```
0001                * FIG FORTH FOR SERIES-16 MACHINES
0002                ****************************************************************
0003                *
0004                * SERIES-16 FORTH          INTRODUCTION          SERIES-16 FORTH
0005                *
0006                ****************************************************************
0007                *
0008                *                                                    AUGUST 2008
0009                *
0010                *       ORIGINALLY DEVELOPED BY THE
0011                *               FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
0012                *               P.O. BOX 1105
0013                *               SAN CARLOS, CA. 94070
0014                *
0015                *
0016                *       PDP-11 FIG-FORTH IMPLEMENTED BY
0017                *               JOHN S. JAMES
0018                *               P.O. BOX 348
0019                *               BERKELEY, CA. 94701
0020                *               JANUARY 1980
0021                *
0022                *       NOVA FIG-FORTH DEVELOPED BY
0023                *               DR. C. H. TING
0024                *               OFFETE ENTERPRISES
0025                *               1306 S. B ST.
0026                *               SAN MATEO, CA. 94402
0027                *               MAY 1981
0028                *
0029                *       SERIES-16 FORTH WAS DEVELOPED BY
0030                *               ADRIAN WISE
0031                *               HTTP://WWW.SERIES16.ADRIANWISE.CO.UK
0032                *
0033                *       THE CODE WAS LARGELY COPIED FROM THE PDP-11 IMPLEMENTAION
0034                *       WITH SIGNIFICANT SECTIONS, PARTICULARLY THOSE RELATING TO
0035                *       CHARACTER HANDLING ON A WORD-ADDRESSED MACHINE, TAKEN
0036                *       FROM THE NOVA CODE
0037                *
0038                *
0039                *       THIS SYSTEM IS IN THE PUBLIC DOMAIN AND CAN BE USED
0040                *       WITHOUT RESTRICTION.  PLEASE CREDIT THE FORTH INTEREST
0041                *       GROUP IF YOU REPUBLISH SUBSTANTIAL PORTIONS.
0042                *
0043                *
0044                     EJCT
```

```
0045                    *       THE FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
0046                    *       ALSO HAS DEVELOPED NEARLY IDENTICAL VERSIONS OF THIS
0047                    *       SYSTEM FOR THE
0048                    *               8080
0049                    *               6800
0050                    *               6502
0051                    *               9900
0052                    *               PACE
0053                    *               PDP-11
0054                    *               NOVA
0055                    *
0056                    *
0057                    *       FOR MORE INFORMATION, WRITE:
0058                    *
0059                    *               JOHN S. JAMES
0060                    *               P.O. BOX 348
0061                    *               BERKELEY, CA. 94701
0062                    *
0063                    *                       OR
0064                    *
0065                    *               FORTH INTEREST GROUP
0066                    *               P.O. BOX 1105
0067                    *               SAN CARLOS, CA. 94070
0068                    *
0069                    *
0070                    *       THIS FORTH SYSTEM HAS
0071                    *               - FULL LENGTH NAMES
0072                    *               - EXTENSIVE COMPILE-TIME CHECKS AND ERROR MESSAGES
0073                    *               - DOUBLE INTEGER I/O
0074                    *               - LINKED VOCABULARIES
0075                    *               - HOOKS FOR MULTITASKING/MULTIUSER (CURRENTLY
0076                    *                 SINGLE TASK)
0077                    *               - AND AS CURRENTLY CONFIGURED IT CALCULATES THE
0078                    *                 EXTENT OF MEMORY AT START-UP AND WILL RUN,
0079                    *                 WITHOUT DISKS, IN SYSTEMS WITH AS LITTLE AS 4K
0080                    *                 WORDS OF RAM, USING ALL AVAILABLE MEMORY UP TO
0081                    *                 16K, OR 32K WHEN CONFIGURED TO USE EXTENDED
0082                    *                 ADDRESSING. THE CODE IS CONFIGURED TO USE THE
0083                    *                 HIGH SPEED ARITHMETIC OPTION, BUT MAY BE
0084                    *                 ASSEMBLED TO RUN WITHOUT HSA SINCE SOFTWARE
0085                    *                 MULTIPLY AND DIVIDE ROUTINES ARE SUPPLIED.
0086                    *                 THE SYSTEM MAY BE EXTENDED TO USE DISK I/O, BUT
0087                    *                 AT THIS TIME (AUGUST 2008) THIS IS NOT
0088                    *                 IMPLEMENTED.
0089                    *
0090                        EJCT
```

```
0091                *       AT A LATER DATE THE SYSTEM, WITH DISK (OR EQUIVALENT) MAY
0092                *       WELL BE EXTENDED TO ALSO PROVIDE:
0093                *               - A FORTH ASSEMBLER, PERMITTING STRUCTURED,
0094                *                 INTERACTIVE DEVELOPMENT OF DEVICE HANDLERS,
0095                *                 SPEED-CRITICAL ROUTINES, AND LINKAGE TO
0096                *                 OPERATING SYSTEMS OR TO SUBROUTINE PACKAGES
0097                *                 WRITTEN IN OTHER LANGUAGES.
