' I
```

```
000321                                  *                       'THEN'
000322                                  *                       'BEGIN'
000323  01B0  E840 FE5A                        LDR     6,D+11
000324  01B2  E940 FE56                        CMR     6,D+9
000325  01B4  0901 0013                        BE      L29
000326                                  *                               LABEL 2;= 'LITERAL' (CW);
000327  01B6  C870 4357                        LDR     4,=17239
000328  01B8  CF40 0000         P              STR     4,LABEL2
000329  01BA  D840 FE4E                        LDR     $R5,D+9
000330  01BC  C840 FE4E                        LDR     $R4,D+11
000331                                  *                               LOAD R4 R5 (DAPCOUNT, I);
000332  01BE  D3C0 0000         P              LNJ     5,JRE
000333                                  *                               REPORT ERROR;
000334  01C0  FBC0 FE4B                        LAB     7,D+12
000335  01C2  9BC0 FE6C                        LAB     1,T30
000336  01C4  9F87                             STB     1,$B7
000337  01C5  D3C0 0000         F              LNJ     5,VKPT
000338                                  *                               PUT TEXT ("; COUNTER WRONG$");
000339                                  *                               'GO TO' END BINCOUNT;
000340  01C7  0F85                             B       >L28
000341                                  *               'END';
000342                                  *             'END';
000343                                  *       END BINCOUNT;
000344  01C8  8AC0 FE40                 L29    INC     D+9
000345  01CA  0F81 FFC6                        B       L27
000346        01CC                      L28    EQU     $
000347                                  *             LABEL1 ;= 'LITERAL' (L2);
000348  01CC  E870 4C32                        LDR     6,=19506
000349  01CE  EF40 0000         P              STR     6,LABEL1
000350  01D0  FBC0 FE3A                        LAB     7,D+11
000351  01D2  9BC0 0000         P              LAB     1,BINARY
000352  01D4  9F87                             STB     1,$B7
000353                                  *             INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000354  01D5  ABC0 0000         P              LAB     2,IOWORD
000355  01D7  AFC0 FE34                        STB     2,D+12
000356  01D9  D3C0 0000         P              LNJ     5,JIO
000357  01DB  FBC0 FE2F                        LAB     7,D+11
000358  01DD  9BC0 0000         P              LAB     1,SETTOA
000359  01DF  9F87                             STB     1,$B7
000360                                  *             INOUT (SET TO ASCII, OUTPUT CONFIGURATION @0@);
000361  01E0  0F00 000B         X      N4      NOP     <IOWORD+11
000362  01E2  E840 FFFE                        LDR     6,N4+1
000363  01E4  EF40 FE27                        STR     6,D+12
000364  01E6  D3C0 0000         P              LNJ     5,JIO
000365                                  *             ASCCNT @2@ := 'HEX' (9C40); (MICRO RESET ASCII DATA REGISTER)
000366  01E8  E870 9C40                        LDR     6,=-25536
000367  01EA  EF40 0002         P              STR     6,ASCCNT+2
000368  01EC  8740 FE1C                 L32    CL      D+9
000369                                  *             'FOR' I:= 0 'STEP' 1 'UNTIL' (4095)
000370                                  *             'DO'
000371                                  *             'BEGIN'
000372  01EE  E840 FE1A                 L33    LDR     6,D+9
000373  01F0  E270 0FFF                        SUB     6,=4095
000374  01F2  6A01 01E8                        BGZ     6,L34
000375                                  *                       MICRO (ASCCNT);
000376  01F4  FBC0 FE16                        LAB     7,D+11
000377  01F6  9BC0 0000         P              LAB     1,ASCCNT
000378  01F8  9F87                             STB     1,$B7
000379  01F9  6C0A                             LDV     6,10
000380  01FA  EF40 FE11                        STR     6,D+12
000381  01FC  D3C0 0000         P              LNJ     5,JMC
000382  01FE  FBC0 FE0C                        LAB     7,D+11
000383  0200  9BC0 FE09                        LAB     1,D+10
000384  0202  9F87                             STB     1,$B7
000385                                  *                       INOUT (ASCII WORD, GET WAIT ADDRESS @0@);
000386  0203  0F00 000E         X      N5      NOP     <IOWORD+14
000387  0205  E840 FFFE                        LDR     6,N5+1
000388  0207  EF40 FE04                        STR     6,D+12
000389  0209  D3C0 0000         P              LNJ     5,JIO
000390                                  *                       ASCCNT @2@ := 0; (DON'T RESET DATA REG)
000391  020B  8740 0002         P              CL      ASCCNT+2
000392  020D  FBC0 FDFD                        LAB     7,D+11
000393  020F  E840 FDF9                        LDR     6,D+9
000394  0211  EF07                             STR     6,$B7
000395  0212  C840 FDF7                        LDR     4,D+10
000396  0214  CF40 FDF7                        STR     4,D+12
000397  0216  D3C0 FE53                        LNJ     5,P2
000398                                  *                       CONVERT BINARY TO ASCII (I, ASCII WORD);
000399                                  *             'END';
000400                                  *          'END'
000401  0218  8AC0 FDF0                        INC     D+9
000402  021A  0FD4                             B       >L33
000403                                  *             'ELSE'
000404                                  *             'BEGIN'
000405                                  **;
000406                                  **    READER / PUNCH;
000407                                  **;
000408                                  ** TEST BINARY OUT AND BACK;
000409        021B                      L25    EQU     $
000410                                  *             LABEL 1 ;= 'LITERAL' (L3);
000411  021B  E870 4C33                        LDR     6,=19507
000412  021D  EF40 0000         P              STR     6,LABEL1
000413  021F  FBC0 FDEB                        LAB     7,D+11
000414  0221  9BC0 0000         P              LAB     1,BINARY
000415  0223  9F87                             STB     1,$B7
000416                                  *             INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000417  0224  ABC0 0000         P              LAB     2,IOWORD
000418  0226  AFC0 FDE5                        STB     2,D+12
000419  0228  D3C0 0000         P              LNJ     5,JIO
000420  022A  FBC0 FDE0                        LAB     7,D+11
000421  022C  9BC0 0000         P              LAB     1,IOX1
000422  022E  9F87                             STB     1,$B7
000423  022F  0F00 000E         X      N6      NOP     <IOWORD+14
000424  0231  E840 FFFE                        LDR     6,N6+1
000425  0233  EF40 FDD8                        STR     6,D+12
000426  0235  D3C0 0000         P              LNJ     5,JIO
000427                                  *             INOUT (WAIT LOOP ADDRESS,GET WAIT ADDRESS @0@);
000428        0237                      L35    EQU     $
000429                                  *             'FOR' I:= 0 'STEP' 1 'UNTIL' 4095
000430  0237  8740 FDD1                        CL      D+9
000431                                  *             'DO'
000432                                  *             'BEGIN'
000433  0239  E840 FDCF                 L36    LDR     6,D+9
```

```
000434  023B  E270 0FFF                    SUB     6,=4095
000435  023D  6A01 004F                    BGZ     6,L37
000436                          *                       'INTEGER' LOOPCOUNT;
000437  023F  FBC0 FDCC                    LAB     7,D+12
000438  0241  9BC0 FDC7                    LAB     1,D+9
000439  0243  9F87                         STB     1,$B7
000440                          *                       INOUT (1, OUTPUT DATA @0@);
000441  0244  0F00 000F        X   N7      NOP     <IOWORD+15
000442  0246  E840 FFFE                    LDR     6,N7+1
000443  0248  EF40 FDC4                    STR     6,D+13
000444  024A  D3C0 0000        P           LNJ     5,JIO
000445                          *                       MICRO (BINOB);
000446  024C  FBC0 FDBF                    LAB     7,D+12
000447  024E  9BC0 0000        P           LAB     1,BINOB
000448  0250  9F87                         STB     1,$B7
000449  0251  6C14                         LDV     6,20
000450  0252  EF40 FDBA                    STR     6,D+13
000451  0254  D3C0 0000        P           LNJ     5,JMC
000452  0256  FBC0 FDB5                    LAB     7,D+12
000453  0258  9BC0 FDB2                    LAB     1,D+11
000454  025A  9F87                         STB     1,$B7
000455                          *                       INOUT (LOOPCOUNT, GET WAIT ADDRESS @0@);
000456  025B  0F00 000E        X   N8      NOP     <IOWORD+14
000457  025D  E840 FFFE                    LDR     6,N8+1
000458  025F  EF40 FDAD                    STR     6,D+13
000459  0261  D3C0 0000        P           LNJ     5,JIO
000460                          *                       'IF' LOOPCOUNT 'NE' I
000461                          *                       'THEN'
000462                          *                       'BEGIN'
000463  0263  E840 FDA7                    LDR     6,D+11
000464  0265  E940 FDA3                    CMR     6,D+9
000465  0267  0901 0021                    BE      L38
000466                          *                           LABEL 2:= 'LITERAL' (DL);
000467  0269  C870 444C                    LDR     4,=17484
000468  026B  CF40 0000        P           STR     4,LABEL2
000469  026D  D840 FD9B                    LDR     $R5,D+9
000470  026F  C840 FD9B                    LDR     $R4,D+11
000471                          *                           LOAD R4 R5 (LOOPCOUNT, I);
000472  0271  D3C0 0000        P           LNJ     5,JRE
000473                          *                           REPORT ERROR;
000474  0273  FBC0 FD98                    LAB     7,D+12
000475  0275  9BC0 FDC2                    LAB     1,T39
000476  0277  9F87                         STB     1,$B7
000477  0278  D3C0 0000        P           LNJ     5,VKPT
000478                          *                           PUT TEXT (", DATA LOOPBACK$");
000479  027A  FBC0 FD91                    LAB     7,D+12
000480  027C  E840 FD8E                    LDR     6,D+11
000481  027E  EF07                         STR     6,$B7
000482  027F  D3C0 0000        P           LNJ     5,VKPH
000483                          *                           PUT HEX (LOOPCOUNT);
000484                          *                           PUT HEX (I);
000485  0281  FBC0 FD8A                    LAB     7,D+12
000486  0283  E840 FD85                    LDR     6,D+9
000487  0285  EF07                         STR     6,$B7
000488  0286  D3C0 0000        P           LNJ     5,VKPH
000489                          *                       'GO TO' TEST RPROMS;
000490  0288  0F85                         B       >L37
000491                          *                   'END';
000492                          *               'END';   (OF BINARY OUT AND BACK)
000493                          *           TEST RPROMS;
000494  0289  6AC0 FD7F        L38         INC     D+9
000495  028B  0F81 FFAD                    B       L36
000496        028D              L37       EQU     $
000497                          *                   LABEL 1:= 'LITERAL' (L4);
000498  028D  E870 4C34                    LDR     6,=19508
000499  028F  EF40 0000        P           STR     6,LABEL1
000500  0291  FBC0 FD79                    LAB     7,D+11
000501  0293  9BC0 0000        P           LAB     1,BINARY
000502  0295  9F87                         STB     1,$B7
000503                          *                   INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000504  0296  ABC0 0000        P           LAB     2,IOWORD
000505  0298  AFC0 FD73                    STB     2,D+12
000506  029A  D3C0 0000        P           LNJ     5,JIO
000507  029C  FBC0 FD6E                    LAB     7,D+11
000508  029E  9BC0 0000        P           LAB     1,STATWA
000509  02A0  9F87                         STB     1,$B7
000510                          *                   INOUT (STATUS WAS, INPUT STATUS WORD @0@); (WAIT FOR OLT)
000511  02A1  0F00 000C        X   N9      NOP     <IOWORD+12
000512  02A3  E840 FFFE                    LDR     6,N9+1
000513  02A5  EF40 FD66                    STR     6,D+12
000514  02A7  D3C0 0000        P           LNJ     5,JIO
000515        02A9              L41       EQU     $
000516                          *                   'FOR' I:= 0 'STEP' 1 'UNTIL' 4095
000517  02A9  8740 FD5F                    CL      D+9
000518                          *                   'DO'
000519                          *                   'BEGIN'
000520  02AB  E840 FD5D                    LDR     6,D+9
000521  02AD  E270 0FFF                    SUB     6,=4095
000522  02AF  6A33                         BGZ     6,>L43
000523  02B0  FBC0 FD5A                    LAB     7,D+11
000524  02B2  9BC0 FD56                    LAB     1,D+9
000525  02B4  9F87                         STB     1,$B7
000526                          *                       INOUT (I, OUTPUT DATA @0@);
000527  02B5  0F00 000F        X   N10     NOP     <IOWORD+15
000528  02B7  E840 FFFE                    LDR     6,N10+1
000529  02B9  EF40 FD52                    STR     6,D+12
000530  02BB  D3C0 0000        P           LNJ     5,JIO
000531                          *                       MICRO (RPROMS);
000532  02BD  FBC0 FD4D                    LAB     7,D+11
000533  02BF  9BC0 0000        P           LAB     1,RPROMS
000534  02C1  9F87                         STB     1,$B7
000535  02C2  6C12                         LDV     6,18
000536  02C3  EF40 FD48                    STR     6,D+12
000537  02C5  D3C0 0000        P           LNJ     5,JMC
000538  02C7  FBC0 FD43                    LAB     7,D+11
000539  02C9  9BC0 FD40                    LAB     1,D+10
000540  02CB  9F87                         STB     1,$B7
000541                          *                       INOUT (ASCII WORD, GET WAIT ADDRESS @0@);
000542  02CC  0F00 000E        X   N11     NOP     <IOWORD+14
000543  02CE  E840 FFFE                    LDR     6,N11+1
000544  02D0  EF40 FD3B                    STR     6,D+12
000545  02D2  D3C0 0000        P           LNJ     5,JIO
000546  02D4  FBC0 FD36                    LAB     7,D+11
```

```
000547  02D6  E840 FD32                    LDR     6,D+9
000548  02D8  EF07                         STR     6,$B7
000549  02D9  C840 FD30                    LDR     4,D+10
000550  02DB  CF40 FD30                    STR     4,D+12
000551  02DD  D3C0 FD8C                    LNJ     5,P2
000552                        *                       CONVERT BINARY TO ASCII (I, ASCII WORD);
000553                        *                   'END';   (OF READER PROM TEST)
000554                        ** TEST PUNCH PROMS;
000555  02DF  8AC0 FD29                    INC     D+9
000556  02E1  0FCA                         B       >L42
000557        02E2                  L43    EQU     $
000558                        *                   LABEL 1 := 'LITERAL' (L5);
000559  02E2  E870 4C35                    LDR     6,=19509
000560  02E4  EF40 0000       P            STR     6,LABEL1
000561  02E6  FBC0 FD24                    LAB     7,D+11
000562  02E8  9BC0 0000       P            LAB     1,BINARY
000563  02EA  9F87                         STB     1,$B7
000564                        *                   INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000565  02EB  ABC0 0000       P            LAB     2,IOWORD
000566  02ED  AFC0 FD1E                    STB     2,D+12
000567  02EF  D3C0 0000       P            LNJ     5,J10
000568  02F1  FBC0 FD19                    LAB     7,D+11
000569  02F3  9BC0 0000       P            LAB     1,STATWA
000570  02F5  9F87                         STB     1,$B7
000571                        *                   INOUT (STATUS WAS, INPUT STATUS WORD @0@); (WAIT FOR GLT)
000572  02F6  0F00 000C       X      N12   NOP     <IOWORD+12
000573  02F8  E840 FFFE                    LDR     6,N12+1
000574  02FA  EF40 FD11                    STR     6,D+12
000575  02FC  D3C0 0000       P            LNJ     5,J10
000576        02FE                  L44    EQU     $
000577                        *                   'FOR' I:=0 'STEP' 1 'UNTIL' 256
000578  02FE  8740 FD0A                    CL      D+9
000579                        *                   'DO'
000580                        *                   'BEGIN'
000581  0300  E840 FD08                    L45     LDR     6,D+9
000582  0302  E270 0100                    SUB     6,=256
000583  0304  6A01 0048                    BGZ     6,L46
000584                        *                       'INTEGER' COUNT;
000585                        *                       'IF' 'BITS' @1,0@ I 'EQ' 0
000586                        *                       'THEN'
000587  0306  E840 FD02                    LDR     6,D+9
000588  0308  6B89                         BODD    6,>L47
000589                        *                           ASCII WORD :=  'BITS' @8,8@ WORD CONVERSION @I/2@
000590                        *                       'ELSE'
000591  0309  6061                         SAR     6,1
000592  030A  9856                         LDR     1,=$R6
000593  030B  C810 0000       X            LDR     4,<WORDCO,$R1
000594  030D  4048                         SOR     4,8
000595  030E  CF40 FCFB                    STR     4,D+10
000596  0310  0F8A                         B       >L48
000597        0311                  L47    EQU     $
000598                        *                           ASCII WORD := 'BITS' @8,0@ WORD CONVERSION @I/2@;
000599  0311  E840 FCF7                    LDR     6,D+9
000600  0313  6061                         SAR     6,1
000601  0314  9856                         LDR     1,=$R6
000602  0315  C810 0000       X            LDR     4,<WORDCO,$R1
000603  0317  C2D4                         LLH     4,=$R4
000604  0318  CF40 FCF1                    STR     4,D+10
000605  031A  FBC0 FCF1              L48    LAB     7,D+12
000606  031C  9BC0 FCED                    LAB     1,D+10
000607  031E  9F87                         STB     1,$B7
000608                        *                   INOUT (ASCII WORD, OUTPUT DATA @0@);
000609  031F  0F00 000F       X      N13   NOP     <IOWORD+15
000610  0321  E840 FFFE                    LDR     6,N13+1
000611  0323  EF40 FCE9                    STR     6,D+13
000612  0325  D3C0 0000       P            LNJ     5,J10
000613                        *                       MICRO (PPROMS);
000614  0327  FBC0 FCE4                    LAB     7,D+12
000615  0329  9BC0 0000       P            LAB     1,PPROMS
000616  032B  9F87                         STB     1,$B7
000617  032C  6C12                         LDV     6,18
000618  032D  EF40 FCDF                    STR     6,D+13
000619  032F  D3C0 0000       P            LNJ     5,JMC
000620  0331  FBC0 FCDA                    LAB     7,D+12
000621  0333  9BC0 FCD7                    LAB     1,D+11
000622  0335  9F87                         STB     1,$B7
000623                        *                   INOUT (COUNT, GET WAIT ADDRESS @0@);
000624  0336  0F00 000E       X      N14   NOP     <IOWORD+14
000625  0338  E840 FFFE                    LDR     6,N14+1
000626  033A  EF40 FCD2                    STR     6,D+13
000627  033C  D3C0 0000       P            LNJ     5,J10
000628  033E  FBC0 FCCD                    LAB     7,D+12
000629  0340  E840 FCCA                    LDR     6,D+11
000630  0342  EF07                         STR     6,$B7
000631  0343  C840 FCC6                    LDR     4,D+10
000632  0345  CF40 FCC7                    STR     4,D+13
000633  0347  D3C0 FD22                    LNJ     5,P2
000634                        *                       CONVERT BINARY TO ASCII (COUNT, ASCII WORD);
000635                        *                   'END';   (OF PUNCH PROM TEST)
000636                        *           'COMMENT' TEST CARD JAM TIME-OUT ONE SHOT;
000637  0349  8AC0 FCBF                    INC     D+9
000638  034B  0F81 FFB4                    B       L45
000639        034D                  L46    EQU     $
000640                        *                   ZHRTCC := HERTZ/2;  (WAIT 1/2 SECOND)
000641  034D  E840 0000       P            LDR     6,HERTZ
000642  034F  6061                         SAR     6,1
000643  0350  EF40 0000       P            STR     6,ZHRTCC
000644                        *                   ZHRTCI :=HERTZ/2;
000645  0352  C840 0000       P            LDR     4,HERTZ
000646  0354  4061                         SAR     4,1
000647  0355  CF40 0000       P            STR     4,ZHRTCI
000648                        *                   CLOCK SEMAPHORE := 0;
000649  0357  8740 0000       P            CL      CKSEM
000650                        *                   STATUS WAS := 0;
000651  0359  8740 0000       P            CL      STATWA
000652                        *                   LABEL 1 := 'LITERAL' (L6);
000653  035B  D870 4C36                    LDR     5,=19510
000654  035D  DF40 0000       P            STR     5,LABEL1
000655  035F  FBC0 FCAB                    LAB     7,D+11
000656  0361  9BC0 0000       P            LAB     1,BINARY
000657  0363  9F87                         STB     1,$B7
000658                        *                   INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000659  0364  ABC0 0000       P            LAB     2,IOWORD
```

```
000660  0366  AFC0 FCA5                    STB     2,D+12
000661  0368  D3C0 0000      P             LNJ     5,JIO
000662  036A  FBC0 FCA0                    LAB     7,D+11
000663  036C  9BC0 0000      P             LAB     1,STATWA
000664  036E  9F87                         STB     1,$B7
000665                              *               INOUT (STATUS WAS, INPUT STATUS WORD @0@); (WAIT FOR QLI)
000666                              *       'COMMENT' WAIT A MAXIMUM OF 1/2 SECOND FOR TIME-OUT TO OCCUR;
000667  036F  0F00 000C      X     N15       NOP     <IOWORD+12
000668  0371  E840 FFFE                    LDR     6,N15+1
000669  0373  EF40 FC98                    STR     6,D+12
000670  0375  D3C0 0000      P             LNJ     5,JIO
000671  0377  0004                         DC      4
000672                              *               CLOCK ON;
000673                              *               MICRO (ONESHOT);
000674                              *       RETRY;
000675  0378  FBC0 FC92                    LAB     7,D+11
000676  037A  9BC0 0000      P             LAB     1,ONESHO
000677  037C  9F87                         STB     1,$B7
000678  037D  6C09                         LDV     6,9
000679  037E  EF40 FC8D                    STR     6,D+12
000680  0380  D3C0 0000      P             LNJ     5,JMC
000681        0382                L49       EQU     $
000682                              *               'IF' CLOCK SEMAPHORE 'NE' 1
000683                              *               'THEN'
000684                              *               'BEGIN'
000685  0382  E840 0000      P             LDR     6,CKSEM
000686  0384  6D01                         CMV     6,1
000687  0385  0914                         BE      >L50
000688  0386  FBC0 FC84                    LAB     7,D+11
000689  0388  9BC0 0000      P             LAB     1,STATWA
000690  038A  9F87                         STB     1,$B7
000691                              *                       INOUT (STATUS WAS, INPUT STATUS WORD @0@);
000692  038B  0F00 000C      X     N16       NOP     <IOWORD+12
000693  038D  C840 FFFE                    LDR     4,N16+1
000694  038F  CF40 FC7C                    STR     4,D+12
000695  0391  D3C0 0000      P             LNJ     5,JIO
000696                              *                       'IF' 'BITS' @1,9@ STATUS WAS 'EQ' 1
000697                              *                       'THEN'
000698                              *                               'GO TO' EXIT;
000699  0393  82C0 0000      P             LB      STATWA,2'0200'
      0395  0200
000700  0396  0501 0012                    BBT     L52
000701                              *                       'GO TO' RETRY;
000702  0398  0FEA                         B       >L49
000703                              *               'END';
000704        0399                L50       EQU     $
000705  0399  0005                         DC      5
000706                              *               CLOCK OFF;
000707                              *               LABEL 2 := 'LITERAL' (NT);
000708  039A  E870 4E54                    LDR     6,=20052
000709  039C  EF40 0000      P             STR     6,LABEL2
000710  039E  D3C0 0000      P             LNJ     5,JRE
000711                              *               REPORT ERROR;
000712                              *               PUT TEXT (", JAM ONE SHOT DIDN'T TIME-OUT$");
000713  03A0  FBC0 FC6A                    LAB     7,D+11
000714  03A2  9BC0 FC9E                    LAB     1,T53
000715  03A4  9F87                         STB     1,$B7
000716  03A5  D3C0 0000      P             LNJ     5,VKPT
000717                              *               'GO TO' END LETTER L;
000718                              *       EXIT:
000719  03A7  0F61 0033                    B       L34
000720        03A9                L52       EQU     $
000721  03A9  0005                         DC      5
000722                              *               CLOCK OFF;
000723                              *       'COMMENT' CHECK TO SEE IF TIME-OUT BEFORE 150 MSEC;
000724                              *               'BEGIN'
000725                              *                       'INTEGER' TICKS,MSEC;
000726                              *                       TICKS := ZHRTC1-ZHRTCC;
000727  03AA  E840 0000      P             LDR     6,ZHRTC1
000728  03AC  E240 0000      P             SUB     6,ZHRTCC
000729  03AE  EF40 FC5C                    STR     6,D+11
000730                              *                       MSEC := 1000*TICKS/HERTZ;
000731  03B0  F840 FC5A                    LDR     7,D+11
000732  03B2  FB70 03E8                    MUL     7,=1000
000733  03B4  F340 0000      P             DIV     7,HERTZ
000734  03B6  FF40 FC55                    STR     7,D+12
000735                              *                       LABEL 2 := 'LITERAL' (TW);
000736  03B8  E870 5451                    LDR     6,=21585
000737  03BA  EF40 0000      P             STR     6,LABEL2
000738                              *                       'IF' MSEC 'LT' 150
000739                              *                       'THEN'
000740                              *                       'BEGIN'
000741  03BC  F970 0096                    CMR     7,=150
000742  03BE  0881 001C                    BAGE    L34
000743  03C0  0840 0000      P             LDR     $R5,SETTOA
000744  03C2  C840 FC49                    LDR     $R4,D+12
000745                              *                               LOAD R4 R5 (MSEC,CLEAR);
000746  03C4  D3C0 0000      P             LNJ     5,JRE
000747                              *                               REPORT ERROR;
000748                              *                               PUT TEXT (", JAM ONE SHOT TIMED OUT IN$");
000749  03C6  FBC0 FC46                    LAB     7,D+13
000750  03C8  9BC0 FC89                    LAB     1,T56
000751  03CA  9F87                         STB     1,$B7
000752  03CB  D3C0 0000      P             LNJ     5,VKPT
000753  03CD  FBC0 FC3F                    LAB     7,D+13
000754  03CF  E840 FC3C                    LDR     6,D+12
000755  03D1  EF07                         STR     6,$B7
000756  03D2  D3C0 0000      P             LNJ     5,JPN
000757                              *                               PUT NUMBER (MSEC);
000758                              *                               PUT TEXT (" MSEC$");
000759  03D4  FBC0 FC38                    LAB     7,D+13
000760  03D6  9BC0 FC8A                    LAB     1,T57
000761  03D8  9F87                         STB     1,$B7
000762  03D9  D3C0 0000      P             LNJ     5,VKPT
000763                              *                       'END';
000764                              *               'END';
000765                              *       'END';          (OF TEST MODE FOR 206A)
000766                              *       END LETTER L;
000767                              *       'END';  (OF DATA PROCEDURE)
000768                              *       'END';  (OF SEGMENT)
000769                              *       'FINISH'
000770  03DB  83C8 FC2C             L34       JMP     *D+8
000771        03DD                L1        EQU     $
```