0098                *               - STRING-HANDLING ROUTINES
0099                *               - A STRING-SEARCH EDITOR
0100                *
0101                *
0102                *       IT IS ALIGNED WITH THE 1978 STANDARD OF THE FORTH
0103                *       INTERNATIONAL STANDARDS TEAM.
0104                *
0105                *
0106                *
0107                *       RECOMMENDED DOCUMENTATION:
0108                *               - A FORTH LANGUAGE MANUAL.  WE PARTICULARLY
0109                *                 RECOMMEND EITHER
0110                *                       (A) 'USING FORTH', BY FORTH, INC.
0111                *                             OR
0112                *                       (B) 'A FORTH PRIMER',
0113                *                            BY W. RICHARD STEVENS, KITT PEAK
0114                *                            NATIONAL OBSERVATORY.
0115                *                 EITHER IS AVAILABLE THROUGH
0116                *                 THE FORTH INTEREST GROUP,
0117                *                 P.O. BOX 1105, SAN CARLOS, CA. 94070.
0118                *               - PDP-11 FORTH USER'S GUIDE, AVAILABLE FROM
0119                *                 JOHN S. JAMES, ADDRESS ABOVE.
0120                *               - FORTH REFERENCE CARD FOR THE FORTH
0121                *                 IMPLEMENTATION TEAM COMMON MODEL, AVAILABLE
0122                *                 FROM FIG.
0123                *               - 'FIG-FORTH INSTALLATION MANUAL', ALSO FROM FIG.
0124                *
0125                *
0126                *
0127                *       ACKNOWLEDGMENTS:
0128                *                    THIS FORTH SYSTEM (IN 'FORTH.MAC') IS A GROUP
0129                *               PRODUCT OF THE FORTH IMPLEMENTATION TEAM OF THE
0130                *               FORTH INTEREST GROUP (P.O. BOX 1105, SAN CARLOS
0131                *               CA. 94070).  THE IMPLEMENTER IS RESPONSIBLE FOR
0132                *               THIS SERIES-16 VERSION OF THE MODEL.
0133                    EJCT
```

```
0134                ****************************************************************
0135                *
0136                *               VARIATIONS FROM F.I.G. MODEL
0137                *
0138                ****************************************************************
0139                *
0140                *  THESE DIFFERENCES WERE INHERITED FROM THE PDP-11
0141                *  IMPLEMENTATION:
0142                *
0143                *
0144                *  'FIRST' AND 'LIMIT' HAVE BEEN MADE USER VARIABLES, NOT
0145                *  CONSTANTS. THEREFORE WHEN THEY ARE USED, 'FIRST @' AND
0146                *  'LIMIT @' ARE REQUIRED.
0147                *
0148                *  ';CODE' AND 'FORTH' ARE NOT PURE CODE, SO THEY WERE MOVED TO
0149                *  THE END OF THE DICTIONARY.  THIS IS SO THE BULK OF THE
0150                *  DICTIONARY COULD BE PUT IN PROM OR USED RE-ENTRANTLY.
0151                *
0152                *  THE MACHINE-INDEPENDENT I/O SECTION WAS MOVED TO NEAR THE END
0153                *  OF THE DICTIONARY, BECAUSE IT IS NOT ALWAYS PURE CODE, AND ALSO
0154                *  TO ALLOW THE I/O TO BE REDEFINED WITHOUT REASSEMBLY.
0155                *
0156                *  THIS SYSTEM MUST TEST FOR FIRST-TIME-THROUGH TERMINAL AND DISK
0157                *  I/O, TO AVOID ERRONEOUS ATTEMPT TO OPEN FILES TWICE AT LATER
0158                *  COLD STARTS.  IT CLEARS DISK BUFFERS AT COLD START.
0159                *
0160                *  *****  *****  *****  *****  *****  *****  *****  *****  *****
0161                *
0162                *  THESE DIFFERENCES WERE INHERITED FROM THE NOVA
0163                *  IMPLEMENTATION:
0164                *
0165                *
0166                *  ALL MEMORY REFERENCES ARE CELL ADDRESSING EXCEPT:
0167                *  ENCOSE, CMOVE, C@, C!, -TRAILING, HOLD, (NUMBER), NUMBER
0168                *
0169                *  TRAVERSE IS NOT NEEDED FOR NFA PROCESSING
0170                *
0171                *  BRANCH, 0BRANCH, (LOOP), AND (+LOOP) USE THE ACTUAL
0172                *  DESTINATION ADDRESS, NOT THE OFFSET FROM THE CURRENT
0173                *  ADDRESS
0174                *
0175                *  ?TERMINAL RETURNS TRUE AFTER ANY KEYSTROKE
0176                *
0177                *  ADDED WORDS ARE: BYTE, CELL, U<
0178                *
0179                *  *****  *****  *****  *****  *****  *****  *****  *****  *****
0180                *
0181                *  IN ADDITION THE FOLLOWING DIFFERENCES ARE PECULIAR TO THIS
0182                *  SERIES-16 IMPLEMENTATION:
0183                *
```

```
0184                    *
0185                    *  THE CODE FIELD IS HANDLED DIFFERENTLY, SEE EXPLANATION OF
0186                    *  THREADING BELOW.