```
000772  03DD  83C0 0000                    JMP       ZK$EX
000773                                     XLOC      ZK$EX
000774                                     CTRL LINK ZK$EX
000775                                     XLOC      ZHRTCI
000776                                     XLOC      ZHRTCC
000777                                     XLOC      VKPI
000778                                     XLOC      VKPNT
000779                                     XLOC      VKPH
000780                                     XLOC      ID
000781                                     XLOC      STATWA
000782                                     XLOC      HERTZ
000783                                     XLOC      SETTOA
000784                                     XLOC      BINARY
000785                                     XLOC      LABEL1
000786                                     XLOC      LABEL2
000787                                     XLOC      CKSEM
000788                                     XLOC      IOX1
000789                                     XLOC      BINCOU
000790                                     XLOC      ASCCNT
000791                                     XLOC      BINOB
000792                                     XLOC      PPROMS
000793                                     XLOC      RPROMS
000794                                     XLOC      ONESHO
000795                                     XLOC      WORDCO
000796                                     XLOC      IOWORD
000797                                     XLOC      JRE
000798                                     XLOC      JIO
000799                                     XLOC      JPN
000800                                     XLOC      JCS
000801                                     XLOC      JMC
000802                                     XLOC      ASKNEX
000803  03DF  0065                         END       ZVKJLL,ZVKJLL
0000  ERR COUNT
L
```

```
E   ZVKJLM.LIST                                  05/19/78  1427.4R W 05/19/78  1422.7      2115270000
000001                                                      TITLE     ZVKJLM,'REV 00'
000002          0035                                        XDEF      KJLM
000003          0030                                        XDEF      ZVKJLM
000004          0008                                ZKCOM   COMM      8
000005  0000    0000                                D       RESV      10,0
000006  000A    0016                                T5      DC        22
000007  000B    5245 4144 5920                              TEXT      'READY PUNCH FOR AS'
        000E    5055 4E43 4820
                464F 5220 4153
000008  0014    4349 4924                                   TEXT      'CII$'
000009  0016    000E                                T9      DC        14
000010  0017    2C20 4449 444E                              TEXT      ', DIDN''T RUPT$'
        001A    2754 2052 5550
                5424
000011  001E    0011                                T14     DC        17
000012  001F    2C20 4D45 4D4F                              TEXT      ', MEMORY ADDRESS$'
        0022    5259 2041 4444
                5245 5353 2400
000013  0028    000E                                T16     DC        14
000014  0029    2C20 5241 4E47                              TEXT      ', RANGE ERROR$'
        002C    4520 4552 524F
                5224
000015          0030                                ZVKJLM  EQU       $
000016  0030    8F00 0001       K                           SAVE      <ZKCOM+1,Z'0011'
        0032    0011
000017                                                      *       'BEGIN'
000018  0033    0F81 014D                                   B         L1
000019                                                      *           'INTEGER' TEMP;
000020                                                      *           'INTEGER' RANGE;
000021  0035    DFC0 FFCA                           KJLM    STB       5,D
000022                                                      *           'IF' ID 'EQ' 'HEX' (2008)
000023                                                      *             'THEN'
000024                                                      *                   'GO TO' END OF LETTER M;          (READER ONLY)
000025  0037    E840 0000       P                           LDR       6,ID
000026  0039    E970 2008                                   CMR       6,=8200
000027  003B    0901 0143                                   BE        L4
000028                                                      *           ERROR LABEL := 'LITERAL' (M1);
000029  003D    C870 4D31                                   LDR       4,=19761
000030  003F    CF40 0000       P                           STR       4,LABEL1
000031                                                      *           PUT NEW TEXT ("READY PUNCH FOR ASCII$");
000032  0041    FBC0 FFC1                                   LAB       7,D+3
000033  0043    9BC0 FFC6                                   LAB       1,T5
000034  0045    9F87                                        STB       1,$B7
000035  0046    D3C0 0000       P                           LNJ       5,VKPNT
000036  0048    D3C0 0000       P                           LNJ       5,JWR
000037                                                      *           WAIT FOR RETURN;
000038                                                      *           CHECK STATE (SHOULD BE ON);
000039  004A    FBC0 FFB8                                   LAB       7,D+3
000040  004C    6C01                                        LDV       6,1
000041  004D    EF07                                        STR       6,$B7
000042  004E    D3C0 0000       P                           LNJ       5,JCS
000043  0050    FBC0 FFB2                                   LAB       7,D+3
000044  0052    9BC0 0000       P                           LAB       1,ASEJCT
000045  0054    9F87                                        STB       1,$B7
000046                                                      *           INOUT (ASCII ERROR EJECT, OUTPUT TASK @0@);
000047  0055    0F00 0005       X               N1          NOP       <IOWORD+5
000048  0057    E840 FFFE                                   LDR       6,N1+1
000049  0059    EF40 FFAA                                   STR       6,D+4
000050  005B    D3C0 0000       P                           LNJ       5,JIO
000051                                                      *           STATUS SHOULD BE := 'HEX' (8000);
000052                                                      **  WAIT FOR FIVE SECONDS FOR FIRST CARD TO RUPT;
000053  005D    E870 8000                                   LDR       6,=-32768
000054  005F    EF40 0000       P                           STR       6,STATUS
000055                                                      *           ZHRTCC := HERTZ *5;
000056  0061    C840 0000       P                           LDR       4,HERTZ
000057  0063    4F05                                        MLV       4,5
000058  0064    CF40 0000       P                           STR       4,ZHRTCC
000059                                                      *           ZHRTCI := HERTZ *5;
000060  0066    D840 0000       P                           LDR       5,HERTZ
000061  0068    5F05                                        MLV       5,5
000062  0069    DF40 0000       P                           STR       5,ZHRTCI
000063                                                      *           ZHRTCL := 20;
000064  006B    7C14                                        LDV       7,20
000065  006C    FF40 0000       P                           STR       7,ZHRTCL
000066                                                      *           CLOCK SEMAPHORE := 0;
000067  006E    8740 0000       P                           CL        CKSEM
000068                                                      *           L21 SEMAPHORE := 0;
000069                                                      **  SET DEVICE TO LEVEL 21;
000070  0070    8740 0000       P                           CL        L21SEM
000071                                                      *           TEMP := SREG 'MASK' 'HEX' (03C0) 'UNION' 21;
000072  0072    9840 0000       P                           LDR       1,SREG
000073  0074    9570 03C0                                   AND       1,=960
000074  0076    9470 0015                                   OR        1,=21
000075  0078    9F40 FF88                                   STR       1,D+1
000076  007A    FBC0 FF88                                   LAB       7,D+3
000077  007C    9BC0 FF84                                   LAB       1,D+1
000078  007E    9F87                                        STB       1,$B7
000079                                                      *           INOUT (TEMP, OUTPUT INTERRUPT @0@);
000080  007F    0F00 0002       X               N2          NOP       <IOWORD+2
000081  0081    A840 FFFE                                   LDR       2,N2+1
000082  0083    AF40 FF80                                   STR       2,D+4
000083  0085    D3C0 0000       P                           LNJ       5,JIO
000084                                                      *           READ STATUS := FALSE;
000085  0087    8740 0000       P                           CL        READSI
000086  0089    FBC0 FF79                                   LAB       7,D+3
000087  008B    9BC0 0000       P                           LAB       1,CHAR
000088  008D    9F87                                        STB       1,$B7
000089  008E    0F00 0010       X               N3          NOP       <IOWORD+16
000090  0090    E840 FFFE                                   LDR       6,N3+1
000091  0092    EF40 FF71                                   STR       6,D+4
000092                                                      *           IOLD (CHAR @0@,PUNCH @0@,80);          (PUNCH ASCII CARD)
000093  0094    4C50                                        LDV       4,80
000094  0095    CF40 FF6F                                   STR       4,D+5
000095  0097    D3C0 0000       P                           LNJ       5,JIL
000096  0099    0004                                        DC        4
000097                                                      *           CLOCK ON;
000098  009A    FBC0 FF68                                   LAB       7,D+3
000099  009C    9BC0 0000       P                           LAB       1,STATWA
000100  009E    9F87                                        STB       1,$B7
000101                                                      *           INOUT (STATUS WAS,INPUT STATUS WORD @0@);
000102                                                      *       WAIT;
000103  009F    0F00 000C       X               N4          NOP       <IOWORD+12
```

```
000104  00A1  E840 FFFE                    LDR     6,N4+1
000105  00A3  EF40 FF60                    STR     6,D+4
000106  00A5  D3C0 0000      P             LNJ     5,JIO
000107        00A7                  L6      EQU     $
000108                              *       'IF' L21 SEMAPHORE 'EQ' 0
000109                              *       'THEN'
000110                              *       'BEGIN'
000111  00A7  E840 0000      P             LDR     6,L21SEM
000112  00A9  6993                         BNEZ    6,>L7
000113                              *       'IF' CLOCK SEMAPHORE 'EQ'0
000114                              *       'THEN'
000115                              *           'GO TO' WAIT;
000116  00AA  C840 0000      P             LDR     4,CKSEM
000117  00AC  497B                         BEZ     4,>L6
000118                              *       LABEL 2 := 'LITERAL' (R1);
000119  00AD  D870 5231                     LDR     5,=21041
000120  00AF  DF40 0000      P             STR     5,LABEL2
000121  00B1  D3C0 0000      P             LNJ     5,JRE
000122                              *       REPORT ERROR;
000123  00B3  FBC0 FF4F                     LAB     7,D+3
000124  00B5  9BC0 FF60                     LAB     1,T9
000125  00B7  9F87                         STB     1,$B7
000126  00B8  D3C0 0000      P             LNJ     5,VKPT
000127                              *       PUT TEXT (", DIDN'T RUPT$");
000128                              *       'GO TO' CONTINUE;
000129  00BA  0F81 0001                     B       L7
000130                              *   'END';
000131                              *  CONTINUE:
000132        00BC                  L7      EQU     $
000133  00BC  0005                         DC      5
000134                              *           CLOCK OFF;
000135  00BD  D3C0 0000      P             LNJ     5,JSE
000136                              *           CHECK STATUS FOR ERRORS;
000137                              **  MEMORY BYTE, MEMORY MODULE ADDRESS CORRECT ?;
000138                              *       'BEGIN'
000139                              *           'INTEGER' MEMORY BYTE, MEMORY MODULE, ADDRESS IS;
000140  00BF  FBC0 FF46                     LAB     7,D+6
000141  00C1  9BC0 FF41                     LAB     1,D+3
000142  00C3  9F87                         STB     1,$B7
000143                              *           INOUT (MEMORY BYTE, INPUT MEMORY BYTE @0@);
000144  00C4  0F00 0006      X       N5    NOP     <IOWORD+6
000145  00C6  E840 FFFE                     LDR     6,N5+1
000146  00C8  EF40 FF3E                     STR     6,D+7
000147  00CA  D3C0 0000      P             LNJ     5,JIO
000148  00CC  FBC0 FF39                     LAB     7,D+6
000149  00CE  9BC0 FF35                     LAB     1,D+4
000150  00D0  9F87                         STB     1,$B7
000151  00D1  0F00 0008      X       N6    NOP     <IOWORD+8
000152  00D3  E840 FFFE                     LDR     6,N6+1
000153  00D5  EF40 FF31                     STR     6,D+7
000154  00D7  D3C0 0000      P             LNJ     5,JIO
000155                              *           INOUT (MEMORY MODULE, INPUT MODULE ADDRESS@0@);
000156                              *           'IF' 'BITS' @1,0@ MEMORY MODULE 'EQ' 1
000157                              *           'THEN'
000158  00D9  E840 FF2A                     LDR     6,D+4
000159  00DB  6B09                         BEVN    6,>L11
000160                              *               ADDRESS IS := 'HEX' (8000)'UNION''BITS'@15,1@MEMORYBYTE
000161                              *           'ELSE'
000162  00DC  C840 FF26                     LDR     4,D+3
000163  00DE  4041                         SOR     4,1
000164  00DF  C470 8000                     OR      4,=-32768
000165  00E1  CF40 FF23                     STR     4,D+5
000166  00E3  0F86                         B       >L12
000167        00E4                  L11     EQU     $
000168                              *               ADDRESS IS := 'BITS' @15,1@ MEMORY BYTE;
000169  00E4  E840 FF1E                     LDR     6,D+3
000170  00E6  6041                         SOR     6,1
000171  00E7  EF40 FF1D                     STR     6,D+5
000172        00E9                  L12     EQU     $
000173                              *       'IF' 'LOCATION'(CHAR @40@)
000174                              *       'NE' ADDRESS IS
000175                              *       'THEN'
000176                              *       'BEGIN'
000177  00E9  0F00 0028      X       N7    NOP     <CHAR+40
000178  00EB  E840 FFFE                     LDR     6,N7+1
000179  00ED  E940 FF17                     CMR     6,D+5
000180  00EF  0901 0012                     BE      L13
000181                              *           LABEL 2:= 'LITERAL' (MA);
000182  00F1  C870 4D41                     LDR     4,=19777
000183  00F3  CF40 0000      P             STR     4,LABEL2
000184  00F5  D840 0000      P             LDR     $R5,SETTOA
000185  00F7  C840 FF0D                     LDR     $R4,D+5
000186  00F9  D3C0 0000      P             LNJ     5,JRE
000187                              *           REPORT ERROR;
000188  00FB  FBC0 FF0A                     LAB     7,D+6
000189  00FD  9BC0 FF20                     LAB     1,T14
000190  00FF  9F87                         STB     1,$B7
000191  0100  D3C0 0000      P             LNJ     5,VKPT
000192                              *           PUT TEXT (", MEMORY ADDRESS$");
000193                              *       'END';
000194  0102  FBC0 FF03                L13  LAB     7,D+6
000195  0104  9BC0 FEFD                     LAB     1,D+2
000196  0106  9F87                         STB     1,$B7
000197                              *           INOUT (RANGE, INPUT RANGE @0@);
000198  0107  0F00 0009      X       N8    NOP     <IOWORD+9
000199  0109  E840 FFFE                     LDR     6,N8+1
000200  010B  EF40 FEFB                     STR     6,D+7
000201  010D  D3C0 0000      P             LNJ     5,JIO
000202                              *           'IF' RANGE 'NE' 0
000203                              *           'THEN'
000204                              *           'BEGIN'
000205  010F  E840 FEF2                     LDR     6,D+2
000206  0111  6901 0012                     BEZ     6,L15
000207                              *               LABEL 2:= 'LITERAL' (RE);
000208  0113  C870 5245                     LDR     4,=21061
000209  0115  CF40 0000      P             STR     4,LABEL2
000210  0117  D840 0000      P             LDR     $R5,SETTOA
000211  0119  C840 FEE8                     LDR     $R4,D+2
000212                              *               LOAD R4 R5 (RANGE, CLEAR);
000213  011B  D3C0 0000      P             LNJ     5,JRE
000214                              *               REPORT ERROR;
000215  011D  FBC0 FEE8                     LAB     7,D+6
000216  011F  9BC0 FF08                     LAB     1,T16
```

```
000217  0121  9F87                          STB     1,$B7
000218  0122  D3C0 0000      P               LNJ     5,VKPT
000219                             *                  PUT TEXT (", RANGE ERROR$");
000220                             *                'END';
000221                             *           'END';
000222        0124                 L15       EQU     $
000223                             *             ERROR LABEL := 'LITERAL' (M2);
000224  0124  E870 4D32                        LDR     6,=19762
000225  0126  EF40 0000      P               STR     6,LABEL1
000226  0128  FBC0 FED4                      LAB     7,D+3
000227  012A  9BC0 0000      P               LAB     1,SETTOH
000228  012C  9F87                          STB     1,$B7
000229                             *             INOUT (SET TO HOLLERITH,OUTPUT TASK@0@);
000230  012D  0F00 0005      X       N9      NOP     <IOWORD+5
000231  012F  C840 FFFE                      LDR     4,N9+1
000232  0131  CF40 FED2                      STR     4,D+4
000233  0133  D3C0 0000      P               LNJ     5,JIO
000234                             *             STATUS SHOULD BE := 'HEX' (8000);
000235  0135  E870 8000                      LDR     6,=-32768
000236  0137  EF40 0000      P               STR     6,STATUS
000237  0139  FBC0 FEC9                      LAB     7,D+3
000238  013B  9BC0 0000      P               LAB     1,ILLEGA
000239  013D  9F87                          STB     1,$B7
000240  013E  0F00 0010      X       N10     NOP     <IOWORD+16
000241  0140  C840 FFFE                      LDR     4,N10+1
000242  0142  CF40 FEC1                      STR     4,D+4
000243  0144  5C04                          LDV     5,4
000244  0145  DF40 FEBF                      STR     5,D+5
000245  0147  D3C0 0000      P               LNJ     5,JIL
000246                             *             IOLD (ILLEGAL ASCII @0@,PUNCH @0@,4);(PUNCH ILLEGAL PUNCHES)
000247  0149  D3C0 0000      P               LNJ     5,JSE
000248                             *             CHECK STATUS FOR ERRORS;
000249                             *             ERROR LABEL := 'LITERAL' (M3);
000250  014B  E870 4D33                        LDR     6,=19763
000251  014D  EF40 0000      P               STR     6,LABEL1
000252  014F  FBC0 FEB3                      LAB     7,D+3
000253  0151  9BC0 0000      P               LAB     1,SETTOA
000254  0153  9F87                          STB     1,$B7
000255                             *             INOUT (SET TO ASCII,OUTPUT TASK @0@);
000256  0154  0F00 0005      X       N11     NOP     <IOWORD+5
000257  0156  C840 FFFE                      LDR     4,N11+1
000258  0158  CF40 FEAB                      STR     4,D+4
000259  015A  D3C0 0000      P               LNJ     5,JIO
000260  015C  FBC0 FEA6                      LAB     7,D+3
000261  015E  9BC0 0000      P               LAB     1,BUFF1
000262  0160  9F87                          STB     1,$B7
000263  0161  6C54                          LDV     6,84
000264  0162  EF40 FEA1                      STR     6,D+4
000265  0164  4C28                          LDV     4,40
000266  0165  CF40 FE9F                      STR     4,D+5
000267                             *             FILL BUFFER (BUFFER 1,40,'HEX'(2020));
000268  0167  D870 2020                      LDR     5,=8224
000269  0169  DF40 FE9C                      STR     5,D+6
000270  016B  D3C0 0000      P               LNJ     5,JFB
000271  016D  FBC0 FE95                      LAB     7,D+3
000272  016F  9BC0 0000      P               LAB     1,BUFF1
000273  0171  9F87                          STB     1,$B7
000274  0172  0F00 0010      X       N12     NOP     <IOWORD+16
000275  0174  E840 FFFE                      LDR     6,N12+1
000276  0176  EF40 FE8D                      STR     6,D+4
000277                             *             IOLD (BUFFER1@0@, PUNCH @0@,80);
000278  0178  4C50                          LDV     4,80
000279  0179  CF40 FE8B                      STR     4,D+5
000280  017B  D3C0 0000      P               LNJ     5,JIL
000281  017D  D3C0 0000      P               LNJ     5,JSE
000282                             *             CHECK STATUS FOR ERRORS;
000283                             *       END OF LETTER M;
000284                             *       'END';      (OF PROCEDURE)
000285                             *       'END';      (OF SEGMENT)
000286                             *       'FINISH'
000287  017F  83C8 FE80                        L4      JMP     *D
000288        0181                 L1        EQU     $
000289  0181  83C0 0000      P               JMP     ZK$EX
000290                                       XLOC    ZK$EX
000291                                       CTRL LINK ZK$EX
000292                                       XLOC    ZHRTCI
000293                                       XLOC    ZHRTCL
000294                                       XLOC    ZHRTCC
000295                                       XLOC    VKPT
000296                                       XLOC    VKPNT
000297                                       XLOC    ID
000298                                       XLOC    STATWA
000299                                       XLOC    STATUS
000300                                       XLOC    HERTZ
000301                                       XLOC    SETTOH
000302                                       XLOC    SETTOA
000303                                       XLOC    ASEJCT
000304                                       XLOC    READST
000305                                       XLOC    LABEL1
000306                                       XLOC    LABEL2
000307                                       XLOC    CKSEM
000308                                       XLOC    LZISEM
000309                                       XLOC    SREG
000310                                       XLOC    BUFF1
000311                                       XLOC    CHAR
000312                                       XLOC    ILLEGA
000313                                       XLOC    IOWORD
000314                                       XLOC    JRE
000315                                       XLOC    JIO
000316                                       XLOC    JIL
000317                                       XLOC    JWR
000318                                       XLOC    JSE
000319                                       XLOC    JCS
000320                                       XLOC    JFB
000321  0183  0030                          END     ZVKJLM,ZVKJLM
0000  ERR COUNT
L
```

```
E   ZVKJLN.LIST                                  05/19/78  1429.3R W 05/19/78  1422.7      4893930000
000001                                                      TITLE     ZVKJLN,'REV 00'
000002          0077                                        XDEF      KJLN
000003          0072                                        XDEF      ZVKJLN
000004          0008                              ZKCOM     COMM      8
000005  0000    0000                              D         RESV      11,0
000006  000B    001A                              T6        DC        26
000007  000C    5245 4144 5920                              TEXT      'READY ASCII DECK F'
        000F    4153 4349 4920
                4445 434B 2046
000008  0015    4F52 2052 4541                              TEXT      'OR READ$'
        0018    4424
000009  0019    000E                              T12       DC        14
000010  001A    2C20 4449 444E                              TEXT      ', DIDN''T RUPT$'
        001D    2754 2052 5550
                5424
000011  0021    0011                              T18       DC        17
000012  0022    2C20 4D45 4D4F                              TEXT      ', MEMORY ADDRESS$'
        0025    5259 2041 4444
                5245 5353 2400
000013  002B    000F                              T20       DC        15
000014  002C    2C20 5241 4E47                              TEXT      ', RANGE ERROR $'
        002F    4520 4552 524F
                5220 2400
000015  0034    0017                              T39       DC        23
000016  0035    494C 4C45 4741                              TEXT      'ILLEGAL ASCII NOT '
        0038    4C20 4153 4349
                4920 4E4F 5420
000017  003E    5245 4144 2400                              TEXT      'READ$'
000018  0041    000F                              T43       DC        15
000019  0042    2C20 5846 4552                              TEXT      ', XFER TO MANY$'
        0045    2054 4F20 4D41
                4E59 2400
000020  004A    000E                              T45       DC        14
000021  004B    2C20 5241 4E47                              TEXT      ', RANGE WRONG$'
        004E    4520 5752 4F4E
                4724
000022  0052    001B                              T49       DC        27
000023  0053    5345 434F 4E44                              TEXT      'SECOND CARD OFFSET'
        0056    2043 4152 4420
                4F46 4653 4554
000024  005C    2053 5441 434B                              TEXT      ' STACKED$'
        005F    4544 2400
000025  0061    000E                              T52       DC        14
000026  0062    2C20 4552 524F                              TEXT      ', ERROR EJECT$'
        0065    5220 454A 4543
                5424
000027  0069    000F                              T53       DC        15
000028  006A    5354 4F50 2C20                              TEXT      'STOP, RUNOUT',Z'0D0A','$'
        006D    5255 4E4F 5554
        0070    0D0A 2400
000029          0072                              ZVKJLN    EQU       $
000030  0072    8F00 0001           K                       SAVE      <ZKCOM+1,Z'0011'
        0074    0011
000031                                            *       'BEGIN'
000032  0075    0F81 02EA                                   B         L1
000033                                            *           'INTEGER' I,ASCII BIT;
000034                                            *           'INTEGER' RANGE;
000035  0077    0FC0 FF88                         KJLN      STB       5,D
000036                                            *           'IF' ID 'EQ' 'HEX' (2088)
000037                                            *           'THEN'
000038                                            *               'GO TO' END OF LETTER N;          (PUNCH ONLY)
000039                                            *       RESTART PROCEDURE;
000040  0079    E840 0000           P                       LDR       6,ID
000041  007B    E970 2088                                   CMR       6,=8328
000042  007D    0901 02E0                                   BE        L47
000043          007F                              L5        EQU       $
000044                                            *           ERROR LABEL := 'LITERAL' (N1);
000045  007F    E870 4E31                                   LDR       6,=20017
000046  0081    EF40 0000           P                       STR       6,LABEL1
000047  0083    FBC0 FF80           P                       LAB       7,D+4
000048  0085    9BC0 0000           P                       LAB       1,BINARY
000049  0087    9F87                                        STB       1,$B7
000050                                            *           INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000051  0088    ABC0 0000           P                       LAB       2,IOWORD
000052  008A    AFC0 FF7A                                   STB       2,D+5
000053  008C    D3C0 0000           P                       LNJ       5,JIO
000054                                            *           PUT NEW TEXT ("READY ASCII DECK FOR READ$");
000055  008E    FBC0 FF75                                   LAB       7,D+4
000056  0090    9BC0 FF7A                                   LAB       1,T6
000057  0092    9F87                                        STB       1,$B7
000058  0093    D3C0 0000           P                       LNJ       5,VKPNT
000059  0095    D3C0 0000           P                       LNJ       5,JWR
000060                                            *           WAIT FOR RETURN;
000061                                            *           CHECK STATE (SHOULD BE ON);
000062  0097    FBC0 FF6C                                   LAB       7,D+4
000063  0099    6C01                                        LDV       6,1
000064  009A    EF07                                        STR       6,$B7
000065  009B    D3C0 0000           P                       LNJ       5,JCS
000066                                            *           'IF' ID 'EQ' 'HEX' (2008)
000067                                            *           'THEN'
000068  009D    E840 0000           P                       LDR       6,ID
000069  009F    E970 2008                                   CMR       6,=8200
000070  00A1    098F                                        BNE       >L7
000071  00A2    FBC0 FF61                                   LAB       7,D+4
000072  00A4    9BC0 0000           P                       LAB       1,SETTOA
000073  00A6    9F87                                        STB       1,$B7
000074                                            *               INOUT (SET TO ASCII,OUTPUT CONFIGURATION @0@)
000075                                            *           'ELSE'
000076  00A7    0F00 000B           X       N1      NOP       <IOWORD+11
000077  00A9    C840 FFFE                                   LDR       4,N1+1
000078  00AB    CF40 FF59                                   STR       4,D+5
000079  00AD    D3C0 0000           P                       LNJ       5,JIO
000080  00AF    0F8E                                        B         >L8
000081  00B0    FBC0 FF53           P       L7              LAB       7,D+4
000082  00B2    9BC0 0000           P                       LAB       1,ASEJCT
000083  00B4    9F87                                        STB       1,$B7
000084  00B5    0F00 0005           X       N2              NOP       <IOWORD+5
000085  00B7    E840 FFFE                                   LDR       6,N2+1
000086  00B9    EF40 FF4B                                   STR       6,D+5
000087  00BB    D3C0 0000           P                       LNJ       5,JIO
000088                                            *               INOUT (ASCII ERROR EJECT,OUTPUT TASK @0@);
000089          00BD                              L8        EQU       $
```

```
000090                                        *          STATUS SHOULD BE := 'HEX' (8000);
000091                                        **  WAIT 5 SECONDS MAXIMUM FOR FIRST CARD TO RUPT;
000092  00BD  E870 8000                             LDR       6,=-32768
000093  00BF  EF40 0000       P                    STR       6,STATUS
000094                                        *          ZHRTCC := HERTZ * 5;
000095  00C1  C840 0000       P                    LDR       4,HERTZ
000096  00C3  4F05                                 MLV       4,5
000097  00C4  CF40 0000       P                    STR       4,ZHRTCC
000098                                        *          ZHRTCI := HERTZ * 5;
000099  00C6  D840 0000       P                    LDR       5,HERTZ
000100  00C8  5F05                                 MLV       5,5
000101  00C9  DF40 0000       P                    STR       5,ZHRTCI
000102                                        *          ZHRTCL := 20;
000103  00CB  7C14                                 LDV       7,20
000104  00CC  FF40 0000       P                    STR       7,ZHRTCL
000105                                        *          CLOCK SEMAPHORE := 0;
000106  00CE  8740 0000       P                    CL        CKSEM
000107                                        *          L21 SEMAPHORE := 0;
000108                                        **  SET DEVICE TO LEVEL 21;
000109  00D0  8740 0000       P                    CL        L21SEM
000110                                        *          I := SREG 'MASK' 'HEX' (03C0) 'UNION' 21;
000111  00D2  9840 0000       P                    LDR       1,SREG
000112  00D4  9570 03C0                           AND       1,=960
000113  00D6  9470 0015                           OR        1,=21
000114  00D8  9F40 FF28                           STR       1,D+1
000115  00DA  FBC0 FF29                           LAB       7,D+4
000116  00DC  9BC0 FF24                           LAB       1,D+1
000117  00DE  9F87                                STB       1,$B7
000118                                        *          INOUT (I,OUTPUT INTERRUPT @0@);
000119  00DF  0F00 0002       X      N3           NOP       <IOWORD+2
000120  00E1  A840 FFFE                           LDR       2,N3+1
000121  00E3  AF40 FF21                           STR       2,D+5
000122  00E5  D3C0 0000       P                    LNJ       5,JIO
000123                                        *          READ STATUS := FALSE;
000124  00E7  8740 0000       P                    CL        READST
000125  00E9  FBC0 FF1A                           LAB       7,D+4
000126  00EB  9BC0 0000       P                    LAB       1,BUFF1
000127  00ED  9F87                                STB       1,$B7
000128  00EE  0F00 0007       X      N4           NOP       <IOWORD+7
000129  00F0  E840 FFFE                           LDR       6,N4+1
000130  00F2  EF40 FF12                           STR       6,D+5
000131                                        *          IOLD (BUFFER @0@,READ @0@,NUMBER OF COLUMNS);
000132  00F4  C840 0000       P                    LDR       4,NUMBER
000133  00F6  CF40 FF0F                           STR       4,D+6
000134  00F8  D3C0 0000       P                    LNJ       5,JIL
000135  00FA  0004                                DC        4
000136                                        *          CLOCK ON;
000137  00FB  FBC0 FF08                           LAB       7,D+4
000138  00FD  9BC0 0000       P                    LAB       1,STATWA
000139  00FF  9F87                                STB       1,$B7
000140                                        *          INOUT (STATUS WAS, INPUT STATUS WORD@0@);
000141                                        *        WAIT:
000142  0100  0F00 000C       X      N5           NOP       <IOWORD+12
000143  0102  E840 FFFE                           LDR       6,N5+1
000144  0104  EF40 FF00                           STR       6,D+5
000145  0106  D3C0 0000       P                    LNJ       5,JIO
000146        0108                        L9      EQU       $
000147                                        *             'IF' L21 SEMAPHORE 'EQ' 0
000148                                        *             'THEN'
000149                                        *             'BEGIN'
000150  0108  E840 0000       P                    LDR       6,L21SEM
000151  010A  6993                                BNEZ      6,>L10
000152                                        *                       'IF' CLOCK SEMAPHORE 'EQ' 0
000153                                        *                       'THEN'
000154                                        *                          'GO TO' WAIT;
000155  010B  C840 0000       P                    LDR       4,CKSEM
000156  010D  497B                                BEZ       4,>L9
000157                                        *                       LABEL 2:= 'LITERAL' (R1);
000158  010E  D870 5231                           LDR       5,=21041
000159  0110  DF40 0000       P                    STR       5,LABEL2
000160  0112  D3C0 0000       P                    LNJ       5,JRE
000161                                        *                       REPORT ERROR;
000162  0114  FBC0 FEEF                           LAB       7,D+4
000163  0116  9BC0 FF02                           LAB       1,T12
000164  0118  9F87                                STB       1,$B7
000165  0119  D3C0 0000       P                    LNJ       5,VKPT
000166                                        *                       PUT TEXT (", DIDN'T RUPT$");
000167                                        *                       'GO TO' CONTINUE;
000168  011B  0F81 0001                           B         L10
000169                                        *             'END';
000170                                        *          CONTINUE:
000171        011D                        L10     EQU       $
000172  011D  0005                                DC        5
000173                                        *          CLOCK OFF;
000174  011E  D3C0 0000       P                    LNJ       5,JSE
000175                                        *          CHECK STATUS FOR ERRORS;
000176                                        *             'IF' 'BITS' @1,15@ STATUS WAS 'NE' 1
000177                                        *             'THEN'
000178                                        *                'GO TO' RESTART PROCEDURE;
000179                                        **  MEMORY MODULE, MEMORY BYTE ADDRESS CORRECT ?;
000180                                        *             'BEGIN'
000181  0120  E840 0000       P                    LDR       6,STATWA
000182  0122  6881 FF5C                           BGEZ      6,L5
000183                                        *             'INTEGER' MEMORY MODULE,MEMORY BYTE,ADDRESS 15;
000184  0124  FBC0 FEE2                           LAB       7,D+7
000185  0126  9BC0 FEDE                           LAB       1,D+5
000186  0128  9F87                                STB       1,$B7
000187                                        *                       INOUT (MEMORY BYTE, INPUT MEMORY BYTE @0@);
000188  0129  0F00 0006       X      N6           NOP       <IOWORD+6
000189  012B  C840 FFFE                           LDR       4,N6+1
000190  012D  CF40 FEDA                           STR       4,D+8
000191  012F  D3C0 0000       P                    LNJ       5,JIO
000192  0131  FBC0 FED5                           LAB       7,D+7
000193  0133  9BC0 FED0                           LAB       1,D+4
000194  0135  9F87                                STB       1,$B7
000195  0136  0F00 0008       X      N7           NOP       <IOWORD+8
000196  0138  E840 FFFE                           LDR       6,N7+1
000197  013A  EF40 FECD                           STR       6,D+8
000198  013C  D3C0 0000       P                    LNJ       5,JIO
000199                                        *                       INOUT (MEMORY MODULE, INPUT MODULE ADDRESS @0@);
000200                                        *                       'IF' 'BITS' @1,0@ MEMORY MODULE 'EQ' 1
000201                                        *                       'THEN'
000202  013E  E840 FEC5                           LDR       6,D+4
```

```
000203 0140 6B09                          BEVN    6,>L15
000204                              *                 ADDRESS IS := 'HEX'(8000)'UNION''BITS'@15,1@MEMORY BYTE
000205                              *               'ELSE'
000206 0141 C840 FEC3                     LDR     4,D+5
000207 0143 4041                          SOR     4,1
000208 0144 C470 8000                     OR      4,=-32768
000209 0146 CF40 FEBF                     STR     4,D+6
000210 0148 0F86                          B       >L16
000211      0149                  L15     EQU     $
000212                              *                 ADDRESS IS := 'BITS' @15,1@ MEMORY BYTE;
000213 0149 E840 FEBB                     LDR     6,D+5
000214 014B 6041                          SOR     6,1
000215 014C EF40 FEB9                     STR     6,D+6
000216      014E                  L16     EQU     $
000217                              *               'IF' 'LOCATION'(BUFFER1@NUMBER OF COLUMNS / 2@)
000218                              *               'NE' ADDRESS IS
000219 014E E840 0000       P             LDR     6,NUMBER
000220 0150 6061                          SAR     6,1
000221                              *               'THEN'
000222                              *               'BEGIN'
000223 0151 0F00 0000       X     N8      NOP     <BUFF1
000224 0153 EA40 FFFE                     ADD     6,N8+1
000225 0155 E940 FEB0                     CMR     6,D+6
000226 0157 0901 0012                     BE      L17
000227                              *                 LABEL 2 := 'LITERAL' (MA);
000228 0159 E870 4D41                     LDR     6,=19777
000229 015B EF40 0000       P             STR     6,LABEL2
000230 015D D840 0000       P             LDR     $R5,SETTOA
000231 015F C840 FEA6                     LDR     $R4,D+6
000232 0161 D3C0 0000       P             LNJ     5,JRE
000233                              *                 REPORT ERROR;
000234 0163 FBC0 FEA3                     LAB     7,D+7
000235 0165 9BC0 FEBB                     LAB     1,T18
000236 0167 9F87                          STB     1,$B7
000237 0168 D3C0 0000       P             LNJ     5,VKPT
000238                              *                 PUT TEXT (", MEMORY ADDRESS$");
000239                              *               'END';
000240 016A FBC0 FE9C               L17   LAB     7,D+7
000241 016C 9BC0 FE96                     LAB     1,D+3
000242 016E 9F87                          STB     1,$B7
000243                              *               INOUT (RANGE, INPUT RANGE @0@);
000244 016F 0F00 0009       X     N9      NOP     <IOWORD+9
000245 0171 E840 FFFE                     LDR     6,N9+1
000246 0173 EF40 FE94                     STR     6,D+8
000247 0175 D3C0 0000       P             LNJ     5,JIO
000248                              *               'IF' RANGE 'NE' 0
000249                              *               'THEN'
000250                              *               'BEGIN'
000251 0177 E840 FE8B                     LDR     6,D+3
000252 0179 6901 0019                     BEZ     6,L19
000253                              *                 LABEL 2 := 'LITERAL' (RE);
000254 017B C870 5245                     LDR     4,=21061
000255 017D CF40 0000       P             STR     4,LABEL2
000256 017F D840 0000       P             LDR     $R5,SETTOA
000257 0181 C840 FE81                     LDR     $R4,D+3
000258                              *                 LOAD R4 R5 (RANGE, CLEAR);
000259 0183 D3C0 0000       P             LNJ     5,JRE
000260                              *                 REPORT ERROR;
000261 0185 FBC0 FE81                     LAB     7,D+7
000262 0187 9BC0 FEA3                     LAB     1,T20
000263 0189 9F87                          STB     1,$B7
000264 018A D3C0 0000       P             LNJ     5,VKPT
000265                              *                 PUT TEXT (", RANGE ERROR $");
000266 018C FBC0 FE7A                     LAB     7,D+7
000267 018E E840 FE74                     LDR     6,D+3
000268 0190 EF07                          STR     6,$B7
000269 0191 D3C0 0000       P             LNJ     5,JPN
000270                              *                 PUT NUMBER (RANGE);
000271                              *               'END';
000272                              *           'END';
000273                              **  IF MARK SENSE THEN DATA ON 2 CARDS;
000274      0193                  L19     EQU     $
000275                              *           'IF' 'BITS' @1,12@ PUESDO CARD STATUS 'EQ' 1
000276                              *           'THEN'
000277                              *           'BEGIN'
000278 0193 82C0 0000       P             LB      PUESDO,2'1000'
       0195 1000
000279 0196 05AB                          BBF     >L21
000280                              *           'INTEGER' TEMP;
000281                              *           ERROR LABEL := 'LITERAL'(N2);
000282                              **  40 COLUMN ?;
000283 0197 E870 4E32                     LDR     6,=20018
000284 0199 EF40 0000       P             STR     6,LABEL1
000285                              *           'IF' 'BITS' @1,11@ PUESDO CARD STATUS 'EQ' 1
000286                              *           'THEN'
000287 019B 82C0 0000       P             LB      PUESDO,2'0800'
       019D 0800
000288 019E 0587                          BBF     >L22
000289                              *                 TEMP := 20 * ZVDAF
000290                              *           'ELSE'
000291 019F C840 0000       P             LDR     4,ZV$AF
000292 01A1 4F14                          MLV     4,20
000293 01A2 CF40 FE61                     STR     4,D+4
000294 01A4 0F86                          B       >L23
000295 01A5 E840 0000       P     L22     LDR     6,ZV$AF
000296 01A7 6F28                          MLV     6,40
000297 01A8 EF40 FE5B                     STR     6,D+4
000298                              *                 TEMP := 40 * ZVDAF;
000299      01AA                  L23     EQU     $
000300 01AA 0F00 0000       X     N10     NOP     <BUFF1
000301 01AC E840 FFFE                     LDR     6,N10+1
000302 01AE EA40 FE55                     ADD     6,D+4
000303 01B0 FBC0 FE54                     LAB     7,D+5
000304 01B2 EF07                          STR     6,$B7
000305 01B3 0F00 0007       X     N11     NOP     <IOWORD+7
000306 01B5 C840 FFFE                     LDR     4,N11+1
000307 01B7 CF40 FE4E                     STR     4,D+6
000308                              *               IOLD (BUFFER1 @TEMP@,READ @0@, NUMBER OF COLUMNS);
000309 01B9 D840 0000       P             LDR     5,NUMBER
000310 01BB DF40 FE4B                     STR     5,D+7
000311 01BD D3C0 0000       P             LNJ     5,JIL
000312 01BF D3C0 0000       P             LNJ     5,JSE
000313                              *               CHECK STATUS FOR ERRORS;
```

```
000314                                  *           'END';
000315                                  **  80 COLUMN MARK SENSE;
000316          01C1                    L21         EQU         $
000317                                  *              'IF' 'BITS' @2,11@ PUESDO CARD STATUS 'EQ' 'HEX' (2)
000318                                  *              'THEN'
000319                                  *              'BEGIN'
000320  01C1  E2C0 0000       P                      LLH         6,PUESDO
000321  01C3  6055                                   SCR         6,5
000322  01C4  604E                                   SOR         6,14
000323  01C5  6D02                                   CMV         6,2
000324  01C6  0981 0046                              BNE         L24
000325                                  *                        'INTEGER' DATA RIGHT BYTE;
000326                                  **  BLANK STROBE / DATA OR DATA / BLANK STROBE;
000327                                  *                        'IF' 'BITS' @8,0@ BUFFER1@0@ 'EQ' 'HEX' (20)
000328                                  *                        'THEN'
000329                                  *                           DATA RIGHT BYTE := FALSE
000330  01C8  C840 0000       P                      LDR         4,BUFF1
000331  01CA  C2D4                                   LLH         4,=$R4
000332  01CB  4D20                                   CMV         4,32
000333  01CC  0984                                   BNE         >L25
000334                                  *                        'ELSE'
000335  01CD  8740 FE36                              CL          D+4
000336  01CF  0F84                                   B           >L26
000337          01D0                    L25         EQU         $
000338                                  *                           DATA RIGHT BYTE := TRUE;
000339  01D0  6C01                                   LDV         6,1
000340  01D1  EF40 FE32                              STR         6,D+4
000341          01D3                    L26         EQU         $
000342          01D3                    L27         EQU         $
000343                                  *                        'FOR' I:=0 'STEP' 1 'UNTIL' 39
000344  01D3  8740 FE2D                              CL          D+1
000345                                  *                        'DO'
000346                                  *                        'BEGIN'
000347  01D5  E840 FE2B               L28            LDR         6,D+1
000348  01D7  6ED9                                   ADV         6,-39
000349  01D8  6A35                                   BGZ         6,>L24
000350                                  *                           'INTEGER' TEMP;
000351                                  **  DELETE  BLANKS AND COMPRESS DATA;
000352                                  *                           'IF' DATA RIGHT BYTE 'EQ' TRUE
000353                                  *                           'THEN'
000354                                  *                           'BEGIN'
000355  01D9  E840 FE2A                              LDR         6,D+4
000356  01DB  6D01                                   CMV         6,1
000357  01DC  0997                                   BNE         >L30
000358  01DD  C840 FE23                              LDR         4,D+1
000359  01DF  4001                                   SOL         4,1
000360                                  *                              TEMP := 'BITS' @8,0@ BUFFER @1*2@ * 256;
000361  01E0  9854                                   LDR         1,=$R4
000362  01E1  D810 0000       X                      LDR         5,<BUFF1.$R1
000363  01E3  5008                                   SOL         5,8
000364  01E4  DF40 FE20                              STR         5,D+5
000365                                  *                              BUFFER@I@;=TEMP'UNION''BITS'@8,0@BUFFER@1*2+1@;
000366                                  *                           'END'
000367  01E6  A840 FE1A                              LDR         2,D+1
000368  01E8  2001                                   SOL         2,1
000369  01E9  2E01                                   ADV         2,1
000370  01EA  F820 0000       X                      LDR         7,<BUFF1.$R2
000371  01EC  F2D7                                   LLH         7,=$R7
000372  01ED  D457                                   OR          5,=$R7
000373  01EE  B840 FE12                              LDR         3,D+1
000374  01F0  DF30 0000       X                      STR         5,<BUFF1.$R3
000375                                  *                           'ELSE'
000376                                  *                           'BEGIN'
000377  01F2  0F98                                   B           >L31
000378  01F3  E840 FE0D               L30            LDR         6,D+1
000379  01F5  6001                                   SOL         6,1
000380                                  *                              TEMP := 'BITS' @8,8@ BUFFER @1*2@ *256;
000381  01F6  9856                                   LDR         1,=$R6
000382  01F7  C810 0000       X                      LDR         4,<BUFF1.$R1
000383  01F9  C570 FF00                              AND         4,=Z'FF00'
000384  01FB  CF40 FE09                              STR         4,D+5
000385                                  *                              BUFFER@I@;=TEMP'UNION''BITS'@8,8@BUFFER@1*2+1@;
000386  01FD  D840 FE03                              LDR         5,D+1
000387  01FF  5001                                   SOL         5,1
000388  0200  5E01                                   ADV         5,1
000389  0201  A855                                   LDR         2,=$R5
000390  0202  F820 0000       X                      LDR         7,<BUFF1.$R2
000391  0204  7048                                   SOR         7,8
000392  0205  C457                                   OR          4,=$R7
000393  0206  B840 FDFA                              LDR         3,D+1
000394  0208  CF30 0000       X                      STR         4,<BUFF1.$R3
000395                                  *                           'END';
000396                                  *                        'END';  (OF FOR LOOP)
000397  020A  8AC0 FDF6               L31            INC         D+1
000398  020C  0FC9                                   B           >L28
000399                                  *           'END';          (OF IF STATEMENT)
000400
000401          020D                    L24         EQU         $
000402  020D  8740 FDF3               L32            CL          D+1
000403                                  *              'FOR' I:=0  'STEP' 1 'UNTIL' 39 'DO'
000404                                  *              'BEGIN'
000405  020F  E840 FDF1               L33            LDR         6,D+1
000406  0211  6ED9                                   ADV         6,-39
000407  0212  6A01 0047                              BGZ         6,L34
000408                                  *                        'INTEGER' TEMP BUFFER,TEMP CHAR,COLUMN;
000409                                  *                        'IF' BUFFER @I@ 'NE' CHAR @I@
000410                                  *                        'THEN'
000411                                  *                        'BEGIN'
000412  0214  9840 FDEC                              LDR         1,D+1
000413  0216  E810 0000       X                      LDR         6,<BUFF1.$R1
000414  0218  E910 0000       X                      CMR         6,<CHAR.$R1
000415  021A  093C                                   BE          >L35
000416                                  *                              TEMP BUFFER := 'BITS' @8,8@ BUFFER@I@;
000417  021B  6048                                   SOR         6,8
000418  021C  EF40 FDE7                              STR         6,D+4
000419                                  *                              TEMP CHAR   := 'BITS' @8,8@ CHAR@I@;
000420  021E  C810 0000       X                      LDR         4,<CHAR.$R1
000421  0220  4048                                   SOR         4,8
000422  0221  CF40 FDE3                              STR         4,D+5
000423  0223  1E01                                   ADV         1,1
000424  0224  1001                                   SOL         1,1
000425  0225  1EFF                                   ADV         1,-1
000426  0226  D2C0 0000       P                      LLH         5,PUESDO
```

```
000427  0228  5055                          SCR     5,5
000428  0229  504F                          SUR     5,15
000429  022A  5E01                          ADV     5,1
000430                        *                       COLUMN :=((I+1)*2-1)*('BITS' @1,12@PUESDOCARDSTATUS+1);
000431  022B  9B55                          MUL     1,=$R5
000432  022C  9F40 FDD9                     STR     1,D+6
000433  022E  FBC0 FDD8                     LAB     7,D+7
000434  0230  EF07                          STR     6,$B7
000435  0231  CF40 FDD6                     STR     4,D+8
000436                        *                       COMPARE ASCII (TEMP BUFFER,TEMP CHAR,COLUMN);
000437  0233  9F40 FDD5                     STR     1,D+9
000438  0235  D3C0 0000       P            LNJ     5,JCA
000439                        *                       TEMP BUFFER := 'BITS' @0,0@ BUFFER@1@;
000440  0237  9840 FDC9                     LDR     1,D+1
000441  0239  E810 0000       X            LDR     6,<BUFF1.$R1
000442  023B  E2D6                          LLH     6,=$R6
000443  023C  EF40 FDC7                     STR     6,D+4
000444                        *                       TEMP CHAR   := 'BITS' @0,0@ CHAR@1@;
000445  023E  C810 0000       X            LDR     4,<CHAR.$R1
000446  0240  C2D4                          LLH     4,=$R4
000447  0241  CF40 FDC3                     STR     4,D+5
000448                        *                       COLUMN :=((I+1)*2)*('BITS' @1,12@PUESDO CARD STATUS+1);
000449  0243  1E01                          ADV     1,1
000450  0244  D2C0 0000       P            LLH     5,PUESDO
000451  0246  5055                          SCR     5,5
000452  0247  504F                          SUR     5,15
000453  0248  5E01                          ADV     5,1
000454  0249  9B55                          MUL     1,=$R5
000455  024A  1001                          SUL     1,1
000456  024B  9F40 FDBA                     STR     1,D+6
000457  024D  FBC0 FDB9                     LAB     7,D+7
000458  024F  EF07                          STR     6,$B7
000459  0250  CF40 FDB7                     STR     4,D+8
000460                        *                       COMPARE ASCII (TEMP BUFFER,TEMP CHAR,COLUMN);
000461  0252  9F40 FDB6                     STR     1,D+9
000462  0254  D3C0 0000       P            LNJ     5,JCA
000463                        *                     'END';
000464                        *                 'END';
000465  0256  8AC0 FDAA                L35  INC     D+1
000466  0258  0F81 FFB6                     B       L33
000467        025A                      L34  EQU     $
000468                        *                 ERROR LABEL := 'LITERAL' (N3);
000469  025A  E870 4E33                     LDR     6,=20019
000470  025C  EF40 0000       P            STR     6,LABEL1
000471                        *                 STATUS SHOULD BE := 'HEX' (8080);
000472  025E  C870 8080                     LDR     4,=-32640
000473  0260  CF40 0000       P            STR     4,STATUS
000474  0262  FBC0 FDA1                     LAB     7,D+4
000475  0264  9BC0 0000       P            LAB     1,BUFF1
000476  0266  9F87                          STB     1,$B7
000477  0267  0F00 0007       X      N12   NOP     <IOWORD+7
000478  0269  D840 FFFE                     LDR     5,N12+1
000479  026B  DF40 FD99                     STR     5,D+5
000480                        *                 TOLD (BUFFER @0@,READ @0@,NUMBER OF COLUMNS);
000481  026D  F840 0000       P            LDR     7,NUMBER
000482  026F  FF40 FD96                     STR     7,D+6
000483  0271  D3C0 0000       P            LNJ     5,JIL
000484                        *                 'IF' ID 'NE' 'HEX' (2008)
000485                        *                 'THEN'
000486                        *                 'BEGIN'
000487  0273  E840 0000       P            LDR     6,ID
000488  0275  E970 2008                     CMR     6,=8200
000489  0277  090C                          BE      >L36
000490                        *                       STATUS SHOULD BE := 'HEX' (9000);
000491  0278  C870 9000                     LDR     4,=-28672
000492  027A  CF40 0000       P            STR     4,STATUS
000493                        *                       ASCII BIT := 'BITS' @1,12@ STATUS WAS;
000494                        *                 'END'
000495  027C  D2C0 0000       P            LLH     5,STATWA
000496  027E  5055                          SCR     5,5
000497  027F  504F                          SUR     5,15
000498  0280  DF40 FD81                     STR     5,D+2
000499                        *                 'ELSE'
000500  0282  0F87                          B       >L37
000501        0283                      L36  EQU     $
000502                        *                       ASCII BIT := 'BITS' @1,7@ STATUS WAS;
000503  0283  E840 0000       P            LDR     6,STATWA
000504  0285  E2D6                          LLH     6,=$R6
000505  0286  6047                          SUR     6,7
000506  0287  EF40 FD7A                     STR     6,D+2
000507  0289  D3C0 0000       P      L37   LNJ     5,JSE
000508                        *                 CHECK STATUS FOR ERRORS;
000509                        *                 'IF' ASCII BIT 'EQ' 0
000510                        *                 'THEN'
000511  028B  E840 FD76                     LDR     6,D+2
000512  028D  6988                          BNEZ    6,>L38
000513                        *                       PUT NEW TEXT ("ILLEGAL ASCII NOT READ$");
000514  028E  FBC0 FD75                     LAB     7,D+4
000515  0290  9BC0 FDA3                     LAB     1,T39
000516  0292  9F87                          STB     1,$B7
000517  0293  D3C0 0000       P            LNJ     5,VKPNT
000518        0295                      L38  EQU     $
000519                        *                 STATUS SHOULD BE := 'HEX' (8000);
000520  0295  E870 8000                     LDR     6,=-32768
000521  0297  EF40 0000       P            STR     6,STATUS
000522                        *                 ERROR LABEL := 'LITERAL' (N4);
000523                        **  READ CARD IN BINARY MODE;
000524  0299  C870 4E34                     LDR     4,=20020
000525  029B  CF40 0000       P            STR     4,LABEL1
000526  029D  FBC0 FD66                     LAB     7,D+4
000527  029F  9BC0 0000       P            LAB     1,BINARY
000528  02A1  9F87                          STB     1,$B7
000529                        *                 INOUT (INITIALIZE,OUTPUT CONTROL @0@);
000530  02A2  ABC0 0000       P            LAB     2,IOWORD
000531  02A4  AFC0 FD60                     STB     2,D+5
000532  02A6  D3C0 0000       P            LNJ     5,JIO
000533                        *                 'IF' ID 'EQ' 'HEX' (2008)
000534                        *                 'THEN'
000535  02A8  E840 0000       P            LDR     6,ID
000536  02AA  E970 2008                     CMR     6,=8200
000537  02AC  098F                          BNE     >L40
000538  02AD  FBC0 FD56                     LAB     7,D+4
000539  02AF  9BC0 0000       P            LAB     1,BINARY
```