0187                    *
0188                    *  ;CODE (ASSEMBLER LABEL 'PSCD') WAS TRADITIONALLY IMMEDIATELY
0189                    *  FOLLOWED, WHERE IT IS REFERENCED IN A COLON DEFINITION,
0190                    *  BY THE ASSEMBLER CODE THAT WAS TO BE USED TO DEFINE THE
0191                    *  PRIMITIVE, SO THAT THE 'RETURN' ADDRESS ON THE RETURN STACK
0192                    *  WAS THE ADDRESS THAT WAS TO BE PLACED IN THE CFA. SINCE THE
0193                    *  USAGE OF CFA DIFFERS IN THIS IMPLEMENTATION (AS DISCUSSED
0194                    *  BELOW) 'DAC PSCD' SHOULD NOW BE FOLLOWED BY THE ONE WORD
0195                    *  INSTRUCTION REQUIRED TO REACH THE ASSEMBLER CODE, TYPICALLY
0196                    *  'JST DOXX', AND THIS WORD IS COPIED INTO THE CFA.
0197                    *
0198                      EJCT
```

```
0199                ****************************************************************
0200                *
0201                *               USE OF REGISTERS
0202                *
0203                ****************************************************************
0204                *
0205                * THE ONLY MACHINE REGISTER WITH A SPECIAL FORTH-RELATED USE
0206                * IS THE INDEX REGISTER. THIS IS USED TO REFER TO THE DATA STACK.
0207                * SINCE THE X REGISTER CANNOT DIRECTLY BE DECREMENTED WHEN VALUES
0208                * ARE PUSHED ONTO THE DATA STACK THE X REGISTER'S VALUE MUST
0209                * ITSELF BE MANIPULATED IN THE ACCUMULATOR. HOWEVER, THE VALUE IN
0210                * THE ACCUMULATOR (THE VALUE TO BE PUSHED) MUST BE SAVED FIRST.
0211                * SO, THE X REGISTER POINTS TO THE FIRST FREE LOCATION ON THE
0212                * STACK (AS OPPOSED TO THE LAST OCCUPIED LOCATION) SO THAT THE
0213                * ACCUMULATOR MAY BE SAVED BY 'STA 0,1' BEFORE THE X REGISTER
0214                * IS DECREMENTED. (IT IS NOT POSSIBLE TO CODE 'STA -1,1')
0215                * TOP-OF-STACK IS ADDRESSED AS '1,1', NEXT-ON-STACK AS '2,1'.
0216                *
0217                * OTHER TRADITIONAL FORTH REGISTERS:
0218                *
0219                * IP - INTERPRETER POINTER
0220                * RP - RETURN STACK POINTER
0221                * UP - USER AREA POINTER
0222                *
0223                * ARE MAINTAINED IN MEMORY WORDS IN SECTOR ZERO (SO THEY CAN
0224                * BE ACCESSED FROM ALL SECTORS).
0225                *
0226                * SINCE THERE IS NO AVAILABLE INDEX REGISTER FOR THE RETURN
0227                * STACK POINTER, THE POINTER ITSELF HAS TO BE MANIPULATED IN THE
0228                * ACCUMULATOR, AND IF A SIMPLE POINTER IS MAINTAINED IN MEMORY
0229                * THEN IT IS DIFFICULT TO PICK UP ANYTHING BUT THE TOP-OF-STACK
0230                * VALUE, WHEN IT IS COMMON (E.G. FOR LOOPING) TO NEED THE TOP TWO
0231                * VALUES. TO ADDRESS THIS, (RP) POINTS TO THE TOP-OF-STACK VALUE
0232                * AND A SECOND POINTER, RP1, IS MAINTAINED POINTING AT THE
0233                * NEXT-ON-STACK.
0234                *
0235                * THE TRADITIONAL FORTH REGISTER 'W', THE WORKING POINTER, DOES
0236                * NOT EXIST IN THIS IMPLEMENTATION.
0237                *
0238                     EJCT
```

```
0239                    ****************************************************************
0240                    *
0241                    *               THREADING METHODOLOGY
0242                    *
0243                    ****************************************************************
0244                    *
0245                    * EARLIER DEVELOPMENT VERSIONS (NEVER RELEASED) OF THIS
0246                    * IMPLEMENTATION USED A MORE TRADIONAL ORGANIZATION USING IP,
0247                    * POINTING AT THE NEXT WORD TO INTERPRET, A 'W' POINTER, AND A
0248                    * CODE FIELD (IN THE WORD'S HEADER) POINTING AT THE CODE TO
0249                    * IMPLEMENT THE PRIMITIVE. HOWEVER, ON AN ACCUMULATOR MACHINE
0250                    * WITH NO GENERAL PURPOSE REGISTERS THIS PROVED VERY CUMBERSOME
0251                    * WITH 'NEXT' REQUIRING ABOUT EIGHT INSTRUCTIONS.
0252                    *
0253                    * IN ORDER TO BETTER TAKE ADVANTAGE OF THE FACILITIES OFFERED BY
0254                    * THE SERIES-16 MACHINES SOME CHANGES WERE MADE.