```
000540  02B1  9F87                          *              STB     1,$B7
000541                                      *           INOUT (BINARY MODE,OUTPUT CONFIGURATION @0@)
000542                                      *              'ELSE'
000543  02B2  0F00 000B       X      N13    NOP     <10WORD+11
000544  02B4  C840 FFFE                     LDR     4,N13+1
000545  02B6  CF40 FD4E                     STR     4,D+5
000546  02B8  D3C0 0000       P             LNJ     5,J10
000547  02BA  0F8E                          B       >L41
000548  02BB  FBC0 FD48              L40    LAB     7,D+4
000549  02BD  9BC0 0000       P             LAB     1,SETTOH
000550  02BF  9F87                          STB     1,$B7
000551  02C0  0F00 0005       X      N14    NOP     <10WORD+5
000552  02C2  E840 FFFE                     LDR     6,N14+1
000553  02C4  EF40 FD40                     STR     6,D+5
000554  02C6  D3C0 0000       P             LNJ     5,J10
000555                                      *           INOUT (SET TO HOLLERITH,OUTPUT TASK @0@);
000556  02C8  FBC0 FD3B              L41    LAB     7,D+4
000557  02CA  9BC0 0000       P             LAB     1,BUFF1
000558  02CC  9F87                          STB     1,$B7
000559  02CD  6C54                          LDV     6,84
000560  02CE  EF40 FD36                     STR     6,D+5
000561  02D0  4C56                          LDV     4,86
000562  02D1  CF40 FD34                     STR     4,D+6
000563                                      *           FILL BUFFER (BUFFER 1,86,'HEX'(FFFF));
000564                                      **  CHECK TO SEE IF MORE THEN A CARD OF DATA IS TRANSFERED;
000565  02D3  5CFF                          LDV     5,-1
000566  02D4  DF40 FD32                     STR     5,D+7
000567  02D6  D3C0 0000       P             LNJ     5,JFB
000568  02D8  FBC0 FD2B                     LAB     7,D+4
000569  02DA  9BC0 0000       P             LAB     1,BUFF1
000570  02DC  9F87                          STB     1,$B7
000571  02DD  0F00 0007       X      N15    NOP     <10WORD+7
000572  02DF  E840 FFFE                     LDR     6,N15+1
000573  02E1  EF40 FD23                     STR     6,D+5
000574                                      *           IOLD (BUFFER1 @0@,READ @0@,172);
000575  02E3  C870 00AC                     LDR     4,=172
000576  02E5  CF40 FD20                     STR     4,D+6
000577  02E7  D3C0 0000       P             LNJ     5,J1L
000578                                      *              'IF' BUFFER@NUMBER OF COLUMNS@ 'NE' 'HEX' (FFFF)
000579                                      *              'THEN'
000580                                      *              'BEGIN'
000581  02E9  9840 0000       P             LDR     1,NUMBER
000582  02EB  E810 0000       X             LDR     6,<BUFF1.$R1
000583  02ED  6DFF                          CMV     6,-1
000584  02EE  090E                          BE      >L42
000585                                      *              LABEL 2:= 'LITERAL' (MW);
000586  02EF  C870 4D57                     LDR     4,=19799
000587  02F1  CF40 0000       P             STR     4,LABEL2
000588  02F3  D3C0 0000       P             LNJ     5,JRE
000589                                      *              REPORT ERROR;
000590  02F5  FBC0 FD0E                     LAB     7,D+4
000591  02F7  9BC0 FD49                     LAB     1,T43
000592  02F9  9F87                          STB     1,$B7
000593  02FA  D3C0 0000       P             LNJ     5,VKPT
000594                                      *              PUT TEXT (", XFER TO MANY$");
000595                                      *              'END';
000596                                      ** RANGE CORRECT  ";
000597  02FC  FBC0 FD07              L42    LAB     7,D+4
000598  02FE  9BC0 FD04                     LAB     1,D+3
000599  0300  9F87                          STB     1,$B7
000600                                      *           INOUT(RANGE, INPUT RANGE @0@);
000601  0301  0F00 0009       X      N16    NOP     <10WORD+9
000602  0303  E840 FFFE                     LDR     6,N16+1
000603  0305  EF40 FCFF                     STR     6,D+5
000604  0307  D3C0 0000       P             LNJ     5,J10
000605                                      *              'IF' RANGE 'NE' 172-(NUMBER OF COLUMNS)*2
000606                                      *              'THEN'
000607                                      *              'BEGIN'
000608  0309  E840 0000       P             LDR     6,NUMBER
000609  030B  6001                          SOL     6,1
000610  030C  E270 00AC                     SUB     6,=172
000611  030E  8256                          NEG     =$R6
000612  030F  E940 FCF3                     CMR     6,D+3
000613  0311  0901 0012                     BE      L44
000614                                      *              LABEL 2:= 'LITERAL' (RE);
000615  0313  E870 5245                     LDR     6,=21061
000616  0315  EF40 0000       P             STR     6,LABEL2
000617  0317  D840 0000       P             LDR     $R5,SETTOA
000618  0319  C840 FCE9                     LDR     $R4,D+3
000619                                      *              LOAD R4 R5 (RANGE, CLEAR);
000620  031B  D3C0 0000       P             LNJ     5,JRE
000621                                      *              REPORT ERROR;
000622  031D  FBC0 FCE6                     LAB     7,D+4
000623  031F  9BC0 FD2A                     LAB     1,T45
000624  0321  9F87                          STB     1,$B7
000625  0322  D3C0 0000       P             LNJ     5,VKPT
000626                                      *              PUT TEXT (", RANGE WRONG$");
000627                                      *              'END';
000628        0324                    L44   EQU     $
000629                                      *              'IF' 'BITS' @1,15@ STATUS WAS 'NE' 1 'AND'
000630  0324  E840 0000       P             LDR     6,STATWA
000631  0326  6807                          BLZ     6,>L46
000632                                      *                 'BITS' @1,14@ STATUS WAS 'NE' 1
000633                                      *              'THEN'
000634                                      *              'BEGIN'
000635  0327  82C0 0000       P             LB      STATWA,2'4000'
      0329  4000
000636  032A  0503                          BBT     >L46
000637  032B  D3C0 0000       P             LNJ     5,JSE
000638                                      *              CHECK STATUS FOR ERRORS;
000639                                      *              'END';
000640        032D                    L46   EQU     $
000641                                      *              'IF' ID 'NE' 'HEX'(2008)
000642                                      *              'THEN'
000643                                      *              'BEGIN'
000644                                      *           TRY AGAIN;
000645  032D  E840 0000       P             LDR     6,ID
000646  032F  E970 2008                     CMR     6,=8200
000647  0331  092D                          BE      >L47
000648        0332                    L48   EQU     $
000649                                      *              ASK NEW QUESTION ("SECOND CARD OFFSET STACKED$");
000650  0332  FBC0 FCD1                     LAB     7,D+4
000651  0334  9BC0 FD1D                     LAB     1,T49
```

```
000652 0336 9F87                          STB   1,$B7
000653 0337 D3C0 0000      P              LNJ   5,VKANQ
000654                           *              I := 'HEX' (590D);
000655 0339 E870 590D                     LDR   6,=22797
000656 033B EF40 FCC5                     STR   6,D+1
000657                           *              GET CHAR (I);
000658 033D FBC0 FCC6                     LAB   7,D+4
000659 033F 9BC0 FCC1                     LAB   1,D+1
000660 0341 9F87                          STB   1,$B7
000661 0342 D3C0 0000      P              LNJ   5,VKGC
000662                           *              'IF' I 'NE' 'LITERAL' (Y)
000663                           *              'THEN'
000664                           *              'BEGIN'
000665 0344 E840 FCBC                     LDR   6,D+1
000666 0346 6D59                          CMV   6,89
000667 0347 0910                          BE    >L50
000668                           *                   'IF' I 'NE' 'LITERAL' (N)
000669                           *                        'THEN'
000670                           *                             'GO TO' TRY AGAIN;
000671 0348 6D4E                          CMV   6,78
000672 0349 09E9                          BNE   >L48
000673                           *                        LABEL 2 := 'LITERAL' (05);
000674 034A C870 4F53                     LDR   4,=20307
000675 034C CF40 0000      P              STR   4,LABEL2
000676 034E D3C0 0000      P              LNJ   5,JRE
000677                           *                        REPORT ERROR;
000678 0350 FBC0 FCB3                     LAB   7,D+44
000679 0352 9BC0 FD0E                     LAB   1,T52
000680 0354 9F87                          STB   1,$B7
000681 0355 D3C0 0000      P              LNJ   5,VKPT
000682                           *                        PUT TEXT (", ERROR EJECT$");
000683                           *              'END';
000684 0357 FBC0 FCAC                L50  LAB   7,D+4
000685 0359 9BC0 FD0F                     LAB   1,T53
000686 035B 9F87                          STB   1,$B7
000687 035C D3C0 0000      P              LNJ   5,VKPNT
000688                           *              PUT NEW TEXT ("STOP, RUNOUT!C L!$");
000689                           *         'END';
000690                           *    END OF LETTER N:
000691                           *    'END';       (OF PROCEDURE)
000692                           *    'END';       (OF SEGMENT)
000693                           *    'FINISH'
000694 035E 83C8 FCA1                L47  JMP   *D
000695 0360                          L1   EQU   $
000696 0360 83C0 0000      P              JMP   ZK$EX
000697                                    XLOC  ZK$EX
000698                                    CTRL  LINK ZK$EX
000699                                    XLOC  ZHRTC1
000700                                    XLOC  ZHRTCL
000701                                    XLOC  ZHRTCC
000702                                    XLOC  ZV$AF
000703                                    CTRL  LINK ZV$AF
000704                                    XLOC  VKPT
000705                                    XLOC  VKPNT
000706                                    XLOC  VKANQ
000707                                    XLOC  VKGC
000708                                    XLOC  ID
000709                                    XLOC  STATWA
000710                                    XLOC  STATUS
000711                                    XLOC  HERTZ
000712                                    XLOC  PUESDO
000713                                    XLOC  SETTOH
000714                                    XLOC  SETTOA
000715                                    XLOC  ASEJCT
000716                                    XLOC  BINARY
000717                                    XLOC  READST
000718                                    XLOC  NUMBER
000719                                    XLOC  LABEL1
000720                                    XLOC  LABEL2
000721                                    XLOC  CKSEM
000722                                    XLOC  L21SEM
000723                                    XLOC  SREG
000724                                    XLOC  BUFF1
000725                                    XLOC  CHAR
000726                                    XLOC  IOWORD
000727                                    XLOC  JRE
000728                                    XLOC  JIO
000729                                    XLOC  JIL
000730                                    XLOC  JPN
000731                                    XLOC  JCA
000732                                    XLOC  JWR
000733                                    XLOC  JSE
000734                                    XLOC  JCS
000735                                    XLOC  JFB
000736 0362 0072                          END   ZVKJLN,ZVKJLN
0000  ERR COUNT
@@L
```

```
E   ZVKJLP.LIST                    05/19/78  1429.5R W 05/19/78  1422.7      4032990000
000001                                        TITLE     ZVKJLP,'REV 00'
000002          009D                          XDEF      KJLP
000003          0098                          XDEF      ZVKJLP
000004          0008                ZKCOM     COMM      8
000005  0000    0000                D         RESV      9,0
000006  0009    0019                T8        DC        25
000007  000A    5055 4E43 4820                TEXT      'PUNCH 100 CARD TES'
        000D    3130 3020 4341
                5244 2054 4553
000008  0013    5420 4445 434B                TEXT      'T DECK$'
        0016    2400
000009  0017    000F                T13       DC        15
000010  0018    5241 4E44 4F4D                TEXT      'RANDOM PUNCHES$'
        001B    2050 554E 4348
                4553 2400
000011  0020    000C                T18       DC        12
000012  0021    434F 5059 2041                TEXT      'COPY A CARDS$'
        0024    2043 4152 4424
000013  0027    0016                T24       DC        22
000014  0028    5245 4144 4552                TEXT      'READER CHANNEL NUM'
        002B    2043 4841 4E4E
                454C 204E 554D
000015  0031    4245 5224                     TEXT      'BER$'
000016  0033    000D                T32       DC        13
000017  0034    5245 4144 5920                TEXT      'READY READER$'
        0037    5245 4144 4552
                2400
000018  003B    000E                T35       DC        14
000019  003C    5255 4E4F 5554                TEXT      'RUNOUT, START$'
        003F    2C20 5354 4152
                5424
000020  0043    0030                T37       DC        48
000021  0044    454E 5445 5220                TEXT      'ENTER 3 HEX VALUES'
        0047    3320 4845 5820
                5641 4C55 4553
000022  004D    2F43 4F4C 2C20                TEXT      '/COL, RETURN ALONE ENDS INPU'
        0050    5245 5455 524E
                2041 4C4F 4E45
                2045 4E44 5320
                494E 5055
000023  005B    5424                          TEXT      'T$'
000024  005C    0001                T41       DC        1
000025  005D    2400                          TEXT      '$'
000026  005E    0012                T44       DC        18
000027  005F    4D41 5849 4D55                TEXT      'MAXIMUM CARDS ARE$'
        0062    4D20 4341 5244
                5320 4152 4524
000028  0068    0019                T45       DC        25
000029  0069    5245 4144 5920                TEXT      'READY PUNCH, TOTAL'
        006C    5055 4E43 482C
                2054 4F54 414C
000030  0072    2043 4152 4453                TEXT      ' CARDS$'
        0075    2400
000031  0076    0009                T46       DC        9
000032  0077    5055 4E43 4845                TEXT      'PUNCHED $'
        007A    4420 2400
000033  007C    001B                T49       DC        27
000034  007D    4552 524F 5220                TEXT      'ERROR CARDS OFFSET'
        0080    4341 5244 5320
                4F46 4653 4554
000035  0086    2053 5441 434B                TEXT      ' STACKED$'
        0089    4544 2400
000036  008B    000E                T52       DC        14
000037  008C    2C20 4552 524F                TEXT      ', ERROR EJECT$'
        008F    5220 454A 4543
                5424
000038  0093    0007                T54       DC        7
000039  0094    5645 5249 4659                TEXT      'VERIFY$'
        0097    2400
000040          0098                ZVKJLP    EQU       $
000041  0098    8F00 0001       K             SAVE      <ZKCOM+1,Z'0011'
        009A    0011
000042                              *       'BEGIN'
000043  009B    0F81 021D                     B         L1
000044                              *           'INTEGER' TEMP;
000045  009D    DFC0 FF62           KJLP      STB       5,D
000046                              *           'IF' ID 'EQ' 'HEX' (2008)
000047                              *           'THEN'
000048                              *           'BEGIN'
000049  009F    E840 0000       P             LDR       6,ID
000050  00A1    E970 2008                     CMR       6,=8200
000051  00A3    0988                          BNE       >L3
000052                              *                   'IF' 'BITS' @1,12@ PUESDO CARD STATUS 'EQ' 1
000053                              *                   'THEN'
000054                              *                   'GO TO' LETTER R
000055                              *                   'ELSE'
000056  00A4    82C0 0000       P             LB        PUESDO,Z'1000'
        00A6    1000
000057  00A7    0501 0000       P             BBT       LETR
000058                              *                       'GO TO' LETTER V;
000059  00A9    83C0 0000       P             JMP       LETV
000060                              *           'END';
000061          00AB                L3        EQU       $
000062                              *           LABEL 1 := 'LITERAL' (P1);
000063  00AB    E870 5031                     LDR       6,=20529
000064  00AD    EF40 0000       P             STR       6,LABEL1
000065  00AF    FBC0 FF52                     LAB       7,D+2
000066  00B1    9BC0 0000       P             LAB       1,SETIOH
000067  00B3    9F87                          STB       1,$B7
000068                              *           INOUT (SET TO HOLLERITH, OUTPUT TASK @0@);
000069  00B4    0F00 0005       X     N1      NOP       <IOWORD+5
000070  00B6    C840 FFFE                     LDR       4,N1+1
000071  00B8    CF40 FF4A                     STR       4,D+3
000072  00BA    D3C0 0000       P             LNJ       5,J10
000073                              *           STATUS SHOULD BE := 'HEX' (8000);
000074  00BC    E870 8000                     LDR       6,=-32768
000075  00BE    EF40 0000       P             STR       6,STATUS
000076  00C0    FBC0 FF41                     LAB       7,D+2
000077  00C2    9BC0 0000       P             LAB       1,BUFF1
000078  00C4    9F87                          STB       1,$B7
000079  00C5    4C54                          LDV       4,84
000080  00C6    CF40 FF3C                     STR       4,D+3
```

```
000081  00C8  5C50                          LDV       5,80
000082  00C9  DF40 FF3A                     STR       5,D+4
000083                              *           FILL BUFFER (BUFFER 1,80,'HEX'(FFF));
000084  00CB  E870 0FFF                     LDR       7,=4095
000085  00CD  FF40 FF37                     STR       7,D+5
000086  00CF  D3C0 0000       P             LNJ       5,JFB
000087                              *           RESET RANDOM DATA (CLEAR);
000088  00D1  FBC0 FF30                     LAB       7,D+2
000089  00D3  9BC0 0000       P             LAB       1,SETTOA
000090  00D5  9F87                          STB       1,$B7
000091  00D6  D3C0 0000       P             LNJ       5,VKRRD
000092                              *           ZHRTC1 := 0;
000093  00D8  8740 0000       P             CL        ZHRTC1
000094                              *           ZHRTCC := 0;
000095  00DA  8740 0000       P             CL        ZHRTCC
000096                              *           LOW ADDRESS := ZVDLR;
000097                              *       INPUT AGAIN:
000098  00DC  E840 0000       P             LDR       6,ZV$LR
000099  00DE  EF40 0000       P             STR       6,LOWADJ
000100        00E0                    L7    EQU       $
000101                              *           ASK NEW QUESTION ("PUNCH 100 CARD TEST DECK$");
000102  00E0  FBC0 FF21                     LAB       7,D+2
000103  00E2  9BC0 FF26                     LAB       1,T8
000104  00E4  9F87                          STB       1,$B7
000105  00E5  D3C0 0000       P             LNJ       5,VKANQ
000106                              *           TEMP := 'HEX' (590D);
000107  00E7  E870 590D                     LDR       6,=22797
000108  00E9  EF40 FF17                     STR       6,D+1
000109                              *           GET CHAR (TEMP);
000110  00EB  FBC0 FF16                     LAB       7,D+2
000111  00ED  9BC0 FF13                     LAB       1,D+1
000112  00EF  9F87                          STB       1,$B7
000113  00F0  D3C0 0000       P             LNJ       5,VKGC
000114                              *           'IF' TEMP 'EQ' 'LITERAL' (Y)
000115                              *           'THEN'
000116                              *           'BEGIN'
000117  00F2  E840 FF0E                     LDR       6,D+1
000118  00F4  6D59                          CMV       6,89
000119  00F5  098B                          BNE       >L9
000120                              *               TYPE OF PUNCHES := 'LITERAL' (R);
000121  00F6  4C52                          LDV       4,82
000122  00F7  CF40 0000       P             STR       4,TYPEOF
000123                              *               CARDS SPECIFIED := 100;
000124  00F9  5C64                          LDV       5,100
000125  00FA  DF40 0000       P             STR       5,CARDSS
000126  00FC  D3C0 0000       P             LNJ       5,JWR
000127                              *               WAIT FOR RETURN;
000128                              *               'GO TO' PUNCH;
000129                              *           'END'
000130  00FE  0F81 014E                     B         L10
000131                              *           'ELSE'
000132        0100                    L9    EQU       $
000133                              *               'IF' TEMP 'NE' 'LITERAL' (N)
000134                              *               'THEN'
000135                              *                   'GO TO' INPUT AGAIN;
000136                              *       TRY AGAIN:
000137  0100  E840 FF00                     LDR       6,D+1
000138  0102  6D4E                          CMV       6,78
000139  0103  09DD                          BNE       >L7
000140        0104                    L12   EQU       $
000141                              *           ASK NEW QUESTION ("RANDOM PUNCHES$");
000142  0104  FBC0 FEFD                     LAB       7,D+2
000143  0106  9BC0 FF10                     LAB       1,T13
000144  0108  9F87                          STB       1,$B7
000145  0109  D3C0 0000       P             LNJ       5,VKANQ
000146                              *           GET CHAR            (TEMP              );
000147  010B  FBC0 FEF6                     LAB       7,D+2
000148  010D  9BC0 FEF3                     LAB       1,D+1
000149  010F  9F87                          STB       1,$B7
000150  0110  D3C0 0000       P             LNJ       5,VKGC
000151                              *           'IF' TEMP 'EQ' 'LITERAL' (Y)
000152                              *           'THEN'
000153                              *             'BEGIN'
000154  0112  E840 FEEE                     LDR       6,D+1
000155  0114  6D59                          CMV       6,89
000156  0115  0986                          BNE       >L14
000157                              *               TYPE OF PUNCHES := 'LITERAL' (R);
000158                              *             'END'
000159  0116  4C52                          LDV       4,82
000160  0117  CF40 0000       P             STR       4,TYPEOF
000161                              *           'ELSE'
000162                              *             'BEGIN'
000163  0119  0F81 0115                     B         L15
000164        011B                    L14   EQU       $
000165                              *               'IF' TEMP 'NE' 'LITERAL' (N)
000166                              *               'THEN'
000167                              *                   'GO TO' TRY AGAIN;
000168                              *       NOT Y OR N:
000169  011B  E840 FEE5                     LDR       6,D+1
000170  011D  6D4E                          CMV       6,78
000171  011E  09E6                          BNE       >L12
000172        011F                    L17   EQU       $
000173                              *           LABEL 1 := 'LITERAL' (P2);
000174  011F  E870 5032                     LDR       6,=20530
000175  0121  EF40 0000       P             STR       6,LABEL1
000176  0123  FBC0 FEDE                     LAB       7,D+2
000177  0125  9BC0 FEFA                     LAB       1,T18
000178  0127  9F87                          STB       1,$B7
000179  0128  D3C0 0000       P             LNJ       5,VKANQ
000180                              *               ASK NEW QUESTION ("COPY A CARD$");
000181                              *               GET CHAR            (TEMP              );
000182  012A  FBC0 FED7                     LAB       7,D+2
000183  012C  9BC0 FED4                     LAB       1,D+1
000184  012E  9F87                          STB       1,$B7
000185  012F  D3C0 0000       P             LNJ       5,VKGC
000186                              *               'IF' TEMP 'NE' 'LITERAL' (N)
000187  0131  E840 FECF                     LDR       6,D+1
000188  0133  6D4E                          CMV       6,78
000189  0134  0903                          BE        >L19
000190                              *               'AND' TEMP 'NE' 'LITERAL' (Y)
000191                              *               'THEN'
000192                              *                   'GO TO' NOT Y OR N;
000193  0135  6D59                          CMV       6,89
```

```
000194  0136  09E9                          BNE     >L17
000195        0137              L19         EQU     $
000196                          *                   'IF' TEMP 'EQ' 'LITERAL' (Y)
000197                          *                   'THEN'
000198                          *                   'BEGIN'
000199  0137  E840 FEC9                      LDR     6,D+1
000200  0139  6D59                          CMV     6,89
000201  013A  0981 00B6                      BNE     L21
000202                          *                       'INTEGER' ID 2;
000203                          *                       ID 2 := ID;
000204  013C  C840 0000       P              LDR     4,ID
000205  013E  CF40 FEC3                      STR     4,D+2
000206                          *                       TYPE OF PUNCHES := 'LITERAL' (D);
000207  0140  5C44                          LDV     5,68
000208  0141  DF40 0000       P              STR     5,TYPEOF
000209                          *                       'IF' ID 2 'EQ' 'HEX' (2088)
000210                          *                       'THEN'
000211                          *                       'BEGIN'
000212  0143  C970 2088                      CMR     4,=8328
000213  0145  0981 0042                      BNE     L22
000214                          *                           'INTEGER' READER CHANNEL;
000215                          *                           ERROR LABEL := 'LITERAL'(P3);
000216  0147  F870 5033                      LDR     7,=20531
000217  0149  FF40 0000       P              STR     7,LABEL1
000218                          *                           READ OR PUNCH := READ @0@;
000219                          *               ASK READER QUESTION;
000220  014B  9800 0007       X              LDR     1,<IOWORD+7
000221  014D  9F40 0000       P              STR     1,READOR
000222        014F              L23         EQU     $
000223                          *                           ASK NEW QUESTION ("READER CHANNEL NUMBER$");
000224  014F  FBC0 FEB4                      LAB     7,D+4
000225  0151  9BC0 FED5                      LAB     1,T24
000226  0153  9F87                          STB     1,$B7
000227  0154  D3C0 0000       P              LNJ     5,VKANQ
000228  0156  D3C0 0000       P              LNJ     5,VKGB
000229                          *                           'IF' GET BREAK 'EQ' TRUE
000230                          *                           'THEN'
000231                          *                               'GO TO' ASK NEXT;
000232  0158  6D01                          CMV     6,1
000233  0159  0901 0000       P              BE      ASKNEX
000234  015B  FBC0 FEA8                      LAB     7,D+4
000235  015D  9BC0 FEA5                      LAB     1,D+3
000236  015F  9F87                          STB     1,$B7
000237  0160  D3C0 0000       P              LNJ     5,VKGH
000238                          *                           GET HEX          (READER CHANNEL          );
000239                          *                           'IF' READER CHANNEL 'MASK' 'HEX' (FFC0)'EQ' 0'OR'
000240  0162  E840 FEA0                      LDR     6,D+3
000241  0164  E570 FFC0                      AND     6,=-64
000242  0166  6969                          BEZ     6,>L23
000243                          *                               READER CHANNEL 'MASK' 'HEX' (003F) 'NE' 0
000244                          *                           'THEN'
000245                          *                               'GO TO' ASK READER QUESTION;
000246  0167  E840 FE9B                      LDR     6,D+3
000247  0169  E570 003F                      AND     6,=63
000248  016B  69E4                          BNEZ    6,>L23
000249                          *                           CREATE FUNCTION CODES (READER CHANNEL);
000250  016C  FBC0 FE97                      LAB     7,D+4
000251  016E  E840 FE94                      LDR     6,D+3
000252  0170  EF07                          STR     6,$B7
000253  0171  D3C0 0000       P              LNJ     5,JFC
000254  0173  FBC0 FE90                      LAB     7,D+4
000255  0175  9BC0 FE8C                      LAB     1,D+2
000256  0177  9F87                          STB     1,$B7
000257                          *                           INOUT (ID 2,INPUT ID @0@);
000258  0178  0F00 000D       X      N2      NOP     <IOWORD+13
000259  017A  E840 FFFE                      LDR     6,N2+1
000260  017C  EF40 FE88                      STR     6,D+5
000261  017E  D3C0 0000       P              LNJ     5,JIO
000262                          *                           'IF' ID 2 'NE' 'HEX' (2008) 'AND'
000263  0180  E840 FE81                      LDR     6,D+2
000264  0182  E970 2008                      CMR     6,=8200
000265  0184  0904                          BE      >L22
000266                          *                               ID 2 'NE' 'HEX' (208A)
000267                          *                           'THEN'
000268                          *                               'GO TO' ASK READER QUESTION;
000269  0185  E970 208A                      CMR     6,=8330
000270  0187  09C8                          BNE     >L23
000271                          *                       'END';
000272  0188  FBC0 FE7A              L22     LAB     7,D+3
000273  018A  9BC0 0000       P              LAB     1,BINARY
000274  018C  9F87                          STB     1,$B7
000275                          *                       INOUT (INITIALIZE,OUTPUT CONTROL @0@);
000276  018D  ABC0 0000       P              LAB     2,IOWORD
000277  018F  AFC0 FE74                      STB     2,D+4
000278  0191  D3C0 0000       P              LNJ     5,JIO
000279                          *                       'IF' ID 2 'EQ' 'HEX' (2008)
000280                          *                       'THEN'
000281  0193  E840 FE6E                      LDR     6,D+2
000282  0195  E970 2008                      CMR     6,=8200
000283  0197  098F                          BNE     >L30
000284  0198  FBC0 FE6A                      LAB     7,D+3
000285  019A  9BC0 0000       P              LAB     1,BINARY
000286  019C  9F87                          STB     1,$B7
000287                          *                           INOUT (BINARY MODE, OUTPUT CONFIGURATION @0@)
000288                          *                       'ELSE'
000289  019D  0F00 000B       X      N3      NOP     <IOWORD+11
000290  019F  C840 FFFE                      LDR     4,N3+1
000291  01A1  CF40 FE62                      STR     4,D+4
000292  01A3  D3C0 0000       P              LNJ     5,JIO
000293  01A5  0F8E                          B       >L31
000294  01A6  FBC0 FE5C              L30     LAB     7,D+3
000295  01A8  9BC0 0000       P              LAB     1,SETTOH
000296  01AA  9F87                          STB     1,$B7
000297  01AB  0F00 0005       X      N4      NOP     <IOWORD+5
000298  01AD  E840 FFFE                      LDR     6,N4+1
000299  01AF  EF40 FE54                      STR     6,D+4
000300  01B1  D3C0 0000       P              LNJ     5,JIO
000301                          *                       INOUT (SET TO HOLLERITH, OUTPUT TASK @0@);
000302  01B3  FBC0 FE4F              L31     LAB     7,D+3
000303  01B5  9BC0 FE7D                      LAB     1,132
000304  01B7  9F87                          STB     1,$B7
000305  01B8  D3C0 0000       P              LNJ     5,VKPNT
000306                          *                       PUT NEW TEXT ("READY READER$");
```

```
000307  01BA  D3C0 0000      P                 LNJ       5,JWR
000308                                  *                       WAIT FOR RETURN;
000309                                  *                       CHECK STATE (SHOULD BE ON);
000310  01BC  FBC0 FE46                         LAB       7,D+3
000311  01BE  6C01                             LDV       6,1
000312  01BF  EF07                             STR       6,$B7
000313  01C0  D3C0 0000      P                 LNJ       5,JCS
000314  01C2  FBC0 FE40                         LAB       7,D+3
000315  01C4  9BC0 0000      P                 LAB       1,BUFF1
000316  01C6  9F87                             STB       1,$B7
000317  01C7  0F00 0007      X       N5        NOP       <IOWORD+7
000318  01C9  E840 FFFE                         LDR       6,N5+1
000319  01CB  EF40 FE38                         STR       6,D+4
000320                                  *                       IOLD (BUFFER@0@,READ@0@,171);
000321  01CD  C870 00AB                         LDR       4,=171
000322  01CF  CF40 FE35                         STR       4,D+5
000323  01D1  D3C0 0000      P                 LNJ       5,JIL
000324                                  *                       'IF' STATUS WAS 'NE' PUESDO CARD STATUS
000325                                  *                            'UNION' 'HEX' (4000)
000326                                  *                            'THEN'
000327  01D3  E840 0000      P                 LDR       6,PUESDO
000328  01D5  E470 4000                         OR        6,=16384
000329  01D7  E940 0000      P                 CMR       6,STATWA
000330  01D9  0903                             BE        >L33
000331  01DA  D3C0 0000      P                 LNJ       5,JSE
000332                                  *                            CHECK STATUS ERRORS;
000333        01DC                      L33       EQU       $
000334                                  *                       'IF' ID 2 'NE' 'HEX' (2008)
000335                                  *                            'THEN'
000336  01DC  E840 FE25                         LDR       6,D+2
000337  01DE  E970 2008                         CMR       6,=8200
000338  01E0  0908                             BE        >L34
000339  01E1  FBC0 FE21                         LAB       7,D+3
000340  01E3  9BC0 FE57                         LAB       1,T35
000341  01E5  9F87                             STB       1,$B7
000342  01E6  D3C0 0000      P                 LNJ       5,VKPNT
000343                                  *                            PUT NEW TEXT ("RUNOUT, START$");
000344  01E8  FBC0 FE1A                 L34     LAB       7,D+3
000345  01EA  E840 0000      P                 LDR       6,CHANNE
000346  01EC  EF07                             STR       6,$B7
000347  01ED  D3C0 0000      P                 LNJ       5,JFC
000348                                  *                            CREATE FUNCTION CODES (CHANNEL);
000349                                  *                       'END'
000350                                  *                       'ELSE'
000351                                  *                       'BEGIN'
000352  01EF  0F81 003F                         B         L15
000353        01F1                      L21       EQU       $
000354                                  *                       'INTEGER' VALUE,I;
000355                                  *                       LABEL I:= 'LITERAL' (P4);
000356                                  *                       PUT NEW TEXT
000357  01F1  E870 5034                         LDR       6,=20532
000358  01F3  EF40 0000      P                 STR       6,LABEL1
000359                                  *                            ("ENTER 3 HEX VALUES/COL, RETURN ALONE ENDS INPUT$");
000360  01F5  FBC0 FE0E                         LAB       7,D+4
000361  01F7  9BC0 FE4B                         LAB       1,T37
000362  01F9  9F87                             STB       1,$B7
000363  01FA  D3C0 0000      P                 LNJ       5,VKPNT
000364  01FC  FBC0 FE07                         LAB       7,D+4
000365  01FE  9BC0 0000      P                 LAB       1,BUFF1
000366  0200  9F87                             STB       1,$B7
000367  0201  6C54                             LDV       6,84
000368  0202  EF40 FE02                         STR       6,D+5
000369  0204  4C50                             LDV       4,80
000370  0205  CF40 FE00                         STR       4,D+6
000371                                  *                       FILL BUFFER (BUFFER1,80,0);
000372                                  **  LOOP TO GET VALUE OF COLUMNS;
000373  0207  8740 FDFF                         CL        D+7
000374  0209  D3C0 0000      P                 LNJ       5,JFB
000375        020B                      L38       EQU       $
000376                                  *                       'FOR' I:=0 'STEP' 1 'UNTIL' 79
000377  020B  8740 FDF7                         CL        D+3
000378                                  *                       'DO'
000379                                  *                       'BEGIN'
000380  020D  E840 FDF5                 L39     LDR       6,D+3
000381  020F  6EB1                             ADV       6,-79
000382  0210  6A1F                             BGZ       6,>L15
000383                                  *                            VALUE := 'HEX' (F000);
000384  0211  E870 F000                         LDR       6,=-4096
000385  0213  EF40 FDEE                         STR       6,D+2
000386                                  *                            PUT NEW TEXT ("$");  (LINEFEED C/R)
000387  0215  FBC0 FDEE                         LAB       7,D+4
000388  0217  9BC0 FE44                         LAB       1,T41
000389  0219  9F87                             STB       1,$B7
000390  021A  D3C0 0000      P                 LNJ       5,VKPNT
000391                                  *                            GET HEX (VALUE);
000392  021C  FBC0 FDE7                         LAB       7,D+4
000393  021E  9BC0 FDE3                         LAB       1,D+2
000394  0220  9F87                             STB       1,$B7
000395  0221  D3C0 0000      P                 LNJ       5,VKGH
000396                                  *                            'IF' VALUE 'EQ' 'HEX' (F000)
000397                                  *                            'THEN'
000398                                  *                                 'GO TO' NO MORE INPUT;
000399  0223  E840 FDDE                         LDR       6,D+2
000400  0225  E970 F000                         CMR       6,=-4096
000401  0227  0908                             BE        >L15
000402                                  *                            BUFFER1 @I@ := VALUE;
000403  0228  9840 FDDA                         LDR       1,D+3
000404  022A  EF10 0000      X                 STR       6,<BUFF1,$R1
000405                                  *                       'END';
000406                                  *       NO MORE INPUT;
000407  022C  8AC0 FDD6                         INC       D+3
000408  022E  0FDF                             B         >L39
000409                                  *                  'END';
000410                                  *             'END';
000411                                  **;
000412                                  **   PUNCH UNTIL BREAK OR SPECIFIED;
000413                                  **;
000414        022F                      L15       EQU       $
000415                                  *                 PUT NEW TEXT ("MAXIMUM CARDS ARE$");
000416  022F  FBC0 FDD2                         LAB       7,D+2
000417  0231  9BC0 FE2C                         LAB       1,T44
000418  0233  9F87                             STB       1,$B7
000419  0234  D3C0 0000      P                 LNJ       5,VKPNT
```

```
000420                                  *           PUT DECIMAL (MAXIMUM CARDS);
000421  0236  FBC0 FDCB                      LAB         7,D+2
000422  0238  E840 0000       P            LDR         6,MAXIMU
000423  023A  EF07                        STR         6,$B7
000424  023B  D3C0 0000       P            LNJ         5,VKPD
000425                                  *           ASK NEW QUESTION ("READY PUNCH, TOTAL CARDS$");
000426  023D  FBC0 FDC4                      LAB         7,D+2
000427  023F  9BC0 FE28                      LAB         1,T45
000428  0241  9F87                        STB         1,$B7
000429  0242  D3C0 0000       P            LNJ         5,VKANQ
000430                                  *           GET DECIMAL (CARDS SPECIFIED);
000431  0244  FBC0 FDBD                      LAB         7,D+2
000432  0246  9BC0 0000       P            LAB         1,CARDSS
000433  0248  9F87                        STB         1,$B7
000434  0249  D3C0 0000       P            LNJ         5,VKGD
000435  024B  D3C0 0000       P            LNJ         5,JWR
000436                                  *           WAIT FOR RETURN;
000437                                  *       PUNCH;
000438        024D                      L10       EQU         $
000439                                  *           LABEL 1 := 'LITERAL' (P5);
000440  024D  E870 5035                      LDR         6,=20533
000441  024F  EF40 0000       P            STR         6,LABEL1
000442                                  *           ERROR FLAG := 0;
000443  0251  8740 0000       P            CL          ERRORF
000444                                  *           CLOCK SEMAPHORE := 0;
000445  0253  8740 0000       P            CL          CKSEM
000446  0255  FBC0 FDAC                      LAB         7,D+2
000447  0257  C840 0000       P            LDR         4,CARDSS
000448  0259  CF07                        STR         4,$B7
000449                                  *           PUNCH A DECK (CARDS SPECIFIED, TYPE OF PUNCHES);
000450  025A  D840 0000       P            LDR         5,TYPEOF
000451  025C  DF40 FDA6                      STR         5,D+3
000452  025E  D3C0 0000       P            LNJ         5,JPD
000453                                  *           PUT NEW TEXT ("PUNCHED $");
000454  0260  FBC0 FDA1                      LAB         7,D+2
000455  0262  9BC0 FE13                      LAB         1,T46
000456  0264  9F87                        STB         1,$B7
000457  0265  D3C0 0000       P            LNJ         5,VKPNT
000458                                  *           CALCULATE ELAPSED TIME (CARD COUNT);
000459  0267  FBC0 FD9A                      LAB         7,D+2
000460  0269  E840 0000       P            LDR         6,CARDSR
000461  026B  EF07                        STR         6,$B7
000462  026C  D3C0 0000       P            LNJ         5,JET
000463                                  *           'IF' ERROR FLAG 'NE' 0
000464                                  *              'THEN'
000465                                  *              'BEGIN'
000466                                  *       WRONG RESPONSE;
000467  026E  E840 0000       P            LDR         6,ERRORF
000468  0270  6926                        BEZ         6,>L47
000469                                  *                       ASK NEW QUESTION
000470        0271                      L48       EQU         $
000471                                  *                       ("ERROR CARDS OFFSET STACKED$");
000472  0271  FBC0 FD90                      LAB         7,D+2
000473  0273  9BC0 FE08                      LAB         1,T49
000474  0275  9F87                        STB         1,$B7
000475  0276  D3C0 0000       P            LNJ         5,VKANQ
000476                                  *                       TEMP := 'HEX' (590D);
000477  0278  E870 590D                      LDR         6,=22797
000478  027A  EF40 FD86                      STR         6,D+1
000479                                  *                       GET CHAR (TEMP);
000480  027C  FBC0 FD85                      LAB         7,D+2
000481  027E  9BC0 FD82                      LAB         1,D+1
000482  0280  9F87                        STB         1,$B7
000483  0281  D3C0 0000       P            LNJ         5,VKGC
000484                                  *                          'IF' TEMP 'NE' 'LITERAL' (Y)
000485                                  *                          'THEN'
000486                                  *                          'BEGIN'
000487  0283  E840 FD7D                      LDR         6,D+1
000488  0285  6D59                        CMV         6,89
000489  0286  0910                        BE          >L47
000490                                  *                                'IF' TEMP 'NE' 'LITERAL' (N)
000491                                  *                                'THEN'
000492                                  *                                     'GO TO' WRONG RESPONSE;
000493  0287  6D4E                        CMV         6,78
000494  0288  09E9                        BNE         >L48
000495                                  *                                LABEL 2 := 'LITERAL' (OS);
000496  0289  C870 4F53                      LDR         4,=20307
000497  028B  CF40 0000       P            STR         4,LABEL2
000498  028D  D3C0 0000       P            LNJ         5,JRE
000499                                  *                                REPORT ERROR;
000500  028F  FBC0 FD72                      LAB         7,D+2
000501  0291  9BC0 FDF9                      LAB         1,T52
000502  0293  9F87                        STB         1,$B7
000503  0294  D3C0 0000       P            LNJ         5,VKPT
000504                                  *                                PUT TEXT (", ERROR EJECT$");
000505                                  *                          'END';
000506                                  *              'END';
000507        0296                      L47       EQU         $
000508                                  *           CARDS SPECIFIED := CARDS READ;
000509                                  *       VERIFY AGAIN;
000510  0296  E840 0000       P            LDR         6,CARDSR
000511  0298  EF40 0000       P            STR         6,CARDSS
000512        029A                      L53       EQU         $
000513                                  *           R OR V MODE := 'LITERAL' (V);
000514  029A  6C56                        LDV         6,86
000515  029B  EF40 0000       P            STR         6,RORVMD
000516                                  *           ASK NEW QUESTION ("VERIFY$");
000517  029D  FBC0 FD64                      LAB         7,D+2
000518  029F  9BC0 FDF3                      LAB         1,T54
000519  02A1  9F87                        STB         1,$B7
000520  02A2  D3C0 0000       P            LNJ         5,VKANQ
000521                                  *           TEMP := 'HEX' (590D);
000522  02A4  E870 590D                      LDR         6,=22797
000523  02A6  EF40 FD5A                      STR         6,D+1
000524                                  *           GET CHAR (TEMP);
000525  02A8  FBC0 FD59                      LAB         7,D+2
000526  02AA  9BC0 FD56                      LAB         1,D+1
000527  02AC  9F87                        STB         1,$B7
000528  02AD  D3C0 0000       P            LNJ         5,VKGC
000529                                  *           'IF' TEMP 'NE' 'LITERAL' (N)
000530                                  *           'THEN'
000531                                  *           'BEGIN'
000532  02AF  E840 FD51                      LDR         6,D+1
```

```
000533  02B1  6D4E                          CMV     6,78
000534  02B2  0905                          BE      >L55
000535                              *               'IF' TEMP 'NE' 'LITERAL' (Y)
000536                              *               'THEN'
000537                              *                       'GO TO' VERIFY AGAIN;
000538  02B3  6D59                          CMV     6,89
000539  02B4  09E6                          BNE     >L53
000540                              *               LETTER RP;
000541  02B5  D3C0 0000       P             LNJ     5,KJLR
000542                              *           'END';
000543                              *       'END';          (OF PROCEDURE)
000544                              *       'END';
000545                              *       'FINISH'
000546  02B7  83C8 FD48             L55     JMP     *D
000547        02B9                  L1      EQU     $
000548  02B9  83C0 0000       P             JMP     ZK$EX
000549                                      XLOC    ZK$EX
000550                                      CTRL LINK ZK$EX
000551                                      XLOC    ZV$LR
000552                                      CTRL LINK ZV$LR
000553                                      XLOC    ZHRTCI
000554                                      XLOC    ZHRTCC
000555                                      XLOC    VKPT
000556                                      XLOC    VKPNT
000557                                      XLOC    VKANQ
000558                                      XLOC    VKPD
000559                                      XLOC    VKGC
000560                                      XLOC    VKGD
000561                                      XLOC    VKGH
000562                                      XLOC    VKGB
000563                                      XLOC    VKRRD
000564                                      XLOC    ID
000565                                      XLOC    STATWA
000566                                      XLOC    STATUS
000567                                      XLOC    CHANNE
000568                                      XLOC    PUESDO
000569                                      XLOC    TYPEOF
000570                                      XLOC    SETTOH
000571                                      XLOC    SETTOA
000572                                      XLOC    BINARY
000573                                      XLOC    LABEL1
000574                                      XLOC    LABEL2
000575                                      XLOC    MAXIMU
000576                                      XLOC    CARDSS
000577                                      XLOC    CARDSR
000578                                      XLOC    LOWADD
000579                                      XLOC    ROKVMO
000580                                      XLOC    ERRORF
000581                                      XLOC    CKSEM
000582                                      XLOC    READOK
000583                                      XLOC    BUFFI
000584                                      XLOC    IOWORD
000585                                      XLOC    JRE
000586                                      XLOC    JIO
000587                                      XLOC    JIL
000588                                      XLOC    JWR
000589                                      XLOC    JSE
000590                                      XLOC    JCS
000591                                      XLOC    JFC
000592                                      XLOC    JET
000593                                      XLOC    JFB
000594                                      XLOC    JPD
000595                                      XLOC    KJLR
000596                                      XLOC    LETR
000597                                      XLOC    LETV
000598                                      XLOC    ASKNEX
000599  02BB  0098                          END     ZVKJLP,ZVKJLP
0000  ERR COUNT
L
```

```
E   ZVKJLR.LIST                          05/19/78  1429.7R W 05/19/78  1422.7      4528440000
000001                                                TITLE     ZVKJLR,'REV 00'
000002          00AB                                  XDEF      KJLR
000003          00A6                                  XDEF      ZVKJLR
000004          0008                      ZKCOM       COMM      8
000005  0000    0000                      D           RESV      8,0
000006  0008    000F                      T6          DC        15
000007  0009    5245 4144 4552                        TEXT      'READER CHANNEL$'
        000C    2043 4841 4E4E
                454C 2400
000008  0011    000C                      T15         DC        12
000009  0012    5645 5249 4659                        TEXT      'VERIFY MODE$'
        0015    204D 4F44 4524
000010  0018    0018                      T17         DC        24
000011  0019    4841 5645 2031                        TEXT      'HAVE 100 CARD TEST'
        001C    3030 2043 4152
                4420 5445 5354
000012  0022    2044 4543 4B24                        TEXT      ' DECK$'
000013  0025    000A                      T22         DC        10
000014  0026    5245 4144 204D                        TEXT      'READ MODE$'
        0029    4F44 4524
000015  002B    0012                      T23         DC        18
000016  002C    4D41 5849 4D55                        TEXT      'MAXIMUM CARDS ARE$'
        002F    4D20 4341 5244
                5320 4152 4524
000017  0035    0010                      T24         DC        16
000018  0036    2C20 4445 5349                        TEXT      ', DESIRED TOTAL$'
        0039    5245 4420 544F
                5441 4C24
000019  003E    0012                      T27         DC        18
000020  003F    5241 4E44 4F4D                        TEXT      'RANDOM PICK DELAY$'
        0042    2050 4943 4B20
                4445 4C41 5924
000021  0048    0006                      T30         DC        6
000022  0049    5641 4C55 4524                        TEXT      'VALUE$'
000023  004C    0012                      T31         DC        18
000024  004D    5245 4144 5920                        TEXT      'READY RANDOM DECK$'
        0050    5241 4E44 4F4D
                2044 4543 4B24
000025  0056    000B                      T35         DC        11
000026  0057    5350 4543 4946                        TEXT      'SPECIFIED $'
        005A    4945 4420 2400
000027  005D    0011                      T36         DC        17
000028  005E    2C20 4255 5420                        TEXT      ', BUT READ ONLY $'
        0061    5245 4144 204F
                4E4C 5920 2400
000029  0067    004A                      T48         DC        74
000030  0068    5354 4154 5553                        TEXT      'STATUS ERROR CARDS'
        006B    2045 5252 4F52
                2043 4152 4453
000031  0071    204F 4646 5345                        TEXT      ' OFFSET STACKED',Z'0D0A','AND CARD AF'
        0074    5420 5354 4143
        0077    4B45 440D 0A41
        007A    4E44 2043 4152
                4420 4146
000032  007F    5445 5220 4441                        TEXT      'TER DATA ERROR CARDS OFFSET$'
        0082    5441 2045 5252
                4F52 2043 4152
                4453 204F 4646
                5345 5424
000033  008D    001D                      T53         DC        29
000034  008E    4156 4552 4147                        TEXT      'AVERAGE DELAY SPEE'
        0091    4520 4445 4C41
                5920 5350 4545
000035  0097    442C 2049 4E20                        TEXT      'D, IN MSEC$'
        009A    4D53 4543 2400
000036  009D    0006                      T54         DC        6
000037  009E    5245 4144 2024                        TEXT      'READ $'
000038  00A1    0007                      T56         DC        7
000039  00A2    5645 5249 4659                        TEXT      'VERIFY$'
        00A5    2400
000040          00A6                      ZVKJLR      EQU       $
000041  00A6    8F00 0001           K                 SAVE      <ZKCOM+1,Z'0011'
        00A8    0011
000042                                    *         'BEGIN'
000043  00A9    0F81 024B                   B         L1
000044                                    *           'INTEGER' TEMP, ID 2;
000045  00AB    DFC0 FF54                 KJLR        STB       5,D
000046                                    *             ID 2 := ID;
000047                                    *         START PROCEDURE;
000048  00AD    E840 0000           P               LDR       6,ID
000049  00AF    EF40 FF52                           STR       6,D+2
000050          00B1                      L3        EQU       $
000051                                    *             'IF' ID 'EQ' 'HEX' (2088)
000052                                    *                'THEN'
000053                                    *                  'BEGIN'
000054  00B1    E840 0000           P               LDR       6,ID
000055  00B3    E970 2088                           CMR       6,=8328
000056  00B5    09BA                                BNE       >L4
000057                                    *                  'INTEGER' READER CHANNEL;
000058                                    *         ASK READER QUESTION;
000059          00B6                      L5        EQU       $
000060                                    *                  ASK NEW QUESTION ("READER CHANNEL$");
000061  00B6    FBC0 FF4D                           LAB       7,D+4
000062  00B8    9BC0 FF4F                           LAB       1,T6
000063  00BA    9F87                                STB       1,$B7
000064  00BB    D3C0 0000           P               LNJ       5,VKANQ
000065  00BD    D3C0 0000           P               LNJ       5,VKGB
000066                                    *                  'IF' GET BREAK 'EQ' TRUE
000067                                    *                  'THEN'
000068                                    *                     'GO TO' ASK NEXT;
000069  00BF    6D01                                CMV       6,1
000070  00C0    0901 0000           P               BE        ASKNEX
000071  00C2    FBC0 FF41                           LAB       7,D+4
000072  00C4    9BC0 FF3E                           LAB       1,D+3
000073  00C6    9F87                                STB       1,$B7
000074  00C7    D3C0 0000           P               LNJ       5,VKGH
000075                                    *                  GET HEX (READER CHANNEL);
000076                                    *                  'IF' READER CHANNEL 'MASK' 'HEX' (FFC0) 'EQ' 0
000077  00C9    E840 FF39                           LDR       6,D+3
000078  00CB    E570 FFC0                           AND       6,=-64
000079  00CD    6969                                BEZ       6,>L5
000080                                    *                  'OR' READER CHANNEL 'MASK' 'HEX' (003F) 'NE' 0
```

```
000081                                  *                       'THEN'
000082                                  *                           'GO TO' ASK READER QUESTION;
000083  00CE  E840 FF34                         LDR     6,D+3
000084  00D0  E570 003F                         AND     6,=63
000085  00D2  69E4                              BNEZ    6,>L5
000086                                  *                       CREATE FUNCTION CODES (READER CHANNEL);
000087  00D3  FBC0 FF30                         LAB     7,D+4
000088  00D5  E640 FF2D                         LDR     6,D+3
000089  00D7  EF07                              STR     6,$B7
000090  00D8  D3C0 0000         P               LNJ     5,JFC
000091  00DA  FBC0 FF29                         LAB     7,D+4
000092  00DC  9BC0 FF25                         LAB     1,D+2
000093  00DE  9F87                              STB     1,$B7
000094                                  *                       INOUT (ID 2, INPUT ID @0@);
000095  00DF  0F00 000D         X       N1      NOP     <IOWORD+13
000096  00E1  E840 FFFE                         LDR     6,N1+1
000097  00E3  EF40 FF2I                         STR     6,D+5
000098  00E5  D3C0 0000         P               LNJ     5,JIO
000099                                  *                       'IF' ID 2 'NE' 'HEX' (2008) 'AND'
000100  00E7  E840 FF1A                         LDR     6,D+2
000101  00E9  E970 2008                         CMR     6,=8200
000102  00EB  0904                              BE      >L4
000103                                  *                           ID 2 'NE' 'HEX' (208A)
000104                                  *                       'THEN'
000105                                  *                           'GO TO' ASK READER QUESTION;
000106  00EC  E970 208A                         CMR     6,=8330
000107  00EE  09C8                              BNE     >L5
000108                                  *                   'END';
000109          00EF                    L4      EQU     $
000110                                  *                   ZHRTCC := 0;
000111  00EF  8740 0000         P               CL      ZHRTCC
000112                                  *                   ZHRTCI := 0;
000113  00F1  8740 0000         P               CL      ZHRTCI
000114                                  *                   SUM := 0;
000115  00F3  8740 0000         P               CL      SUM
000116                                  *                   TOTAL := 0;
000117  00F5  8740 0000         P               CL      TOTAL
000118                                  *                   RESPONSE := 0;
000119  00F7  8740 0000         P               CL      RESPON
000120                                  *                   BASE := 'HEX' (6369);
000121  00F9  E870 6369                         LDR     6,=25449
000122  00FB  EF40 0000         P               STR     6,BASE
000123  00FD  FBC0 FF05                         LAB     7,D+3
000124  00FF  9BC0 0000         P               LAB     1,BINARY
000125  0101  9F87                              STB     1,$B7
000126                                  *                   INOUT (INITIALIZE,OUTPUT CONTROL @0@);
000127  0102  ABC0 0000         P               LAB     2,IOWORD
000128  0104  AFC0 FEFF                         STB     2,D+4
000129  0106  D3C0 0000         P               LNJ     5,JIO
000130                                  *                   ERROR LABEL := 'LITERAL' (R1);
000131  0108  E870 5231                         LDR     6,=21041
000132  010A  EF40 0000         P               STR     6,LABEL1
000133                                  *                   LOW ADDRESS := ZVDLR;
000134  010C  C840 0000         P               LDR     4,ZV$LR
000135  010E  CF40 0000         P               STR     4,LOWADD
000136                                  *                   'IF' ID 2 'EQ' 'HEX' (2008)
000137                                  *                   'THEN'
000138  0110  D840 FEF1                         LDR     5,D+2
000139  0112  D970 2008                         CMR     5,=8200
000140  0114  098F                              BNE     >L12
000141  0115  FBC0 FEED                         LAB     7,D+3
000142  0117  9BC0 0000         P               LAB     1,BINARY
000143  0119  9F87                              STB     1,$B7
000144                                  *                   INOUT (BINARY MODE, OUTPUT CONFIGURATION @0@)
000145                                  *                   'ELSE'
000146  011A  0F00 000B         X       N2      NOP     <IOWORD+11
000147  011C  F840 FFFE                         LDR     7,N2+1
000148  011E  FF40 FEE5                         STR     7,D+4
000149  0120  D3C0 0000         P               LNJ     5,JIO
000150  0122  0F8E                              B       >L13
000151  0123  FBC0 FEDF                 L12     LAB     7,D+3
000152  0125  9BC0 0000         P               LAB     1,SETTOH
000153  0127  9F87                              STB     1,$B7
000154  0128  0F00 0005         X       N3      NOP     <IOWORD+5
000155  012A  E840 FFFE                         LDR     6,N3+1
000156  012C  EF40 FED7                         STR     6,D+4
000157  012E  D3C0 0000         P               LNJ     5,JIO
000158                                  *                   INOUT (SET TO HOLLERITH, OUTPUT TASK @0@);
000159          0130                    L13     EQU     $
000160                                  *                   'IF' HAVE GENERATED CHECKSUMS 'NE' TRUE 'AND'
000161  0130  E840 0000         P               LDR     6,HAVEGE
000162  0132  6D01                              CMV     6,1
000163  0133  0939                              BE      >L14
000164                                  *                   R OR V MODE 'EQ' 'LITERAL' (V)
000165                                  *                   'THEN'
000166                                  *                   'BEGIN'
000167  0134  C840 0000         P               LDR     4,RORVMD
000168  0136  4D56                              CMV     4,86
000169  0137  09B5                              BNE     >L14
000170                                  *                   PUT NEW TEXT ("VERIFY MODE$");
000171                                  *               TRY AGAIN;
000172  0138  FBC0 FECA                         LAB     7,D+3
000173  013A  9BC0 FED6                         LAB     1,T15
000174  013C  9F87                              STB     1,$B7
000175  013D  D3C0 0000         P               LNJ     5,VKPNT
000176          013F                    L16     EQU     $
000177                                  *                   ASK NEW QUESTION ("HAVE 100 CARD TEST DECK$");
000178  013F  FBC0 FEC3                         LAB     7,D+3
000179  0141  9BC0 FED6                         LAB     1,T17
000180  0143  9F87                              STB     1,$B7
000181  0144  D3C0 0000         P               LNJ     5,VKANQ
000182                                  *                   TEMP := 'HEX' (590D);
000183  0146  E870 590D                         LDR     6,=22797
000184  0148  EF40 FEB8                         STR     6,D+1
000185                                  *                   GET CHAR (TEMP);
000186  014A  FBC0 FEB8                         LAB     7,D+3
000187  014C  9BC0 FEB4                         LAB     1,D+1
000188  014E  9F87                              STB     1,$B7
000189  014F  D3C0 0000         P               LNJ     5,VKGC
000190                                  *                   'IF' TEMP 'EQ' 'LITERAL' (Y)
000191                                  *                   'THEN'
000192                                  *                   'BEGIN'
000193  0151  E840 FEAF                         LDR     6,D+1
```