0255                    *
0256                    * FIRSTLY, THE INTERPRETER POINTER, IP, IS MAINTAINED WITH THE
0257                    * INDIRECT BIT SET. FURTHERMORE, SINCE THERE IS NO POST-INCREMENT
0258                    * ADDRESSING MODE, IP IS MODIFIED TO POINTER TO THE WORD CURRENTLY
0259                    * BEING INTERPRETED, SO THAT IT SHOULD BE INCREMENTED BEFORE
0260                    * IT IS USED. THIS OFFSET-BY-ONE IS TAKEN ACCOUNT OF SO THAT WHEN
0261                    * A RETURN ADDRESS IS PUSHED ONTO THE STACK, FOR EXAMPLE, IT IS
0262                    * FIRST INCREMENTED (AND THE INDIRECT BIT ZEROED) SO THAT THE
0263                    * VALUE WILL BE THE SAME AS IN OTHER FIG-FORTHS.
0264                    *
0265                    * 'NEXT' THERFORE BECOMES THE SEQUENCE:
0266                    *
0267                    * IRS  IP
0268                    * JMP* IP
0269                    *
0270                    * AND A TWO-WORD 'NEXT' IS POSSIBLE, WHICH IS REMARKABLE FOR SUCH
0271                    * A SIMPLE MACHINE.
0272                    *
0273                    * THE NEXT ISSUE IS GETTING A POINTER TO THE PARAMETERS, WHICH
0274                    * WOULD NORMALLY BE IN THE 'W' POINTER. THIS IS ADDRESSED BY
0275                    * MODIFYING WHAT IS PLACED IN THE CODE FIELD OF THE HEADER. THE
0276                    * 'W' POINTER IS ONLY REQUIRED FOR THOSE WORDS WHERE THE ASSEMBLER
0277                    * ROUTINE IS RE-USED MANY TIMES, WITH DIFFERING PARAMETERS. FOR
0278                    * EXAMPLE 'DOCOLON' ('DOCL' HERE, BECAUSE OF THE 4-CHARACTER LIMIT
0279                    * ON LABELS IN THE DAP ASSEMBLER) WHICH EXECUTES A COLON
0280                    * DEFINITION THIS REQUIRES 'W', WHICH POINTS TO THE LIST OF WORDS
0281                    * TO EXECUTE I.E. THE NEW VALUE TO BE PLACED IN 'IP'. SIMILAR WORDS
0282                    * ARE:
0283                    *
0284                    * DOCN - DEAL WITH A CONSTANT
0285                    * DOVR - DEAL WITH A VARIABLE
0286                    * DOUS - DEAL WITH A USER VARIABLE
0287                    * DODS - IMPLEMENT 'DOES>'
0288                        EJCT
```

```
0289                * IN CONTRAST 'W' IS NOT USED BY TRUE PRIMITIVES. FOR EXAMPLE
0290                * THE '+' WORD ADDS THE TOP TWO VALUES ON THE DATA STACK, AND
0291                * HAS NO NEED OF 'W'.
0292                *
0293                * THE CODE FIELD FOR WORDS REQURING 'W' IS FILLED WITH A 'JST'
0294                * INSTRUCTION (I.E. SUBROUTINE CALL) TO THE ACTUAL ASSEMBLER
0295                * ROUTINE (E.G. DOCL, DOCN,...). THIS MEANS THAT THE RETURN
0296                * ADDRESS (STORED IN THE ADDRESS REFERRED TO BY THE 'JST'
0297                * INSTRUCTION) HOLDS THE VALUE THAT WOULD HAVE BEEN IN 'W';
0298                * IT POINTS TO THE PARAMETERS.
0299                *
0300                * CROSS-SECTOR REFERENCE ISSUES ARE AVOIDED BY THE SIMPLE
0301                * EXPEDIENT OF PLACING ALL OF THESE ROUTINES IN SECTOR ZERO.
0302                *
0303                * FOR TRUE PRIMITIVES THE CODE FIELD ISN'T REALLY IDENTIFIABLE
0304                * AS SUCH, SINCE IT IS JUST THE FIRST INSTRUCTION OF THE
0305                * ASSEMBLER DEFINING THE PRIMITIVE. THIS HAS THE ADDITIONAL
0306                * ADVANTAGE THAT ONE WORD IS SAVED IN EACH PRIMITIVE SINCE
0307                * TRADITIONALLY THE CODE FIELD WOULD HAVE HELD A POINTER TO
0308                * THE CODE WHICH WAS LOCATED IN THE FOLLOWING WORD.
0309                *
0310                * DIAGRAMATICALLY:
0311                *
0312                *                                    +----------+
0313                *                 LIST OF WORDS      | NAME     | NFA
0314                *                 CURRENTLY BEING    | ...      |
0315                *                 EXECUTED           | NAME     |
0316                *                  +----------+      +----------+
0317                * IP                | WORD N-1 |      | LINK PTR | LFA
0318                * +----------+      +----------+      +----------+
0319                * |*|        | ---> | WORD N   | ---> | JST DOXX | CFA
0320                * +----------+      +----------+      +----------+
0321                * (INDIRECT         | WORD N+1 |      | PARM. 1  | PFA
0322                *  BIT SET)         +----------+      +----------+
0323                *                   | ...      |      | PARM. 2  |
0324                *                                    +----------+
0325                *                                    | ...      |
0326                *
0327                * SO, WHEN 'NEXT' DOES 'JMP* IP' CONTROL PASSES TO THE
0328                * ADDRESS IN 'WORD N', I.E. THE 'JST DOXX' INSTRUCTION,
0329                * TRANSFERRING CONTROL TO THE 'DOXX' SUBROUTINE (AT
0330                * DOXX+1), AND PLACING THE PFA (THE ADDRESS OF 'PARM. 1')
0331                * IN THE 'DOXX' LOCATION.
0332                *
0333                    EJCT
```

```
0334                ****************************************************************
0335                *
0336                *               MACROS
0337                *
0338                ****************************************************************
0339                *
0340                * THE DAP ASSEMBLER IS VERY POOR AT DEALING WITH STRINGS, AND THE
0341                * MACRO PREPROCESSOR CAN ONLY HANDLE VERY SIMPLE CASES. SO THE
0342                * 'HEAD' MACRO COULD NOT BE EFFICIENTLY DEALT WITH USING THESE
0343                * TOOLS. FOR THIS REASON A SEPARATE DEDICATED MACRO PREPROCESSOR
0344                * WAS WRITTEN TO EXPAND THE 'HEAD' MACRO. THIS IS AVAILABLE
0345                * AS 'HEADMAC.C' AND NEEDS A MORE MODERN MACHINE THAN A
0346                * SERIES-16 - SO IT'S A BIT OF A CHEAT...
0347                *
0348                * 'HEAD' TAKES THREE OR FOUR ARGUMENTS:
0349                *
0350                * (1) A FLAG - NORMALLY 'FNUL', OR 'FIMD' WHICH INDICATES AN
0351                *     IMMEDIATE OPERATION.
0352                * (2) THE NAME OF THE WORD, WHICH IS CONVERTED TO A STRING
0353                * (3) A LABEL USED FOR THE CODE FIELD
0354                * (4) OPTIONALLY A LABEL FOR THE 'DO' ROUTINE. IF NOT PASSED
0355                *     THEN A PRIMITIVE IS BEING DEALT WITH AND NO CODE FIELD
0356                *     IS PRODUCED - THE FOLLOWING ASSEMBLER STARTS AT CFA.
0357                *
0358                * THE HEAD MACRO PRODUCES A FORTH HEADER COMPRISING:
0359                * (1) THE NAME FIELD. ON SERIES-16 MACHINES ASCII CHARACTERS
0360                *     USUALLY HAVE THE TOP BIT SET, SO USAGE IS REVERSED FROM
0361                *     MOST FIG FORTHS - MOST CHARACTERS HAVE THE TOP BIT SET
0362                *     WHILE THE LENGTH BYTE, AND THE LAST BYTE HAVE IT CLEAR.
0363                *     THE IMMEDIATE FLAG IS PLACED IN THE 2^64 BIT OF THE LENGTH
0364                *     BYTE. THE LOWER 6 BITS HOLD THE LENGTH, SINCE THE MAXIMUM
0365                *     NAME LENGTH IS 31 CHARACTERS, THE 2^32 BIT IS NEVER SET
0366                *     AND IS USED, IN THE TRADITIONAL WAY, AS THE SMUDGE BIT.
0367                *     STRINGS ARE PACKED WITH THE EARLIER CHARACTER IN THE
0368                *     MORE SIGNIFICANT BYTE, THE LENGTH BYTE BEING IN THE UPPER
0369                *     BYTE OF THE FIRST WORD.
0370                *     FOR STRINGS OF ODD LENGTH THAT, BECAUSE OF THE LENGTH
0371                *     BYTE, FILL A 16-BIT WORD, THE TOP BIT OF THE LAST
0372                *     CHARACTER IS CLEARED.
0373                *     FOR STRINGS OF EVEN LENGTH THE TOP BIT OF THE LAST ACTUAL
0374                *     CHARACTER (IN THE UPPER BYTE OF THE LAST WORD) IS SET IN
0375                *     THE NORMAL WAY, THE LOWER BYTE IS ALL-ZEROS, AND SO THE
0376                *     LAST WORD CAN BE LOCATED FOR BOTH ODD AND EVEN LENGTH
0377                *     NAMES BY CHECKING THE TOP BIT OF THE LOWER BYTE.
0378                *     SIMILARLY THE FIRST WORD CAN BE IDENTIFIED BY THE MOST
0379                *     SIGNIFICANT BIT OF THE WORD BEING CLEARED.
0380                * (2) THE LINK WORD, POINTING AT THE NFA OF THE PREVIOUS
0381                *     DICTIONARY ENTRY.
0382                * (3) WHERE THE OPTIONAL FOURTH ARGUMENT TO THE MACRO IS USED,
0383                *     A 'JST' TO THE SUPPLIED LABEL.