```
000194  0153  6D59                          CMV     6,89
000195  0154  0991                          BNE     >L18
000196                              *                       CARDS SPECIFIED := 100;
000197  0155  4C64                          LDV     4,100
000198  0156  CF40 0000      P             STR     4,CARDSS
000199                              *                       RESPONSE := 'HEX' (AAAA);
000200  0158  D870 AAAA                      LDR     5,=-21846
000201  015A  DF40 0000      P             STR     5,RESPON
000202  015C  D3C0 0000      P             LNJ     5,JGT
000203                              *                       GENERATE TEST CHECKSUMS;
000204                              *                       LOW ADDRESS := ZVDLR;
000205  015E  E840 0000      P             LDR     6,ZV$LR
000206  0160  EF40 0000      P             STR     6,LOWADD
000207                              *                       RESPONSE := 0;
000208                              *                   'END'
000209  0162  8740 0000      P             CL      RESPON
000210                              *                   'ELSE'
000211                              *                   'BEGIN'
000212  0164  0F88                          B       >L14
000213        0165                    L18   EQU     $
000214                              *                       'IF' TEMP 'NE' 'LITERAL' (N)
000215                              *                       'THEN'
000216                              *                           'GO TO' TRY AGAIN;
000217  0165  E840 FE9B                      LDR     6,D+1
000218  0167  6D4E                          CMV     6,78
000219  0168  09D7                          BNE     >L16
000220                              *                       R OR V MODE := 'LITERAL' (R);
000221  0169  4C52                          LDV     4,82
000222  016A  CF40 0000      P             STR     4,RORVMD
000223                              *                   'END';
000224                              *               'END';
000225        016C                    L14   EQU     $
000226                              *           CARDS READ := 1;
000227  016C  6C01                          LDV     6,1
000228  016D  EF40 0000      P             STR     6,CARDSR
000229                              *           'IF' R OR V MODE 'EQ' 'LITERAL' (R)
000230                              *           'THEN'
000231                              *           'BEGIN'
000232  016F  C840 0000      P             LDR     4,RORVMD
000233  0171  4D52                          CMV     4,82
000234  0172  09A6                          BNE     >L21
000235                              *                   PUT NEW TEXT ("READ MODE$");
000236  0173  FBC0 FE8F                      LAB     7,D+3
000237  0175  9BC0 FEAF                      LAB     1,T22
000238  0177  9F87                          STB     1,$B7
000239  0178  D3C0 0000      P             LNJ     5,VKPNT
000240                              *                   PUT NEW TEXT ("MAXIMUM CARDS ARE$");
000241  017A  FBC0 FE88                      LAB     7,D+3
000242  017C  9BC0 FEAE                      LAB     1,T23
000243  017E  9F87                          STB     1,$B7
000244  017F  D3C0 0000      P             LNJ     5,VKPNT
000245                              *                   PUT DECIMAL  (MAXIMUM CARDS       );
000246  0181  FBC0 FE81                      LAB     7,D+3
000247  0183  E840 0000      P             LDR     6,MAXIMD
000248  0185  EF07                          STR     6,$B7
000249  0186  D3C0 0000      P             LNJ     5,VKPD
000250                              *                   ASK QUESTION (", DESIRED TOTAL$");
000251  0188  FBC0 FE7A                      LAB     7,D+3
000252  018A  9BC0 FEAA                      LAB     1,T24
000253  018C  9F87                          STB     1,$B7
000254  018D  D3C0 0000      P             LNJ     5,VKAQ
000255                              *                   GET DECIMAL  (CARDS SPECIFIED     )
000256                              *               'END'
000257  018F  FBC0 FE73                      LAB     7,D+3
000258  0191  9BC0 0000      P             LAB     1,CARDSS
000259  0193  9F87                          STB     1,$B7
000260  0194  D3C0 0000      P             LNJ     5,VKGD
000261                              *           'ELSE'
000262                              *           'BEGIN'
000263                              *       ASK AGAIN;
000264  0196  0F81 0038                      B       L25
000265        0198                    L21   EQU     $
000266        0198                    L26   EQU     $
000267                              *                   ASK NEW QUESTION ("RANDOM PICK DELAY$");
000268  0198  FBC0 FE6A                      LAB     7,D+3
000269  019A  9BC0 FEA3                      LAB     1,T27
000270  019C  9F87                          STB     1,$B7
000271  019D  D3C0 0000      P             LNJ     5,VKANQ
000272                              *                   TEMP := 'HEX' (590D);
000273  019F  E870 590D                      LDR     6,=22797
000274  01A1  EF40 FE5F                      STR     6,D+1
000275                              *                   GET CHAR (TEMP);
000276  01A3  FBC0 FE5F                      LAB     7,D+3
000277  01A5  9BC0 FE5B                      LAB     1,D+1
000278  01A7  9F87                          STB     1,$B7
000279  01A8  D3C0 0000      P             LNJ     5,VKGC
000280                              *                   'IF' TEMP 'NE' 'LITERAL' (Y)
000281                              *                   'THEN'
000282                              *                   'BEGIN'
000283  01AA  E840 FE56                      LDR     6,D+1
000284  01AC  6D59                          CMV     6,89
000285  01AD  0901 0021                      BE      L25
000286                              *                       'IF' TEMP 'NE' 'LITERAL' (N)
000287                              *                       'THEN'
000288                              *                           'GO TO' ASK AGAIN;
000289  01AF  6D4E                          CMV     6,78
000290  01B0  09E8                          BNE     >L26
000291                              *                           ASK NEW QUESTION ("VALUE$");
000292  01B1  FBC0 FE51                      LAB     7,D+3
000293  01B3  9BC0 FE94                      LAB     1,T30
000294  01B5  9F87                          STB     1,$B7
000295  01B6  D3C0 0000      P             LNJ     5,VKANQ
000296  01B8  FBC0 FE4A                      LAB     7,D+3
000297  01BA  9BC0 0000      P             LAB     1,RESPON
000298  01BC  9F87                          STB     1,$B7
000299  01BD  D3C0 0000      P             LNJ     5,VKGD
000300                              *                       GET DECIMAL (RESPONSE);
000301  01BF  F840 0000      P             LDR     7,RESPON
000302  01C1  FB40 0000      P             MUL     7,HERTZ
000303                              *                       RESPONSE := RESPONSE*HERTZ/1000;
000304  01C3  F370 03E8                      DIV     7,=1000
000305  01C5  FF40 0000      P             STR     7,RESPON
000306                              *                   'CODE''BEGIN']
```

```
000307                                  *                                   SUB $R6,=500   CHECK REMAINDER
000308                                  *                                   BLZ $R6,>+$A
000309  01C7  E270 01F4                         SUB     $R6,=500
000310  01C9  6800             T               BLZ     $R6,>+$A
000311                                  *                                   )CB)
000312                                  *                                   INC =$R7
000313                                  *                               $A  STR $R7,)RESPONSE
000314  01CA  8AD7                             INC     =$R7
000315  01CB  FF40 0000         P      $A       STR     $R7,RESPON
000316                                  *                                       BASE := 0;
000317  01CD  8740 0000         P              CL      BASE
000318                                  *                                   'END';
000319                                  *                               'END';
000320        01CF                      L25    EQU     $
000321                                  *                               PUT NEW TEXT ("READY RANDOM DECK$");
000322  01CF  FBC0 FE33                         LAB     7,D+3
000323  01D1  9BC0 FE7A                         LAB     1,T31
000324  01D3  9F87                              STB     1,$B7
000325  01D4  D3C0 0000         P               LNJ     5,VKPNT
000326  01D6  D3C0 0000         P               LNJ     5,JWR
000327                                  *                               WAIT FOR RETURN;
000328                                  *                               READ OR PUNCH := READ @0@;
000329  01D8  E800 0007         X               LDR     6,<IOWORD+7
000330  01DA  EF40 0000         P               STR     6,READOR
000331                                  *                               CHECK STATE (SHOULD BE ON);
000332  01DC  FBC0 FE26                         LAB     7,D+3
000333  01DE  4C01                              LDV     4,1
000334  01DF  CF07                              STR     4,$B7
000335  01E0  D3C0 0000         P               LNJ     5,JCS
000336                                  *                               STATUS SHOULD BE := 'HEX' (8000);
000337  01E2  E870 8000                         LDR     6,=-32768
000338  01E4  EF40 0000         P               STR     6,STATUS
000339  01E6  FBC0 FE1C                         LAB     7,D+3
000340  01E8  9BC0 0000         P               LAB     1,BUFF1
000341  01EA  9F87                              STB     1,$B7
000342  01EB  0F00 0007         X      N4       NOP     <IOWORD+7
000343  01ED  C840 FFFE                         LDR     4,N4+1
000344  01EF  CF40 FE14                         STR     4,D+4
000345                                  *                               IOLD (BUFFER1 @0@,READ @0@,171);
000346  01F1  D870 00AB                         LDR     5,=171
000347  01F3  DF40 FE11                         STR     5,D+5
000348  01F5  D3C0 0000         P               LNJ     5,JIL
000349  01F7  D3C0 0000         P               LNJ     5,JSE
000350                                  *                               CHECK STATUS FOR ERRORS;
000351                                  *                               'IF' STATUS WAS 'NE' STATUS SHOULD BE
000352                                  *                               'THEN'
000353                                  *                                   'GO TO' START PROCEDURE;
000354  01F9  E840 0000         P               LDR     6,STATWA
000355  01FB  E940 0000         P               CMR     6,STATUS
000356  01FD  0981 FEB3                         BNE     L3
000357                                  *                               ERROR LABEL := 'LITERAL' (R2);
000358  01FF  C870 5232                         LDR     4,=21042
000359  0201  CF40 0000         P               STR     4,LABEL1
000360                                  *                               ERROR FLAG := 0;
000361  0203  8740 0000         P               CL      ERRORF
000362                                  *                               CLOCK SEMAPHORE := 0;
000363  0205  8740 0000         P               CL      CKSEM
000364  0207  0004                              DC      4
000365                                  *                               CLOCK ON;
000366                                  *                           READ ANOTHER;
000367        0208                      L33    EQU     $
000368                                  *                               'IF' 'BITS' @1,15@ STATUS WAS 'EQ' 0 'AND'
000369  0208  E840 0000         P               LDR     6,STATWA
000370  020A  6801 0038                         BLZ     6,L34
000371                                  *                                   R OR V MODE 'EQ' 'LITERAL'(R) 'AND'
000372  020C  C840 0000         P               LDR     4,RORVMD
000373  020E  4D52                              CMV     4,82
000374  020F  0981 0033                         BNE     L34
000375                                  *                                   CARDS READ 'LT' CARDS SPECIFIED
000376                                  *                               'THEN'
000377                                  *                               'BEGIN'
000378  0211  D840 0000         P               LDR     5,CARDSR
000379  0213  D940 0000         P               CMR     5,CARDSS
000380  0215  0881 002D                         BAGE    L34
000381  0217  0005                              DC      5
000382                                  *                                   CLOCK OFF;
000383                                  *                                   CARDS READ := CARDS READ - 1;
000384  0218  88C0 0000         P               DEC     CARDSR
000385                                  *                                   PUT NEW TEXT ("SPECIFIED $"    );
000386  021A  FBC0 FDE8                         LAB     7,D+3
000387  021C  9BC0 FE39                         LAB     1,T35
000388  021E  9F87                              STB     1,$B7
000389  021F  D3C0 0000         P               LNJ     5,VKPNT
000390                                  *                                   PUT NUMBER   (CARDS SPECIFIED  );
000391  0221  FBC0 FDE1                         LAB     7,D+3
000392  0223  E840 0000         P               LDR     6,CARDSS
000393  0225  EF07                              STR     6,$B7
000394  0226  D3C0 0000         P               LNJ     5,JPN
000395  0228  FBC0 FDDA                         LAB     7,D+3
000396  022A  9BC0 FE32                         LAB     1,T36
000397  022C  9F87                              STB     1,$B7
000398  022D  D3C0 0000         P               LNJ     5,VKPT
000399                                  *                                   PUT TEXT     (", BUT READ ONLY $");
000400  022F  FBC0 FDD3                         LAB     7,D+3
000401  0231  E840 0000         P               LDR     6,CARDSR
000402  0233  EF07                              STR     6,$B7
000403  0234  D3C0 0000         P               LNJ     5,JPN
000404                                  *                                   PUT NUMBER   (CARDS READ       );
000405                                  *                                   STATUS SHOULD BE := 'HEX' (4000);
000406  0236  E870 4000                         LDR     6,=16384
000407  0238  EF40 0000         P               STR     6,STATUS
000408                                  *                                   'IF' ZVDITY 'EQ' 0
000409                                  *                                   'THEN'
000410  023A  C840 0000         P               LDR     4,ZV$TTY
000411  023C  4983                              BNEZ    4,>L37
000412                                  *                                       STATUS SHOULD BE := 0;
000413  023D  8740 0000         P               CL      STATUS
000414  023F  D3C0 0000         P      L37      LNJ     5,JSE
000415                                  *                                   CHECK STATUS FOR ERRORS;
000416                                  *                                   'GO TO' END OF LETTER R;
000417  0241  0F81 00AA                         B       L58
000418                                  *                               'END';
000419        0243                      L34    EQU     $
```

```
000420                                  *           GENERATE CHECKSUMS (READ OR PUNCH);
000421  0243  FBC0 FDBF                  *         LAB       7,D+3
000422  0245  E840 0000       P         *         LDR       6,READOR
000423  0247  EF07                      *         STR       6,$B7
000424  0248  D3C0 0000       P         *         LNJ       5,JGC
000425                                  *                   'IF' 'BITS' @2,14@ STATUS WAS 'NE' 'HEX' (2)
000426  024A  E2C0 0000       P         *         LLH       6,STATWA
000427  024C  6046                      *         SOR       6,6
000428  024D  6D02                      *         CMV       6,2
000429  024E  090D                      *         BE        >L39
000430                                  *                   'AND' CARDS READ 'GE' CARDS SPECIFIED - 1
000431                                  *                   'THEN'
000432  024F  C840 0000       P         *         LDR       4,CARDSS
000433  0251  4EFF                      *         ADV       4,-1
000434  0252  C940 0000       P         *         CMR       4,CARDSR
000435  0254  0A07                      *         BAG       >L39
000436                                  *                   STATUS SHOULD BE := STATUS WAS 'MASK' 'HEX' (C000);
000437  0255  C840 0000       P         *         LDR       4,STATWA
000438  0257  C570 C000                 *         AND       4,=-16384
000439  0259  CF40 0000       P         *         STR       4,STATUS
000440  025B  D3C0 0000       P         *L39      LNJ       5,JSE
000441                                  *                   CHECK STATUS FOR ERRORS;
000442                                  *                   CARDS READ := CARDS READ + 1;
000443  025D  8AC0 0000       P         *         INC       CARDSR
000444                                  *                   'IF' R OR V MODE 'EQ' 'LITERAL' (V)
000445                                  *                   'THEN'
000446                                  *                   'BEGIN'
000447  025F  E840 0000       P         *         LDR       6,RORVMD
000448  0261  6D56                      *         CMV       6,86
000449  0262  0983                      *         BNE       >L40
000450  0263  D3C0 0000       P         *         LNJ       5,JCC
000451                                  *                   COMPARE CHECKSUMS;
000452                                  *                   'END';
000453        0265                      *L40      EQU       $
000454                                  *                   'IF' CARDS READ 'LT' CARDS SPECIFIED
000455                                  *                   'THEN'
000456                                  *                         'GO TO' READ ANOTHER;
000457  0265  E840 0000       P         *         LDR       6,CARDSR
000458  0267  E940 0000       P         *         CMR       6,CARDSS
000459  0269  0801 FF9E                 *         BAL       L33
000460                                  *                   'IF' CARDS SPECIFIED 'EQ' CARDS READ
000461                                  *                   'THEN'
000462                                  *                   'BEGIN'
000463  026B  E940 0000       P         *         CMR       6,CARDSS
000464  026D  0985                      *         BNE       >L42
000465                                  *                   READ OR PUNCH := FALSE;
000466  026E  8740 0000       P         *         CL        READOR
000467                                  *                   'GO TO' READ ANOTHER;
000468  0270  0F81 FF97                 *         B         L33
000469                                  *                   'END';
000470        0272                      *L42      EQU       $
000471  0272  0005                      *         DC        5
000472                                  *                   CLOCK OFF;
000473                                  *                   'IF' TOTAL 'NE' 0
000474  0273  E840 0000       P         *         LDR       6,TOTAL
000475  0275  6984                      *         BNEZ      6,>L43
000476                                  *                   'OR' ERROR FLAG 'NE' 0
000477                                  *                   'THEN'
000478                                  *                   'BEGIN'
000479  0276  C840 0000       P         *         LDR       4,ERRORF
000480  0278  4924                      *         BEZ       4,>L44
000481        0279                      *L43      EQU       $
000482                                  *                         'IF' ID 2 'EQ' 'HEX' (2008)
000483                                  *                         'THEN'
000484                                  *                               'GO TO' REPORT SPEED;
000485                                  *       WRONG RESPONSE:
000486  0279  E840 FD88                 *         LDR       6,D+2
000487  027B  E970 2008                 *         CMR       6,=8200
000488  027D  091F                      *         BE        >L44
000489        027E                      *L47      EQU       $
000490                                  *                         ASK NEW QUESTION ("STATUS ERROR CARDS OFFSET STACKED!C
000491                                  *                           L !AND CARD AFTER DATA ERROR CARDS OFFSET$");
000492  027E  FBC0 FD84                 *         LAB       7,D+3
000493  0280  9BC0 FDE6                 *         LAB       1,T48
000494  0282  9F87                      *         STB       1,$B7
000495  0283  D3C0 0000       P         *         LNJ       5,VKANQ
000496                                  *                         TEMP := 'HEX' (590D);
000497  0285  E870 590D                 *         LDR       6,=22797
000498  0287  EF40 FD79                 *         STR       6,D+1
000499                                  *                         GET CHAR (TEMP);
000500  0289  FBC0 FD79                 *         LAB       7,D+3
000501  028B  9BC0 FD75                 *         LAB       1,D+1
000502  028D  9F87                      *         STB       1,$B7
000503  028E  D3C0 0000       P         *         LNJ       5,VKGC
000504                                  *                         'IF' TEMP 'NE' 'LITERAL' (Y)
000505                                  *                         'THEN'
000506                                  *                         'BEGIN'
000507  0290  E840 FD70                 *         LDR       6,D+1
000508  0292  6D59                      *         CMV       6,89
000509  0293  0909                      *         BE        >L44
000510                                  *                               'IF' TEMP 'NE' 'LITERAL' (N)
000511                                  *                               'THEN'
000512                                  *                                     'GO TO' WRONG RESPONSE;
000513  0294  6D4E                      *         CMV       6,78
000514  0295  09E9                      *         BNE       >L47
000515                                  *                                LABEL 2 := 'LITERAL' (PE);
000516  0296  C870 5045                 *         LDR       4,=20549
000517  0298  CF40 0000       P         *         STR       4,LABEL2
000518  029A  D3C0 0000       P         *         LNJ       5,JRE
000519                                  *                                REPORT ERROR;
000520                                  *                         'END';
000521                                  *                   'END';
000522                                  *       REPORT SPEED;
000523        029C                      *L44      EQU       $
000524                                  *                   'IF' R OR V MODE 'EQ' 'LITERAL' (V)
000525                                  *                   'THEN'
000526                                  *                   'BEGIN'
000527  029C  E840 0000       P         *         LDR       6,RORVMD
000528  029E  6D56                      *         CMV       6,86
000529  029F  099E                      *         BNE       >L51
000530                                  *                         'IF' TOTAL 'NE' 0
000531                                  *                         'THEN'
000532                                  *                         'BEGIN'
```

```
000533  02A0  C640 0000      P                 LDR       4,TOTAL
000534  02A2  492B                             BEZ       4,>L52
000535                                  *                       'INTEGER' AVERAGE ERROR SPEED;
000536                                  *                       'INTEGER' DENOMINATOR;
000537                                  *                       DENOMINATOR := HERTZ * TOTAL;
000538  02A3  CB40 0000      P                 MUL       4,HERTZ
000539  02A5  CF40 FD5E                        STR       4,D+4
000540                                  *                       AVERAGE ERROR SPEED := SUM*1000/DENOMINATOR;
000541  02A7  F840 0000      P                 LDR       7,SUM
000542  02A9  FB70 03E8                        MUL       7,=1000
000543  02AB  F354                            DIV       7,=$R4
000544  02AC  FF40 FD56                        STR       7,D+3
000545                                  *                       PUT NEW TEXT ("AVERAGE DELAY SPEED, IN MSEC$");
000546  02AE  FBC0 FD56                        LAB       7,D+5
000547  02B0  9BC0 FDDC                        LAB       1,T53
000548  02B2  9F87                             STB       1,$B7
000549  02B3  D3C0 0000      P                 LNJ       5,VKPNT
000550                                  *                       PUT DECIMAL (AVERAGE ERROR SPEED);
000551  02B5  FBC0 FD4F                        LAB       7,D+5
000552  02B7  E840 FD4B                        LDR       6,D+3
000553  02B9  EF07                             STR       6,$B7
000554  02BA  D3C0 0000      P                 LNJ       5,VKPD
000555                                  *                   'END';
000556                                  *               'END'
000557                                  *               'ELSE'
000558                                  *               'BEGIN'
000559  02BC  0F91                             B         >L52
000560        02BD                      L51     EQU       $
000561                                  *                   CARDS READ := CARDS READ -1;
000562  02BD  88C0 0000      P                 DEC       CARDSR
000563                                  *                   PUT NEW TEXT ("READ $");
000564  02BF  FBC0 FD43                        LAB       7,D+3
000565  02C1  9BC0 FDDB                        LAB       1,T54
000566  02C3  9F87                             STB       1,$B7
000567  02C4  D3C0 0000      P                 LNJ       5,VKPNT
000568                                  *                   CALCULATE ELAPSED TIME (CARDS READ     );
000569  02C6  FBC0 FD3C                        LAB       7,D+3
000570  02C8  E840 0000      P                 LDR       6,CARDSR
000571  02CA  EF07                             STR       6,$B7
000572  02CB  D3C0 0000      P                 LNJ       5,JET
000573                                  *               'END';
000574                                  *           VERIFY AGAIN:
000575        02CD                      L52     EQU       $
000576                                  *                   R OR V MODE := 'LITERAL' (V);
000577  02CD  6C56                             LDV       6,86
000578  02CE  EF40 0000      P                 STR       6,RORVMD
000579                                  *                   ASK NEW QUESTION ("VERIFY$");
000580  02D0  FBC0 FD32                        LAB       7,D+3
000581  02D2  9BC0 FDCE                        LAB       1,T56
000582  02D4  9F87                             STB       1,$B7
000583  02D5  D3C0 0000      P                 LNJ       5,VKANQ
000584                                  *                   TEMP := 'HEX' (590D);
000585  02D7  E870 590D                        LDR       6,=22797
000586  02D9  EF40 FD27                        STR       6,D+1
000587                                  *                   GET CHAR (TEMP);
000588  02DB  FBC0 FD27                        LAB       7,D+3
000589  02DD  9BC0 FD23                        LAB       1,D+1
000590  02DF  9F87                             STB       1,$B7
000591  02E0  D3C0 0000      P                 LNJ       5,VKGC
000592                                  *                   'IF' TEMP 'EQ' 0
000593                                  *                   'THEN'
000594                                  *                       'GO TO' ASK NEXT;
000595  02E2  E840 FD1E                        LDR       6,D+1
000596  02E4  6901 0000      P                 BEZ       6,ASKNEX
000597                                  *                   'IF' TEMP 'NE' 'LITERAL' (N)
000598                                  *                   'THEN'
000599                                  *                   'BEGIN'
000600  02E6  6D4E                             CMV       6,78
000601  02E7  0905                             BE        >L58
000602                                  *                       'IF' TEMP 'NE' 'LITERAL' (Y)
000603                                  *                       'THEN'
000604                                  *                           'GO TO' VERIFY AGAIN;
000605  02E8  6D59                             CMV       6,89
000606  02E9  09E4                             BNE       >L52
000607                                  *                   'GO TO' START PROCEDURE;
000608  02EA  0F81 FDC6                        B         L3
000609                                  *               'END';
000610                                  *           END OF LETTER K:
000611  02EC  FBC0 FD16                L58     LAB       7,D+3
000612  02EE  E840 0000      P                 LDR       6,CHANNE
000613  02F0  EF07                             STR       6,$B7
000614  02F1  D3C0 0000      P                 LNJ       5,JFC
000615                                  *                   CREATE FUNCTION CODES (CHANNEL);
000616                                  *           'END';    (OF PROCEDURE)
000617                                  *           'END';    (OF SEGMENT)
000618                                  *           'FINISH'
000619  02F3  83C8 FD0C                        JMP       *D
000620        02F5                      L1      EQU       $
000621  02F5  83C0 0000      P                 JMP       ZK$EX
000622                                          XLOC      ZK$EX
000623                                          CTRL LINK ZK$EX
000624                                          XLOC      ZV$LR
000625                                          CTRL LINK ZV$LR
000626                                          XLOC      ZHRTCI
000627                                          XLOC      ZHRTCC
000628                                          XLOC      VKPI
000629                                          XLOC      VKPNT
000630                                          XLOC      VKAQ
000631                                          XLOC      VKANQ
000632                                          XLOC      VKPD
000633                                          XLOC      VKGC
000634                                          XLOC      VKGD
000635                                          XLOC      VKGH
000636                                          XLOC      VKGB
000637                                          XLOC      ID
000638                                          XLOC      STATWA
000639                                          XLOC      STATUS
000640                                          XLOC      HERTZ
000641                                          XLOC      CHANNE
000642                                          XLOC      SETTOH
000643                                          XLOC      BINARY
000644                                          XLOC      LABEL1
000645                                          XLOC      LABEL2
```