```

```
0384                    *
0385                    * 'HEADMAC.C' ALSO IMPLEMENTS A SECOND MACRO - 'STRG' THAT
0386                    * CONVERTS ITS SINGLE ARGUMENT TO A STRING WITH LENGTH BYTE. IT
0387                    * IS USED WHERE STRINGS ARE EMBEDDED IN THE PRECOMPILED FORTH.
0388                    *
0389                    * BOTH 'STRG' AND 'HEAD' ALLOW CHARACTERS IN THEIR STRING
0390                    * ARGUMENT TO BE ESCAPED BY PRECEDING THEM BY A BACK-SLASH.
0391                    * BACK-SLASH-COMMA ALLOWS A COMMA TO BE INCLUDED IN THE NAME,
0392                    * (IT IS NOT INTERPRETED AS A SEPARATOR FOR THE NEXT ARGUMENT)
0393                    * BACK-SLASH-SPACE ALLOWS A SPACE (AT THE START OR END OF THE
0394                    * ARGUMENT) TO BE INCLUDED IN THE STRING.
0395                    *
0396                    * THE SINGLE 'PROPER' MACRO IMPLEMENTED USING THE 'MAC'
0397                    * PREPROCESSOR IS 'NEXT' WHICH SIMPLY ASSEMBLES TO:
0398                    *
0399                    * IRS  IP
0400                    * JMP* IP
0401                    *
0402                    ****************************************************************
0403                    *
0404                    *              32-BIT ARITHMETIC
0405                    *
0406                    ****************************************************************
0407                    *
0408                    * ONE FAIRLY SUBSTANTIAL ISSUE WAS HOW TO DEAL WITH 32-BIT
0409                    * ARITHMETIC. THIS IS AN ISSUE BECAUSE THE SERIES-16 MACHINES
0410                    * WERE NOT INTENDED TO PERFORM 32-BIT ARITHMETIC. INSTEAD THEIR
0411                    * NATURAL DOUBLE-WORD FORMAT HOLDS 31-BITS:
0412                    *
0413                    * A REGISTER         B REGISTER
0414                    * +--------------+   +--------------+
0415                    * |S|M.S. 15 BITS|   |0|L.S. 15 BITS|
0416                    * +--------------+   +--------------+
0417                    *
0418                    * THE SIGN BIT OF THE LOWER WORD (OFTEN IN THE B REGISTER) IS
0419                    * ALWAYS ZERO. THIS ISN'T TOTALLY RIDICULOUS AS A FORMAT SINCE
0420                    * WHEN TWO SIGNED 16-BIT NUMBERS ARE MULTIPLIED THEY WILL FIT
0421                    * IN A SIGNED 31-BIT NUMBER (EXCEPT FOR THE SINGLE OVERFLOWING
0422                    * CASE OF -2^15 * -2^15).
0423                    *
0424                    * SOME CONSIDERATION WAS GIVEN TO ONLY IMPLEMENTING 31-BIT DOUBLE
0425                    * WORD ARITHMETIC, BUT IN THE END THIS WAS REJECTED IN FAVOUR OF
0426                    * 32-BIT ARITHMETIC, LIKE OTHER FIG FORTHS - THOUGH INEVITABLY
0427                    * THIS WILL LEAD TO LOSS OF PERFORMANCE IN MATH-DOMINATED
0428                    * APPLICATIONS.
0429                    *
0430                    * ONE MAJOR HURDLE IS THAT THE 16-BIT ADDITION AND SUBTRACTION
0431                    * ROUTINES DO NOT PRODUCE A CARRY BIT - INSTEAD THE SO-CALLED
0432                    * 'C' BIT GETS THE TWO'S COMPLEMENT OVERFLOW. THIS DOES HANG
0433                    * TOGETHER AS A SELF-CONSISTENT SET OF DESIGN DECISIONS BECAUSE
```

```
0434                    * THE CARRY OUT OF THE 15-BIT ADDITION IS AVAILABLE IN THE TOP BIT
0435                    * OF THE RESULT. HOWEVER, IT'S NOT DIRECTLY APPLICABLE TO
0436                    * PERFORMING 32-BIT ADDITION AND SUBTRACTION.
0437                    *
0438                    * THE APPROACH TAKEN IS TO NOTE THAT:
0439                    *
0440                    * IF A, B ARE INPUTS TO THE TOP BIT OF ADDITION AND S IS THE TOP
0441                    * BIT FROM THE SUM, THEN THE CARRY INTO THE TOP BIT CIN=(A^B^S)
0442                    * SINCE OVERFLOW, V=CIN^COUT, IT FOLLOWS THAT COUT=V^CIN
0443                    * SO CARRY CALCULATED AS (A^B^S^V)
0444                    *
0445                    * CIN A B | S V |COUT
0446                    *  0  0 0 | 0 0 |  0
0447                    *  0  0 1 | 1 0 |  0
0448                    *  0  1 0 | 1 0 |  0
0449                    *  0  1 1 | 0 1 |  1
0450                    *  1  0 0 | 1 1 |  0
0451                    *  1  0 1 | 0 0 |  1
0452                    *  1  1 0 | 0 0 |  1
0453                    *  1  1 1 | 1 0 |  1
0454                    *
0455                    * THE SAME PROCEDURE WILL ALSO YIELD BORROW FROM A 16-BIT
0456                    * SUBTRACTION.
0457                    *
0458                    * WHERE THE HIGH SPEED ARITHMETIC (HSA) OPTION IS AVAILABLE
0459                    * IT IS USED TO IMPROVE SPEED OF THE MULTIPLY AND DIVIDE
0460                    * OPERATIONS. HOWEVER, THERE IS INEVITABLY SOME MUCKING
0461                    * AROUND TO CONVERT BETWEEN FORTH'S 32-BIT FORMAT AND THE
0462                    * NATURAL SERIES-16 31-BIT FORMAT, EVEN FOR NUMBERS THAT
0463                    * FIT WITHIN THE 31-BIT RANGE.
0464                    * THE HSA ONLY PROVIDES SIGNED OPERATIONS, BUT UNSIGNED
0465                    * MULTIPLY CAN STILL BE ACCELERATED USING THE SIGNED MULTIPLY
0466                    * OPERATION. NO SATISFACTORY WAY TO USE THE SIGNED DIVIDE
0467                    * OPERATION COULD BE FOUND TO ACCELERATE THE UNSIGNED DIVIDE.
0468                    *
0469                        EJCT
```

```
0470                         CF1                    SHOULD WORK ON 116, 316, 516, 716
0471                         ABS
0472                         SETB  NXTZ
0473                         SUBR  GFORTH,ORGN      GLOBAL LABEL - NORMALLY NOT USED
0474                    ****************************************************************
0475                    *
0476                    * SYSTEM PARAMETERS
0477                    *
0478                    ****************************************************************
0479         000001     HSA  EQU   1               SET TO 1 TO USE HIGH-SPEED ARITHMETIC OPTION
0480         000000     XTND EQU   0               SET TO 1 TO USE EXTENDED ADDRESSING OPTION
0481         000000     DISK EQU   0               SET TO 1 IF HAVE DISK
0482         000001     PTW  EQU   1               SET TO 1 TO INCLUDE PAPERTAPE WORDS
0483                    *
0484         000000     DBGW EQU   0               SET TO 1 TO INCLUDE 'DEBUG' WORD
0485                    *
0486         000000     RSRV EQU   0               WORDS TO RESERVE AT MEMORY TOP
0487                    *
0488         000000     ECHO EQU   0               SET TO 1 IF ECHO TO TERMINAL REQUIRED
0489         000001     ECLF EQU   1               SET TO 1 TO ECHO LF IN RESPONSE TO CR
0490                    *
0491                    * CHARACTER CONSTANTS
0492                    *
0493         000204     CEOT EQU   '204            END OF TRANSMISSION (END OF PAPERTAPE)
0494         000215     CCR  EQU   '215            CARRIAGE RETURN
0495         000212     CLF  EQU   '212            LINE FEED
0496         000210     CBS  EQU   '210            BACKSPACE
0497         000240     CSPC EQU   '240            =' ' SPACE
0498         000242     CDQT EQU   '242            ='"' DOUBLE QUOTE
0499         000251     CRPR EQU   '251            =')' RIGHT PARENTHESIS
0500         000255     CMNS EQU   '255            ='-' MINUS
0501         000256     CDOT EQU   '256            ='.' FULL STOP (PERIOD)
0502         000260     CZRO EQU   '260            ='0' (DIGIT ZERO)
0503         000377     CDEL EQU   '377            DELETE
0504                    *
0505                    * OTHER CONSTANTS
0506                    *
0507         000042     KPAD EQU   34              =68 BYTES
0508                    ****************************************************************
0509                    *
0510                    * VARIABLES
0511                    *
0512                    ****************************************************************
0513                         IFN   XTND
0514                         EXD
0515                         ENDC
0516                         ORG   '100
0517 00100               IP   BSS   1               INTERPRETER POINTER
0518 00101               RP   BSS   1               RETURN STACK POINTER
0519 00102               RP1  BSS   1               RETURN STACK POINTER+1
```

```
0520 00103             UP   BSS   1                USER AREA POINTER
0521                   *
0522                   * TEMPORARIES
0523 00104             T1   BSS   1
0524 00105             T2   BSS   1
0525 00106             T3   BSS   1
0526 00107             T4   BSS   1
0527 00110             T5   BSS   1
0528                   *
0529 00111             CADR BSS   1                USED BY CHARACTER ACCESS ROUTINES
0530                   *
0531                   ****************************************************************
0532                   *
0533                   * MACRO DEFINITIONS
0534                   *
0535                   ****************************************************************
0536                   * NEXT MAC*
0537                   *      IRS   IP
0538                   *      JMP*  IP
0539                   *      ENDM
0540                        EJCT
```

```
0541                    ****************************************************************
0542                    *
0543                    * INNER INTERPRETER - CODE ENDINGS
0544                    *
0545                    ****************************************************************
0546                    *
0547                    * POP AND POP2 DISCARD 1 AND 2 (RESPECTIVELY) OPERANDS FROM THE
0548                    * STACK AND GO TO NEXT
0549                    *
0550 00112   0 12 00000 POP2 IRS   0
0551 00113   0 12 00000 POP  IRS   0
0552                    *
0553                    *    NEXT
0554 00114   0 12 00100      IRS   IP
0555 00115  -0 01 00100      JMP*  IP
0556                    *
0557                    * PUSH PUSHES THE VALUE IN A REG. ONTO THE STACK AND GOES TO
0558                    * NEXT, WHILE 'NEXT' IS AVAILABLE WHERE IT IS JUST USEFUL TO
0559                    * JUMP TO A LOCATION THAT GOES TO NEXT (E.G. UNDER A SKIP
0560                    * OR CAS INSTRUCTION).