```
000646                                  XLOC    MAXIMU
000647                                  XLOC    CARDSS
000648                                  XLOC    CARDSR
000649                                  XLOC    LOWADD
000650                                  XLOC    RORVMO
000651                                  XLOC    HAVEGE
000652                                  XLOC    ERRORF
000653                                  XLOC    CKSEM
000654                                  XLOC    BASE
000655                                  XLOC    SUM
000656                                  XLOC    TOTAL
000657                                  XLOC    RESPON
000658                                  XLOC    ZV$TTY
000659                                  CTRL LINK ZV$TT
000660                                  XLOC    READOR
000661                                  XLOC    BUFF1
000662                                  XLOC    IOWORD
000663                                  XLOC    JGT
000664                                  XLOC    JGC
000665                                  XLOC    JRE
000666                                  XLOC    JIO
000667                                  XLOC    JIL
000668                                  XLOC    JPN
000669                                  XLOC    JWR
000670                                  XLOC    JSE
000671                                  XLOC    JCS
000672                                  XLOC    JFC
000673                                  XLOC    JET
000674                                  XLOC    JCC
000675                                  XLOC    ASKNEX
000676  02F7  00A6                      END     ZVKJLR,ZVKJLR
0000  ERR COUNT
```

```
E   ZVKJFS.LIST                                  05/19/78  1429.8R W 05/19/78  1422.5      3638070000
000001                                                      TITLE     ZVKJFS,'REV 00'
000002          0078                                        XDEF      KJFS
000003          0073                                        XDEF      ZVKJFS
000004          0008                              ZKCOM     COMM      8
000005  0000    0000                              D         RESV      6,0
000006  0006    001B                              T3        DC        27
000007  0007    4352 4D53 332C                              TEXT      'CRMS3, REV G, APR '
        000A    2052 4556 2047
                2C20 4150 5220
000008  0010    3230 2C20 3139                              TEXT      '20, 1978$'
        0013    3738 2400
000009  0015    000F                              T4        DC        15
000010  0016    4348 414E 4E45                              TEXT      'CHANNEL NUMBER$'
        0019    4C20 4E55 4D42
                4552 2400
000011  001E    001A                              T8        DC        26
000012  001F    2C20 5752 4F4E                              TEXT      ', WRONG DEVICE ON '
        0022    4720 4445 5649
                4345 204F 4E20
000013  0028    4348 414E 4E45                              TEXT      'CHANNEL$'
        002B    4C24
000014  002C    0011                              T10       DC        17
000015  002D    2C20 4348 4543                              TEXT      ', CHECK LIGHT ON$'
        0030    4B20 4C49 4748
                5420 4F4E 2400
000016  0036    0010                              T15       DC        16
000017  0037    504F 5745 5220                              TEXT      'POWER FREQUENCY$'
        003A    4652 4551 5545
                4E43 5924
000018  003F    000B                              T21       DC        11
000019  0040    4D41 524B 2053                              TEXT      'MARK SENSE$'
        0043    454E 5345 2400
000020  0046    000B                              T26       DC        11
000021  0047    3430 2043 4F4C                              TEXT      '40 COLUMNS$'
        004A    554D 4E53 2400
000022  004D    001E                              T30       DC        30
000023  004E    5357 4954 4348                              TEXT      'SWITCHES INCORRECT'
        0051    4553 2049 4E43
                4F52 5245 4354
000024  0057    204F 5220 4144                              TEXT      ' OR ADAPTER$'
        005A    4150 5445 5224
000025  005D    001A                              T35       DC        26
000026  005E    2741 2720 4F50                              TEXT      '''A'' OPTION IS RECO'
        0061    5449 4F4E 2049
                5320 5245 434F
000027  0067    4D4D 454E 4445                              TEXT      'MMENDED$'
        006A    4424
000028  006B    000D                              T36       DC        13
000029  006C    2C20 5354 5543                              TEXT      ', STUCK AT 0$'
        006F    4B20 4154 2030
                2400
000030          0073                              ZVKJFS    EQU       $
000031  0073    8F00 0001       K                           SAVE      <ZKCOM+1,Z'0011'
        0075    0011
000032                                            *         'BEGIN'
000033  0076    0F81 0201                                   B         L1
000034                                            *           'INTEGER' TEMP CHARACTER;
000035  0078    DFC0 FF87                         KJFS      STB       5,D
000036                                            *             PUT TITLE         ("CRMS3, REV G, APR 20, 1978$");
000037  007A    FBC0 FF87                                   LAB       7,D+2
000038  007C    9BC0 FF89                                   LAB       1,T3
000039  007E    9F87                                        STB       1,$B7
000040  007F    D3C0 0000       P                           LNJ       5,VKPTI
000041                                            *             NUMBER OF COLUMNS := 80;
000042  0081    6C50                                        LDV       6,80
000043  0082    EF40 0000       P                           STR       6,NUMBER
000044                                            *             MAXIMUM CARDS := 'BITS'@15,1@ZVDHR - 'BITS'@15,1@ ZVDLR;
000045  0084    C840 0000       P                           LDR       4,ZV$HR
000046  0086    4041                                        SOR       4,1
000047  0087    D840 0000       P                           LDR       5,ZV$LR
000048  0089    5041                                        SOR       5,1
000049  008A    C255                                        SUB       4,=$R5
000050  008B    CF40 0000       P                           STR       4,MAXIMU
000051                                            *             MAXIMUM CARDS := MAXIMUM CARDS/7*2;
000052  008D    F854                                        LDR       7,=$R4
000053  008E    FF56                                        STR       7,=$R6
000054  008F    8757                                        CL        =$R7
000055  0090    70EF                                        DAR       7,15
000056  0091    F370 0007                                   DIV       7,=7
000057  0093    FF40 0000       P                           STR       7,MAXIMU
000058                                            *             LABEL 1 := 'LITERAL' (FS);
000059  0095    E870 4653                                   LDR       6,=18003
000060  0097    EF40 0000       P                           STR       6,LABEL1
000061                                            *             HAVE GENERATED CHECKSUMS := FALSE;
000062  0099    8740 0000       P                 CL        HAVEGE
000063                                            *             READ STATUS := TRUE;
000064  009B    4C01                                        LDV       4,1
000065  009C    CF40 0000       P                           STR       4,READS1
000066                                            *             ASK NEW QUESTION ("CHANNEL NUMBER$"    );
000067  009E    FBC0 FF63                                   LAB       7,D+2
000068  00A0    9BC0 FF74                                   LAB       1,T4
000069  00A2    9F87                                        STB       1,$B7
000070  00A3    D3C0 0000       P                           LNJ       5,VKANQ
000071  00A5    FBC0 FF5C                                   LAB       7,D+2
000072  00A7    9BC0 0000       P                           LAB       1,CHANNE
000073  00A9    9F87                                        STB       1,$B7
000074  00AA    D3C0 0000       P                           LNJ       5,VKGH
000075                                            *             GET HEX           (CHANNEL              );
000076                                            *             'IF' CHANNEL 'MASK' 'HEX' (FFC0) 'EQ' 0 'OR'
000077  00AC    E840 0000       P                           LDR       6,CHANNE
000078  00AE    E570 FFC0                                   AND       6,=-64
000079  00B0    6901 0000       P                           BEZ       6,FRESHS
000080                                            *               CHANNEL 'MASK' 'HEX' (003F) 'NE' 0
000081                                            *             'THEN'
000082                                            *               'GO TO' FRESH START;
000083  00B2    E840 0000       P                           LDR       6,CHANNE
000084  00B4    E570 003F                                   AND       6,=63
000085  00B6    6981 0000       P                           BNEZ      6,FRESHS
000086  00B8    FBC0 FF49                                   LAB       7,D+2
000087  00BA    E840 0000       P                           LDR       6,CHANNE
000088  00BC    EF07                                        STR       6,$B7
000089  00BD    D3C0 0000       P                           LNJ       5,JFC
```

```
000090                                  *           CREATE FUNCTION CODES(CHANNEL);
000091                                  **;
000092                                  *       'COMMENT' CLEAR CHANNEL - MAY BE BUSY;
000093                                  **;
000094  00BF  FBC0 FF42                         LAB       7,D+2
000095  00C1  9BC0 0000      P                  LAB       1,BINARY
000096  00C3  9F87                              STB       1,$B7
000097                                  *           INOUT (INITIALIZE, OUTPUT CONTROL @0@);
000098                                  **;
000099                                  **    IDENTIFY DEVICE;
000100                                  **;
000101  00C4  ABC0 0000      P                  LAB       2,IOWORD
000102  00C6  AFC0 FF3C                         STB       2,D+3
000103  00C8  D3C0 0000      P                  LNJ       5,JIO
000104  00CA  FBC0 FF37                         LAB       7,D+2
000105  00CC  9BC0 0000      P                  LAB       1,ID
000106  00CE  9F87                              STB       1,$B7
000107                                  *           INOUT (ID,INPUT DEVICE ID @0@);
000108  00CF  0F00 000D      X     N1           NOP       <IOWORD+13
000109  00D1  E840 FFFE                         LDR       6,N1+1
000110  00D3  EF40 FF2F                         STR       6,D+3
000111  00D5  D3C0 0000      P                  LNJ       5,JIO
000112  00D7  E840 0000      P                  LDR       6,ID
000113  00D9  E970 2008                         CMR       6,=8200
000114  00DB  0901 0023                         BE        L7
000115                                  *           'IF' ID 'NE' 'HEX' (2008) 'AND' ID 'NE' 'HEX' (208A) 'AND'
000116  00DD  E970 208A                         CMR       6,=8330
000117  00DF  0901 001F                         BE        L7
000118                                  *                ID 'NE' 'HEX' (2088)
000119                                  *           'THEN'
000120                                  *           'BEGIN'
000121  00E1  E970 2088                         CMR       6,=8328
000122  00E3  0901 001B                         BE        L7
000123                                  *           LABEL2 := 'LITERAL' (ID);
000124  00E5  C870 4944                         LDR       4,=18756
000125  00E7  CF40 0000      P                  STR       4,LABEL2
000126  00E9  D840 0000      P                  LDR       $R5,CHANNE
000127  00EB  C840 0000      P                  LDR       $R4,ID
000128                                  *           LOAD R4 R5 (ID, CHANNEL);
000129  00ED  D3C0 0000      P                  LNJ       5,JRE
000130                                  *           REPORT ERROR;
000131                                  *           PUT NEW TEXT (", WRONG DEVICE ON CHANNEL$");
000132  00EF  FBC0 FF12                         LAB       7,D+2
000133  00F1  9BC0 FF2C                         LAB       1,T8
000134  00F3  9F87                              STB       1,$B7
000135  00F4  D3C0 0000      P                  LNJ       5,VKPNT
000136  00F6  FBC0 FF0B                         LAB       7,D+2
000137  00F8  E840 0000      P                  LDR       6,CHANNE
000138  00FA  EF07                             STR       6,$B7
000139  00FB  D3C0 0000      P                  LNJ       5,VKPH
000140                                  *           PUT HEX       (CHANNEL                    );
000141                                  *           'GO TO' FRESH START;
000142  00FD  83C0 0000      P                  JMP       FRESHS
000143                                  *       'END';
000144        00FF                      L7     EQU       $
000145                                  *       'CODE' 'BEGIN'
000146                                  *            STS =$R7
000147                                  *            STR $R7,)SREG
000148  00FF  8C57                              STS       =$R7
000149  0100  FF40 0000      P                  STR       $R7,SREG
000150                                  *           'IF' 'BITS' @1,15@ SREG 'EQ' 1
000151                                  *           'THEN'
000152                                  *           'BEGIN'
000153  0102  E840 0000      P                  LDR       6,SREG
000154  0104  6881 0012                         BGEZ      6,L9
000155                                  *           LABEL 2:= 'LITERAL' (CL);
000156  0106  C870 434C                         LDR       4,=17228
000157  0108  CF40 0000      P                  STR       4,LABEL2
000158  010A  D840 0000      P                  LDR       $R5,SETTOA
000159  010C  C840 0000      P                  LDR       $R4,SREG
000160                                  *           LOAD R4 R5 (SREG, CLEAR);
000161  010E  D3C0 0000      P                  LNJ       5,JRE
000162                                  *           REPORT ERROR;
000163  0110  FBC0 FEF1                         LAB       7,D+2
000164  0112  9BC0 FF19                         LAB       1,T10
000165  0114  9F87                              STB       1,$B7
000166  0115  D3C0 0000      P                  LNJ       5,VKPT
000167                                  *           PUT TEXT (", CHECK LIGHT ON$");
000168                                  *       'END';
000169        0117                      L9     EQU       $
000170                                  *       'IF' TEST TYPE 'NE' 0
000171                                  *       'THEN'
000172                                  *            'GO TO' SKIP POWER QUESTION;
000173  0117  E840 0000      P                  LDR       6,TESTTY
000174  0119  69A2                              BNEZ      6,>L12
000175                                  *            HERTZ := 60;
000176  011A  4C3C                              LDV       4,60
000177  011B  CF40 0000      P                  STR       4,HERTZ
000178                                  *           'IF' 'BITS' @1,13@ SREG 'EQ' 1
000179                                  *           'THEN'
000180                                  *                'GO TO' NO POWER QUESTION;
000181                                  **;
000182                                  **    POWER QUESTION ON 6/34 OR 6/36;
000183                                  **;
000184  011D  82C0 0000      P                  LB        SREG,Z'2000'
      011F  2000
000185  0120  0516                              BBT       >L16
000186                                  *           ASK NEW QUESTION ("POWER FREQUENCY$"   );
000187  0121  FBC0 FEE0                         LAB       7,D+2
000188  0123  9BC0 FF12                         LAB       1,T15
000189  0125  9F87                              STB       1,$B7
000190  0126  D3C0 0000      P                  LNJ       5,VKANQ
000191  0128  FBC0 FED9                         LAB       7,D+2
000192  012A  9BC0 0000      P                  LAB       1,HERTZ
000193  012C  9F87                              STB       1,$B7
000194  012D  D3C0 0000      P                  LNJ       5,VKGD
000195                                  *           GET DECIMAL       (HERTZ                  );
000196  012F  E840 0000      P                  LDR       6,HERTZ
000197  0131  6D32                             CMV       6,50
000198  0132  0904                             BE        >L16
000199                                  *           'IF' HERTZ 'NE' 50 'AND'HERTZ 'NE' 60
000200                                  *           'THEN'
000201                                  *                'GO TO' FRESH START;
```

```
000202                                  *       NO POWER QUESTION;
000203  0133  6D3C                              CMV     6,60
000204  0134  0981 0000      P                   BNE     FRESHS
000205  0136  E840 0000      P         L16       LDR     6,HERTZ
000206  0138  6001                              SOL     6,1
000207  0139  EF40 0000      P                   STR     6,HERTZ
000208                                  *         HERTZ := HERTZ * 2;
000209                                  *       SKIP POWER QUESTION;
000210        013B                      L12       EQU     $
000211                                  *           PUESDO CARD STATUS := 0;
000212                                  **;
000213                                  **    SKIP MARK SENSE QUESTIONS IF ONLY A PUNCH;
000214                                  **;
000215  013B  8740 0000      P                     CL      PUESDO
000216                                  *             'IF' ID 'NE''HEX' (2008)
000217                                  *           'THEN'
000218                                  *                 'GO TO'  SET LEVEL AND ISA;
000219                                  **;
000220                                  **    MARK SENSE AND COLUMN NUMBER;
000221                                  **;
000222                                  *       ASK MARK SENSE QUESTION;
000223  013D  E840 0000      P                   LDR     6,ID
000224  013F  E970 2008                          CMR     6,=8200
000225  0141  0981 007D                          BNE     L29
000226        0143                      L20       EQU     $
000227                                  *           ASK NEW QUESTION ("MARK SENSE$"         );
000228  0143  FBC0 FEBE                          LAB     7,D+2
000229  0145  9BC0 FEF9                          LAB     1,121
000230  0147  9F87                              STB     1,$B7
000231  0148  D3C0 0000      P                   LNJ     5,VKANQ
000232                                  *           TEMP CHARACTER := 'HEX' (4E0D);
000233  014A  E870 4E0D                          LDR     6,=19981
000234  014C  EF40 FEB4                          STR     6,D+1
000235  014E  FBC0 FEB3                          LAB     7,D+2
000236  0150  9BC0 FEB0                          LAB     1,D+1
000237  0152  9F87                              STB     1,$B7
000238  0153  D3C0 0000      P                   LNJ     5,VKGC
000239                                  *           GET CHAR          (TEMP CHARACTER         );
000240                                  *           'IF' TEMP CHARACTER 'EQ' 'LITERAL' (N)
000241                                  *           'THEN'
000242                                  *                 'GO TO' CHECK SWITCHES;
000243  0155  E840 FEAB                          LDR     6,D+1
000244  0157  6D4E                              CMV     6,78
000245  0158  0926                              BE      >L27
000246                                  *           'IF' TEMP CHARACTER 'EQ' 'LITERAL' (Y)
000247                                  *           'THEN'
000248                                  *                 'BEGIN'
000249                                  *       ASK COLUMN QUESTION;
000250  0159  6D59                              CMV     6,89
000251  015A  09E9                              BNE     >L20
000252        015B                      L25       EQU     $
000253                                  *             PUESDO CARD STATUS := 'HEX' (1000);
000254  015B  E870 1000                          LDR     6,=4096
000255  015D  EF40 0000      P                   STR     6,PUESDO
000256                                  *             ASK NEW QUESTION ("40 COLUMNS$");
000257  015F  FBC0 FEA2                          LAB     7,D+2
000258  0161  9BC0 FEE4                          LAB     1,T26
000259  0163  9F87                              STB     1,$B7
000260  0164  D3C0 0000      P                   LNJ     5,VKANQ
000261                                  *             TEMP CHARACTER := 'HEX' (4E0D);
000262  0166  E870 4E0D                          LDR     6,=19981
000263  0168  EF40 FE98                          STR     6,D+1
000264  016A  FBC0 FE97                          LAB     7,D+2
000265  016C  9BC0 FE94                          LAB     1,D+1
000266  016E  9F87                              STB     1,$B7
000267  016F  D3C0 0000      P                   LNJ     5,VKGC
000268                                  *               GET CHAR (TEMP CHARACTER    );
000269                                  *               'IF' TEMP CHARACTER 'EQ' 'LITERAL' (N)
000270                                  *               'THEN'
000271                                  *                     'GO TO' CHECK SWITCHES;
000272  0171  E840 FE8F                          LDR     6,D+1
000273  0173  6D4E                              CMV     6,78
000274  0174  090A                              BE      >L27
000275                                  *                   'IF' TEMP CHARACTER 'EQ' 'LITERAL' (Y)
000276                                  *                   'THEN'
000277                                  *                   'BEGIN'
000278  0175  6D59                              CMV     6,89
000279  0176  09E5                              BNE     >L25
000280                                  *                         PUESDO CARD STATUS := 'HEX' (1800);
000281  0177  C870 1800                          LDR     4,=6144
000282  0179  CF40 0000      P                   STR     4,PUESDO
000283                                  *                   NUMBER OF COLUMNS := 40;
000284                                  *                   'END'
000285  017B  5C28                              LDV     5,40
000286  017C  DF40 0000      P                   STR     5,NUMBER
000287                                  *                 'ELSE'
000288                                  *                     'GO TO' ASK COLUMN QUESTION;
000289                                  *               'END'
000290                                  *             'ELSE'
000291                                  *                 'GO TO' ASK MARK SENSE QUESTION;
000292                                  **;
000293                                  ** CHECK  FOR PROPER SETTING OF CARDREADER SWITCHES;
000294                                  **;
000295                                  *       CHECK SWITCHES:
000296                                  *           'BEGIN'
000297  017E  FBC0 FE83      P         L27       LAB     7,D+2
000298  0180  9BC0 0000      P                   LAB     1,STATWA
000299  0182  9F87                              STB     1,$B7
000300                                  *                 INOUT (STATUS WAS,INPUT STATUS WORD @0@);
000301  0183  0F00 000C      X         N2        NOP     <IOWORD+12
000302  0185  E840 FFFE                          LDR     6,N2+1
000303  0187  EF40 FE7B                          STR     6,D+3
000304  0189  D3C0 0000      P                   LNJ     5,JIO
000305                                  *                 'IF' 'BITS' @3,10@ STATUS WAS 'NE'
000306                                  *                      'BITS' @3,10@ PUESDO CARD STATUS
000307                                  *                 'THEN'
000308                                  *                 'BEGIN'
000309  018B  E2C0 0000      P                   LLH     6,STATWA
000310  018D  6055                              SCR     6,5
000311  018E  604D                              SOR     6,13
000312  018F  C2C0 0000      P                   LLH     4,PUESDO
000313  0191  4055                              SCR     4,5
000314  0192  404D                              SOR     4,13
```

```
000315  0193  E954                                  CMR       6,=$R4
000316  0194  0901 002A                             BE        L29
000317                                      *                     LABEL 2 := 'LITERAL' (SW);
000318  0196  D870 5357                             LDR       5,=21335
000319  0198  DF40 0000        P                    STR       5,LABEL2
000320                                      *                     STATUS SHOULD BE := STATUS WAS 'MASK' 'HEX' (C000)
000321                                      *                                         'UNION' PUESDO CARD STATUS;
000322  019A  F840 0000        P                    LDR       7,STATWA
000323  019C  F570 C000                             AND       7,=-16384
000324  019E  F440 0000        P                    OR        7,PUESDO
000325  01A0  FF40 0000        P                    STR       7,STATUS
000326  01A2  D840 0000        P                    LDR       $R5,STATUS
000327  01A4  C840 0000        P                    LDR       $R4,STATWA
000328  01A6  D3C0 0000        P                    LNJ       5,JRE
000329                                      *                     REPORT ERROR;
000330  01A8  FBC0 FE59                             LAB       7,D+2
000331  01AA  E840 0000        P                    LDR       6,STATWA
000332  01AC  EF07                                  STR       6,$B7
000333  01AD  D3C0 0000        P                    LNJ       5,VKPH
000334                                      *                     PUT HEX (STATUS WAS);
000335  01AF  FBC0 FE52                             LAB       7,D+2
000336  01B1  E840 0000        P                    LDR       6,STATUS
000337  01B3  EF07                                  STR       6,$B7
000338  01B4  D3C0 0000        P                    LNJ       5,VKPH
000339                                      *                     PUT HEX (STATUS SHOULD BE);
000340                                      *                     PUT NEW TEXT ("SWITCHES INCORRECT OR ADAPTER$");
000341  01B6  FBC0 FE4B                             LAB       7,D+2
000342  01B8  9BC0 FE94                             LAB       1,T30
000343  01BA  9F87                                  STB       1,$B7
000344  01BB  D3C0 0000        P                    LNJ       5,VKPNT
000345                                      *                     'GO TO' FRESH START;
000346  01BD  83C0 0000        P                    JMP       FRESHS
000347                                      *                 'END';
000348                                      *             'END';
000349                                      *         SET LEVEL AND ISA:
000350  01BF  9840 0000        P    L29             LDR       1,ZV$AF
000351  01C1  1E05                                  ADV       1,5
000352                                      *                 ISA FOR PROGRAM @5+ZVDAF@ := SREG; (SAVE PRIVILEDGED BITS + CPID)
000353  01C2  E840 0000        P                    LDR       6,SREG
000354  01C4  EF10 0000        X                    STR       6,<ISAP.$R1
000355  01C6  9840 0000        P                    LDR       1,ZV$AF
000356  01C8  1E05                                  ADV       1,5
000357                                      *                 ISA FOR DEVICE  @5+ZVDAF@ := SREG;
000358  01C9  E840 0000        P                    LDR       6,SREG
000359  01CB  EF10 0000        X                    STR       6,<ISAD.$R1
000360  01CD  9840 0000        P                    LDR       1,ZV$AF
000361  01CF  1E05                                  ADV       1,5
000362                                      *                 ISA FOR CLOCK   @5+ZVDAF@ := SREG;
000363  01D0  E840 0000        P                    LDR       6,SREG
000364  01D2  EF10 0000        X                    STR       6,<ISAC.$R1
000365                                      *                 ISA FOR PROGRAM @3@ := 'HEX' (7FFF); (SAVE ALL REGISTERS BUT M)
000366  01D4  E870 7FFF                             LDR       6,=32767
000367  01D6  EF40 0003        P                    STR       6,ISAP+3
000368                                      *                 ISA FOR DEVICE  @3@ := 'HEX' (7FFF);
000369  01D8  EF40 0003        P                    STR       6,ISAD+3
000370                                      *                 ISA FOR CLOCK   @3@ := 'HEX' (7FFF);
000371  01DA  EF40 0003        P                    STR       6,ISAC+3
000372  01DC  9FC0 FE24                             STB       $B1,D+1
000373  01DE  0F80             T                    B         >+$A
000374  01DF  83C0 0000        P                    JMP       LEV20
000375  01DF                                        ORG       $-1-$AF
000376  01E0                           $E           ORG       $+1
000377  01E0                           $B           ORG       $-1+$AF
000378  01E1                           $C           ORG       $+1
000379  01E1  82C0 FFFD                $A           LB        -$E,=X'40'
        01E3  0040
000380  01E4  0581 0000        T                    BBF       +$D
000381  01E6  ABC0 FFF9                             LAB       $B2,-$C
000382  01E8  9802                                  LDR       $R1,$B2
000383  01E9  9B92                                  LAB       $B1,$B2.$R1
000384  01EA  0F81 0000        T                    B         +$F
000385  01EC  9BC8 FFF3        $D                   LAB       $B1,*-$B
000386  01EE  9FC0 0005        P    $F              STB       $B1,ISAC+4+$AF
000387  01F0  9CC0 FE10                             LDB       $B1,D+1
000388                                      *                 ADDRESS OF LABEL (ISACP,LEV20); (SET ISA TO POINT TO HANDLER)
000389  01F2  9FC0 FE0E                             STB       $B1,D+1
000390  01F4  0F80             T                    B         >+$A
000391  01F5  83C0 0000        P                    JMP       LEV21
000392  01F5                                        ORG       $-1-$AF
000393  01F6                           $E           ORG       $+1
000394  01F6                           $B           ORG       $-1+$AF
000395  01F7                           $C           ORG       $+1
000396  01F7  82C0 FFFD                $A           LB        -$E,=X'40'
        01F9  0040
000397  01FA  0581 0000        T                    BBF       +$D
000398  01FC  ABC0 FFF9                             LAB       $B2,-$C
000399  01FE  9802                                  LDR       $R1,$B2
000400  01FF  9B92                                  LAB       $B1,$B2.$R1
000401  0200  0F81 0000        T                    B         +$F
000402  0202  9BC8 FFF3        $D                   LAB       $B1,*-$B
000403  0204  9FC0 0005        P    $F              STB       $B1,ISAD+4+$AF
000404  0206  9CC0 FDFA                             LDB       $B1,D+1
000405                                      *                 ADDRESS OF LABEL (ISARP,LEV21); (SET ISA TO POINT TO HANDLER)
000406  0208  9FC0 FDF8                             STB       $B1,D+1
000407  020A  0F80             T                    B         >+$A
000408  020B  83C0 0000        P                    JMP       ASKNEX
000409  020B                                        ORG       $-1-$AF
000410  020C                           $E           ORG       $+1
000411  020C                           $B           ORG       $-1+$AF
000412  020D                           $C           ORG       $+1
000413  020D  82C0 FFFD                $A           LB        -$E,=X'40'
        020F  0040
000414  0210  0581 0000        T                    BBF       +$D
000415  0212  ABC0 FFF9                             LAB       $B2,-$C
000416  0214  9802                                  LDR       $R1,$B2
000417  0215  9B92                                  LAB       $B1,$B2.$R1
000418  0216  0F81 0000        T                    B         +$F
000419  0218  9BC8 FFF3        $D                   LAB       $B1,*-$B
000420  021A  9FC0 0005        P    $F              STB       $B1,ISAP+4+$AF
000421  021C  9CC0 FDE4                             LDB       $B1,D+1
000422  021E  9840 0000        P                    LDR       1,ZV$AF
000423  0220  1F14                                  MLV       1,20
000424  0221  0F00 0000        X    N3              NOP       <ZHISAZ
```

```
000425  0223  9A40 FFFE                    *          ADD       1,N3+1
000426                                     *            @'LOCATION' (ZHISAZ)+20*ZVDAF@ := 'LOCATION' (ISACD @U@);
000427  0225  9F40 FDDC                              STR       1,D+2
000428  0227  0F00 0002        X     N4             NOP       <ISAC+2
000429  0229  E840 FFFE                              LDR       6,N4+1
000430  022B  EF48 FDD6                              STR       6,*D+2
000431  022D  9840 0000        P                    LDR       1,ZV$AF
000432  022F  1F15                                  MLV       1,21
000433  0230  0F00 0000        X     N5             NOP       <ZHISAZ
000434  0232  9A40 FFFE                              ADD       1,N5+1
000435                                     *            @'LOCATION' (ZHISAZ)+21*ZVDAF@ := 'LOCATION' (ISARD @U@);
000436  0234  9F40 FDCD                              STR       1,D+2
000437  0236  0F00 0002        X     N6             NOP       <ISAD+2
000438  0238  E840 FFFE                              LDR       6,N6+1
000439  023A  EF48 FDC7                              STR       6,*D+2
000440  023C  9840 0000        P                    LDR       1,ZV$AF
000441  023E  1F16                                  MLV       1,22
000442  023F  0F00 0000        X     N7             NOP       <ZHISAZ
000443  0241  9A40 FFFE                              ADD       1,N7+1
000444                                     *            @'LOCATION' (ZHISAZ)+22*ZVDAF@ := 'LOCATION' (ISAID @U@);
000445  0243  9F40 FDBE                              STR       1,D+2
000446  0245  0F00 0002        X     N8             NOP       <ISAP+2
000447  0247  E840 FFFE                              LDR       6,N8+1
000448  0249  EF48 FDB8                              STR       6,*D+2
000449                                     *            'IF' TEST TYPE 'EQ' 0
000450                                     *            'THEN'
000451  024B  EF40 FDB6                              STR       6,D+2
000452  024D  C840 0000        P                    LDR       4,TESTTY
000453  024F  4988                                  BNEZ      4,>L34
000454                                     *            PUT NEW TEXT ("'A' OPTION IS RECOMMENDED$");
000455  0250  FBC0 FDB1                              LAB       7,D+2
000456  0252  9BC0 FE0A                              LAB       1,T35
000457  0254  9F87                                  STB       1,$B7
000458  0255  D3C0 0000        P                    LNJ       5,VKPNT
000459        0257                             L34  EQU       $
000460                                     *            TEST TYPE := 'LITERAL' (A);            (DEFAULT TO ALL)
000461  0257  6C41                                  LDV       6,65
000462  0258  EF40 0000        P                    STR       6,TESTTY
000463                                     *              ZHRTCC := 0;
000464  025A  8740 0000        P                    CL        ZHRTCC
000465                                     *              ZHRTC1 := 0;
000466  025C  8740 0000        P                    CL        ZHRTC1
000467                                     *              ZHRTCL := 20;
000468                                     *          'COMMENT' MOVE CP TO LEVEL 22;
000469  025E  4C14                                  LDV       4,20
000470  025F  CF40 0000        P                    STR       4,ZHRTCL
000471                                     *              ZHIAFB := 0;                       (CLEAR ACTIVITY FLAGS)
000472  0261  8740 0000        P                    CL        ZHIAFB
000473  0263  D870 8016                              LDR       5,=-32746
000474  0265  DF40 0000        P                    STR       5,LEVTYP
000475                                     *              LEV ('HEX' (8000),22);             (LEVEL CHANGE TO LEVEL 22)
000476  0267  8E40 0000        P                    LEV       LEVTYP
000477                                     *              LABEL2 := 'LITERAL' (IR);
000478  0269  E870 4952                              LDR       6,=18770
000479  026B  EF40 0000        P                    STR       6,LABEL2
000480  026D  D3C0 0000        P                    LNJ       5,JRE
000481                                     *              REPORT ERROR;
000482  026F  FBC0 FD92                              LAB       7,D+2
000483  0271  9BC0 FDF9                              LAB       1,T36
000484  0273  9F87                                  STB       1,$B7
000485  0274  D3C0 0000        P                    LNJ       5,VKPT
000486                                     *              PUT TEXT (", STUCK AT 0$");
000487                                     *              'GO TO' ASK NEXT;
000488                                     *          'END';
000489  0276  83C0 0000        P                      JMP       ASKNEX
000490                                     *          'END';
000491                                     *          'FINISH'
000492        0278                             L1   EQU       $
000493  0278  83C0 0000        P                    JMP       ZK$EX
000494                                               XLOC      ZK$EX
000495                                               CTRL LINK ZK$EX
000496                                               XLOC      ZV$LR
000497                                               CTRL LINK ZV$LR
000498                                               XLOC      ZV$HR
000499                                               CTRL LINK ZV$HR
000500                                               XLOC      ZHISAZ
000501                                               XLOC      ZHRTCI
000502                                               XLOC      ZHRTCL
000503                                               XLOC      ZHRTCC
000504                                               XLOC      ZV$AF
000505                                               CTRL LINK ZV$AF
000506                                               XLOC      VKPTI
000507                                               XLOC      VKPT
000508                                               XLOC      VKPNT
000509                                               XLOC      VKANQ
000510                                               XLOC      VKPH
000511                                               XLOC      VKGC
000512                                               XLOC      VKGD
000513                                               XLOC      VKGH
000514                                               XLOC      ID
000515                                               XLOC      STATWA
000516                                               XLOC      STATUS
000517                                               XLOC      HERTZ
000518                                               XLOC      CHANNE
000519                                               XLOC      PUESDO
000520                                               XLOC      TESTTY
000521                                               XLOC      SETTOA
000522                                               XLOC      BINARY
000523                                               XLOC      READST
000524                                               XLOC      NUMBER
000525                                               XLOC      LABEL1
000526                                               XLOC      LABEL2
000527                                               XLOC      MAXIMU
000528                                               XLOC      HAVEGE
000529                                               XLOC      LEVTYP
000530                                               XLOC      SREG
000531                                               XLOC      ZHIAFB
000532                                               XLOC      ISAC
000533                                               XLOC      ISAD
000534                                               XLOC      ISAP
000535                                               XLOC      IOWORD
000536                                               XLOC      JRE
000537                                               XLOC      JIO
```

```
000538                                          XLOC    JFC
000539                                          XLOC    FRESHS
000540                                          XLOC    LEV20
000541                                          XLOC    LEV21
000542                                          XLOC    ASKNEX
000543  027A  0073                             END     ZVKJFS,ZVKJFS
0000  ERR COUNT
L
```

```
E  ZV$BK,LIST                    05/19/78  1429.8R W 05/19/78  1424.3       866610000
000001                                 TITLE ZV$BK
000002                                       XLOC  ZHCOMM
000003                                       XLOC FRESHS (FRESH START) NORMAL STARTING POINT TO PROGRAMS
000004          0100                         XDEF ZK$EX
000005          007E                         XDEF  ZHTH2,ZHIVBS,ZHTSA,ZHNTSA,ZHTH13,ZHTH15,ZHRTCI,ZHRTCC
                0080
                0002
                0010
                0073
                0071
                0014
                0015
000006          006F                         XDEF  ZHTH17
000007          0080                         XDEF  ZHISAZ,ZHRTCL
                0016
000008          0017                        XDEF   ZHWDTC,ZHMERC,ZHIAFB,ZHTH29,ZHTH28,ZHTH27,ZHTH26,ZHTH25
                001F
                0020
                0063
                0064
                0065
                0066
                0067
000009          0068                        XDEF   ZHTH24,ZHTH23,ZHTH22,ZHTH21,ZHTH20,ZHTH19,ZHTH18,ZHMEMP
                0069
                006A
                006B
                006C
                006D
                006E
                006F
000010          0070                        XDEF   ZHTH16,ZHLERR,ZHNRES,ZHTH14,ZHPMEM,ZHP-OP,ZHTH12,ZHTH11
                0070
                0071
                0072
                0072
                0073
                0074
                0075
000011          0076                        XDEF   ZHTH10,ZHTH9,ZHTH8,ZHTH7,ZHTH6,ZHOVFL,ZHTH5,ZHOP-N
                0077
                0078
                0079
                007A
                007A
                007B
                007B
000012          007C                        XDEF   ZHTH4,ZHTH3,ZHSC-N,ZHTRC,ZHTH1,ZHMCL,ZHTVBS,ZHPFR
                007D
                007D
                007E
                007F
                007F
                0080
                0000
000013          0000                ZERO    EQU    $
000014          0000                ZHPFR   EQU    ZERO+X'0000'
000015          0002                ZHTSA   EQU    ZERO+X'0002'
000016          0010                ZHNTSA  EQU    ZERO+X'0010'
000017          0014                ZHRTCI  EQU    ZERO+X'0014'
000018          0015                ZHRTCC  EQU    ZERO+X'0015'
000019          0016                ZHRTCL  EQU    ZERO+X'0016'
000020          0017                ZHWDTC  EQU    ZERO+X'0017'
000021          001F                ZHMERC  EQU    ZERO+X'001F'
000022          0020                ZHIAFB  EQU    ZERO+X'0020'
000023          0063                ZHTH29  EQU    ZERO+X'0063'
000024          0064                ZHTH28  EQU    ZERO+X'0064'
000025          0065                ZHTH27  EQU    ZERO+X'0065'
000026          0066                ZHTH26  EQU    ZERO+X'0066'
000027          0067                ZHTH25  EQU    ZERO+X'0067'
000028          0068                ZHTH24  EQU    ZERO+X'0068'
000029          0069                ZHTH23  EQU    ZERO+X'0069'
000030          006A                ZHTH22  EQU    ZERO+X'006A'
000031          006B                ZHTH21  EQU    ZERO+X'006B'
000032          006C                ZHTH20  EQU    ZERO+X'006C'
000033          006D                ZHTH19  EQU    ZERO+X'006D'
000034          006E                ZHTH18  EQU    ZERO+X'006E'
000035          006F                ZHMEMP  EQU    ZERO+X'006F'
000036          006F                ZHTH17  EQU    ZERO+X'006F'
000037          0070                ZHTH16  EQU    ZERO+X'0070'
000038          0070                ZHLERR  EQU    ZERO+X'0070'
000039          0071                ZHNRES  EQU    ZERO+X'0071'
000040          0071                ZHTH15  EQU    ZERO+X'0071'
000041          0072                ZHTH14  EQU    ZERO+X'0072'
000042          0072                ZHPMEM  EQU    ZERO+X'0072'
000043          0073                ZHP-OP  EQU    ZERO+X'0073'
000044          0073                ZHTH13  EQU    ZERO+X'0073'
000045          0074                ZHTH12  EQU    ZERO+X'0074'
000046          0075                ZHTH11  EQU    ZERO+X'0075'
000047          0076                ZHTH10  EQU    ZERO+X'0076'
000048          0077                ZHTH9   EQU    ZERO+X'0077'
000049          0078                ZHTH8   EQU    ZERO+X'0078'
000050          0079                ZHTH7   EQU    ZERO+X'0079'
000051          007A                ZHTH6   EQU    ZERO+X'007A'
000052          007A                ZHOVFL  EQU    ZERO+X'007A'
000053          007B                ZHTH5   EQU    ZERO+X'007B'
000054          007B                ZHOP-N  EQU    ZERO+X'007B'
000055          007C                ZHTH4   EQU    ZERO+X'007C'
000056          007D                ZHTH3   EQU    ZERO+X'007D'
000057          007D                ZHSC-N  EQU    ZERO+X'007D'
000058          007E                ZHTRC   EQU    ZERO+X'007E'
000059          007E                ZHTH2   EQU    ZERO+X'007E'
000060          007F                ZHTH1   EQU    ZERO+X'007F'
000061          007F                ZHMCL   EQU    ZERO+X'007F'
000062          0080                ZHTVBS  EQU    ZERO+X'0080'
000063          0080                ZHIVBS  EQU    ZERO+X'0080'
000064          0080                ZHISAZ  EQU    ZERO+X'0080'
000065          0100                ZK$EX   EQU    ZERO+X'0100'
000066          0100                JUMP    EQU    ZERO+X'0100'
000067  0100                                ORG    JUMP
000068  0100  8380 0000       X            JMP <FRESHS GO TO BEGINNING OF PROGRAM
000069                                 CTRL LINK ZV$FR
```

```
000070                                  CTRL LINK ZV$T
000071                                  CTRL LINK ZV$IH
000072                                  CTRL LINK ZV$IH
000073                                  CTRL LINK ZV$IA
000074                                  CTRL LINK ZV$ER
000075                                  CTRL LINK ZV$BRK
000076                                  CTRL LINK ZV$PCH
000077                                  CTRL LINK ZVKCP
000078                                  CTRL LINK ZVKPTI
000079                                  CTRL LINK ZVKPT
000080                                  CTRL LINK ZVKPNT
000081                                  CTRL LINK ZVKAQ
000082                                  CTRL LINK ZVKANQ
000083                                  CTRL LINK ZVKPC
000084                                  CTRL LINK ZVKPD
000085                                  CTRL LINK ZVKPH
000086                                  CTRL LINK ZVKGT
000087                                  CTRL LINK ZVKGD
000088                                  CTRL LINK ZVKGH
000089                                  CTRL LINK ZVKRRD
000090                                  CTRL LINK ZVKFRD
000091                                  CTRL LINK CARD
000092                                  CTRL LINK ZVKGC
000093                                  CTRL LINK ZVKGB
000094                                  CTRL LINK ZVKJCA
000095                                  CTRL LINK ZVKJCC
000096                                  CTRL LINK ZVKJCS
000097                                  CTRL LINK ZVKJET
000098                                  CTRL LINK ZVKJFB
000099                                  CTRL LINK ZVKJFC
000100                                  CTRL LINK ZVKJGC
000101                                  CTRL LINK ZVKJIL
000102                                  CTRL LINK ZVKJIO
000103                                  CTRL LINK ZVKJIR
000104                                  CTRL LINK ZVKJMC
000105                                  CTRL LINK ZVKJPD
000106                                  CTRL LINK ZVKJPN
000107                                  CTRL LINK ZVKJRE
000108                                  CTRL LINK ZVKJSE
000109                                  CTRL LINK ZVKJTC
000110                                  CTRL LINK ZVKJTD
000111                                  CTRL LINK ZVKJWR
000112                                  CTRL LINK ZVKJLB
000113                                  CTRL LINK ZVKJLC
000114                                  CTRL LINK ZVKJLF
000115                                  CTRL LINK ZVKJLI
000116                                  CTRL LINK ZVKJLK
000117                                  CTRL LINK ZVKJLL
000118                                  CTRL LINK ZVKJLM
000119                                  CTRL LINK ZVKJLN
000120                                  CTRL LINK ZVKJLP
000121                                  CTRL LINK ZVKJLR
000122                                  CTRL LINK ZVKJFS
000123  0102                        END ZV$BK
0000  ERR COUNT
e@L
```

```
E    CRMS3.MAP                    05/22/78  0952.9R W 05/22/78  0949.9        713250000
RS LINKER VERSION 5.00 05/22/78  0945.9 EDT MON
LINK MAP FOR CRMS3
 START          066F
 LOW            0000
 HIGH           3645
 CURRENT        3646
*LOC DEFS
  ZHCOMM        0000
 *ZV$BK         0000
  ZK$EX         0100
  ZHTH2         007E
  ZH1VBS        0080
  ZHTSA         0002
  ZHNTSA        0010
  ZHTH13        0073
  ZHTH15        0071
  ZHRTCI        0014
  ZHRTCC        0015
  ZHTH17        006F
  ZHISAZ        0080
  ZHRTCL        0016
  ZHWDTC        0017
  ZHMERC        001F
  ZHIABD        0020
  ZHTH29        0063
  ZHTH28        0064
  ZHTH27        0065
  ZHTH26        0066
  ZH1H25        0067
  ZHTH24        0068
  ZHTH23        0069
  ZHTH22        006A
  ZHTH21        006B
  ZHTH20        006C
  ZHTH19        006D
  ZHTH18        006E
  ZHMEMP        006F
  ZHTH16        0070
  ZHLERR        0070
  ZHNRES        0071
  ZHTH14        0072
  ZHPMEM        0072
  ZHP-UP        0073
  ZHTH12        0074
  ZHTH11        0075
  ZHTH10        0076
  ZHTH9         0077
  ZHTH8         0078
  ZHTH7         0079
  ZHTH6         007A
  ZHOVFL        007A
  ZHTH5         007B
  ZHOP-N        007B
  ZHTH4         007C
  ZHTH3         007D
  ZHSC-N        007D
  ZHTRC         007E
  ZHTH1         007F
  ZHMCL         007F
  ZHTVBS        0080
  ZHPFR         0000
 *ZV$FR         0102
  ZV$FR         0102
  ZV$F1         0124
  ZV$FS         0147
  ZV$FRA        0154
  ZV$FRX        0155
  ZV$FRR        0119
  ZV$FRB        0156
  ZV$FRM        0153
 *ZV$I          0159  REV. 5.0
  ZV$I          0159
  ZV$IC         0162
  ZV$Q          016B
  ZV$QC         0176
 *ZV$IH         018A
  ZV$IH         018A
  ZV$IHZ        01B2
  ZV$ID         01BF
 *ZV$IH         01DA
  ZV$IH         01DA
  ZV$ID         01DF
  ZV$IAD        01E4
  ZV$--2        01FC
  ZV$--3        020E
 *ZV$IA         0273  REV. 7
  ZV$IA         0276
  ZV$ARG        0325
  ZV$ABF        0327
  ZV$--1        02E2
  ZV$IAV        0274
 *ZV$ER         0332  REV. 5.0
  ZV$ER         0332
  ZV$TA         035E
  ZV$--0        0345
 *ZV$BRK        03A2
  ZV$BRK        03A2
 *ZV$PCH        03BC
  ZV$PCH        03BC
 *ZVKCP         04C6  REV 00
  ZKCOM         04BE
  VKCP          04CD
  ZVKCP         04C8
 *ZVKPT1        04E9  REV 00
  VKPT1         04F1
  ZVKPT1        04EC
 *ZVKPT         0503  REV 00
  VKPT          050B
  ZVKPT         0506
 *ZVKPNT        0519  REV 00
  VKPNT         0521
  ZVKPNT        051C
```

```
*ZVKAQ      052F  REV 00
 VKAQ       0537
 ZVKAQ      0532
*ZVKANQ     0545  REV 00
 VKANQ      054D
 ZVKANQ     0548
*ZVKPC      055B  REV 00
 VKPC       0564
 ZVKPC      0560
*ZVKPD      0575  REV 00
 VKPD       057D
 ZVKPD      0578
*ZVKPH      058A  REV 00
 VKPH       0592
 ZVKPH      058D
*ZVKGT      059F  REV 00
 VKGT       05A9
 ZVCONS     059F
 ZVKGT      05A4
*ZVKGD      05BD  REV 00
 VKGD       05C5
 ZVKGD      05C0
*ZVKGH      05D4  REV 00
 VKGH       05DC
 ZVKGH      05D7
*ZVKRRD     05EB  REV 00
 VKRRD      05F3
 ZVKRRD     05EE
*ZVKFRD     0604  REV 00
 VKFRD      060D
 ZVKFRD     0608
*CARD       0620  REV 00
 FRESHS     066F
 ID         0620
 STATWA     0621
 STATUS     0622
 HERTZ      0623
 CHANNE     0627
 PUESDO     0624
 TESTTY     0625
 TYPEOF     0626
 SETTOH     0628
 SETTOA     0629
 ASEJCT     062A
 BINARY     062B
 READST     0634
 NUMBER     062F
 LABEL1     062D
 LABEL2     062E
 MAXIMU     0631
 CARDSS     0632
 CARDSR     0633
 BITCOU     0635
 LOWADD     0636
 RORVMO     0637
 HAVEGE     0639
 SPEED      0630
 ERRORF     062C
 CKSEM      0642
 L21SEM     0640
 DLVL       0643
 DVSEM      0641
 CPLEVE     0644
 LEVTYP     0645
 TEMPCH     063B
 SREG       0638
 SCK        063C
 IOTM       063D
 IOX1       063E
 IOX2       063F
 READOR     063A
 CHECKS     07BE
 BINCOU     0766
 ASCCNT     0773
 BINOB      077D
 PPROMS     0791
 RPROMS     07A3
 ONESHO     07B5
 WORDCO     0894
 BUFF1      07C5
 BUFF2      0819
 CHAR       0869
 ILLEGA     0891
 ISAC       0709
 ISAD       0728
 ISAP       0747
 IOWORD     0915
 LETR       06EA
 LETV       06EE
 ASKNEX     0671
 CARD       066C
*ZVKGC      0926  REV 00
 VKGC       092F
 ZVKGC      092A
*ZVKGB      094C  REV 00
 VKGB       0962
 ZVKGB      095E
*ZVKJCA     0977  REV 00
 JCA        099C
 ZVKJCA     0997
*ZVKJCC     09F6  REV 00
 JCC        0A25
 ZVKJCC     0A20
*ZVKJCS     0BC2  REV 00
 JCS        0BF1
 ZVKJCS     0BEC
*ZVKJET     0C8A  REV 00
 JET        0CA2
 ZVKJET     0C9D
*ZVKJFB     0CDD  REV 00
 JFB        0CEA
 ZVKJFB     0CE6
*ZVKJFC     0D0A  REV 00
```

```
 JFC          0D13
 ZVKJFC       0D0F
*ZVKJGC       0D30  REV 00
 JGC          0D42
 ZVKJGC       0D3D
*ZVKJIL       0E4C  REV 00
 JIL          0E62
 ZVKJIL       0E5D
*ZVKJIO       0EAA  REV 00
 JIO          0EC4
 ZVKJIO       0EBF
*ZVKJIR       0F06  REV 00
 TIMER        0F6A
 CPRUPT       0F87
 DEV1H        0F98
 LEV20        0F55
 LEV21        0F60
 ZVKJIR       0F4E
*ZVKJMC       1068  REV 00
 JMC          1076
 ZVKJMC       1071
*ZVKJPD       10CB  REV 00
 JPD          10DB
 ZVKJPD       10D6
*ZVKJPN       11B1  REV 00
 JPN          11BA
 ZVKJPN       11B5
*ZVKJRE       11CB  REV 00
 JRE          11D2
 ZVKJRE       11CD
*ZVKJSE       11E1  REV 00
 JSE          12B1
 ZVKJSE       12AC
*ZVKJTC       1450  REV 00
 JGT          145C
 ZVKJTC       1457
*ZVKJTD       14F3  REV 00
 SUM          14F4
 TIME1        14F7
 TOTAL        14F5
 JTD          1502
 RESPON       14F6
 BASE         14F3
 ZVKJTD       14FD
*ZVKJWK       1532  REV 00
 JWK          1557
 ZVKJWK       1553
*ZVKJLB       1580  REV 00
 KJLB         1617
 ZVKJLB       1612
*ZVKJLC       1797  REV 00
 KJLC         1859
 ZVKJLC       1803
*ZVKJLF       18EE  REV 00
 KJLF         194A
 ZVKJLF       1945
*ZVKJLI       1AE5  REV 00
 KJLI         1B47
 STOPTI       1C00
 RUPTI        1D27
 ZVKJLI       1B42
*ZVKJLK       1DF6  REV 00
 KJLK         1E62
 ZVKJLK       1E5D
*ZVKJLL       2030  REV 00
 KJLL         217D
 ZVKJLL       2095
*ZVKJLM       240F  REV 00
 KJLM         2444
 ZVKJLM       243F
*ZVKJLN       2592  REV 00
 KJLN         2609
 ZVKJLN       2604
*ZVKJLP       28F4  REV 00
 KJLP         2991
 ZVKJLP       298C
*ZVKJLR       2BAF  REV 00
 KJLR         2C5A
 ZVKJLR       2C55
*ZVKJFS       2EA6  REV 00
 KJFS         2F1E
 ZVKJFS       2F19
*ZV$GP        3120
 ZV$GP        3120
 ZV$---4      3140
*ZV$HA        314C
 ZV$HA        314C
 ZV$HZ        3156
 ZV$HS        3151
*ZV$HD        3185
 ZV$HD        3185
*ZV$RD        31B7  REV. 7
 ZV$SV1       338C
 ZV$SV2       339C
 ZV$DTP       325E
 ZV$SV3       33AC
 ZV$AF        31C8
 ZV$TTY       31CA
 ZV$TID       31C9
 ZV$CF2       31D3
 ZV$TK        31CF
 ZV$RAK       31D0
 ZV$STI       31D4
 ZV$RCC       31D5
 ZV$DUD       31C6
 ZV$OLB       31D7
 ZV$RCB       31D8
 ZV$NSR       31DC
 ZV$STR       31DA
 ZV$BKF       31DF
 ZV$BRS       31DE
 ZV$RD        31B7
 ZV$DAT       31C6
```

```
  ZV$---5        31E9
  ZV$HM          322D
  ZV$LR          31E3
  ZV$HR          31E6
  ZV$IZ          31F1
  ZV$HRU         31E0
  ZV$HRL         31E1
  ZV$LRU         31E2
  ZV$LRL         31E3
  ZV$HBD         31E4
  ZV$CF1         31D2
  ZV$RMD         31C7
  ZV$MCP         31E5
  HIBAUD         31E4
  ZV$RAW         31D1
  ZV$RDT         33E8
  ZV$CTL         31CE
  ZV$B1          3309
  ZV$TST         343E
  ZV$MDC         3412
  ZV$R99         3610
  ZV$ISA         31EC
  ZV$UIH         31E7
  ZV$ZRO         326B
  ZV$BSH         326D
  ZV$CPU         31CD
  ZV$R50         324B
  ZV$R60         3256
  ZV$RT          354D
  ZV$ALL         31CC
 *MLCHPG         3615   T+V
  MLCHPG         3615
  ENDCHP         3646
*UNLINK MODULE(S)
 ZK$EX
 ZV$TT
 ZV$BR
```