0561                    *
0562 00116   1 04 00000 PUSH STA   0,1              X POINTS TO NEXT FREE LOCATION
0563 00117   0 02 00000      LDA   0                NOW DECREMENT X
0564 00120   0 07 00736      SUB   =1
0565 00121   0 04 00000      STA   0
0566                    * FALL THROUGH
0567         000122     NEXT EQU   *
0568                    *
0569                    *    NEXT
0570 00122   0 12 00100      IRS   IP
0571 00123  -0 01 00100      JMP*  IP
0572                    *
0573                    * BINA DISCARDS ONE VALUE FROM THE STACK AND REPLACES THE
0574                    *      VALUE NOW AT TOP-OF-STACK WITH THE VALUE IN A REG.
0575                    *      IT THUS DOES WHAT IS COMMONLY REQUIRED FOR BINARY
0576                    *      OPERATORS
0577                    *
0578                    * PUT  JUST REPLACES TOS WITH A REG. STACK POINTER UNCHANGED
0579                    *
0580 00124   0 12 00000 BINA IRS   0
0581 00125   1 04 00001 PUT  STA   1,1
0582                    *
0583                    *    NEXT
0584 00126   0 12 00100      IRS   IP
0585 00127  -0 01 00100      JMP*  IP
0586                         EJCT
```

```
0587                    * EXECUTE A COLON DEFINITION
0588                    * SAVES CORRECTED IP TO THE RETURN STACK, THEN PICKS UP
0589                    * THE PARAMETER ADDRESS, THAT WOULD COMMONLY BE IN THE
0590                    * W REGISTER, FROM DOCL - WHERE IT WAS PLACED BY THE
0591                    * JST INSTRUCTION AT THE CFA IN THE HEADER - AND
0592                    * JUMPS TO THAT LOCATION
0593                    *
0594 00130  -0 000000   DOCL DAC*  **               INDIRECT BIT SET
0595 00131   0 02 00100      LDA   IP
0596 00132   140100           SSP
0597 00133   141206           AOA
0598 00134   0 10 00220      JST   RPSH
0599 00135   0 02 00130      LDA   DOCL             NOTE INDIRECT BIT IS SET BY DAC*
0600 00136   0 04 00100 NXT1 STA   IP
0601 00137  -0 01 00100      JMP*  IP               EFFECTVELY NEXT
0602                    *
0603                    * EXECUTE A CONSTANT
0604                    * SIMPLY PICKS UP THE VALUE AT THE PARAMETER ADDRESS
0605                    * AND PUSHES IT ONTO THE STACK
0606                    *
0607 00140   0 000000   DOCN DAC   **
0608 00141  -0 02 00140      LDA*  DOCN
0609 00142   0 01 00116      JMP   PUSH
0610                    *
0611                    * EXECUTE A VARIABLE
0612                    * PICKS UP THE ADDRESS OF THE PARAMETER ADDRESS AND
0613                    * PUSHES IT ONTO THE STACK
0614                    *
0615 00143   0 000000   DOVR DAC   **
0616 00144   0 02 00143      LDA   DOVR
0617 00145   0 01 00116      JMP   PUSH
0618                    *
0619                    * EXECUTE A USER VARIABLE
0620                    * PICK UP THE VALUE IN THE PARAMETER ADDRESS, WHICH IS
0621                    * AN OFFSET INTO THE USER AREA, ADD IT ONTO THE USER AREA
0622                    * BASE POINTER AND PUSH THE RESULTING ADDRESS ONTO THE
0623                    * STACK
0624                    *
0625 00146   0 000000   DOUS DAC   **
0626 00147  -0 02 00146      LDA*  DOUS
0627 00150   0 06 00103      ADD   UP
0628 00151   0 01 00116      JMP   PUSH
0629                         EJCT
```

```
0630                    * EXECUTE A 'DOES>'
0631                    * SAVES CORRECTED IP TO THE RETURN STACK, THEN PICKS UP
0632                    * THE FIRST PARAMETER, WHICH WILL BE JUMPED TO, AND
0633                    * THE SECOND PARAMETER, WHICH IS PUSHED ONTO THE STACK
0634                    *
0635 00152   0 000000   DODS DAC   **                NO INDIRECT BIT SET
0636 00153   0 02 00100      LDA   IP
0637 00154   140100           SSP
0638 00155   141206           AOA
0639 00156   0 10 00220      JST   RPSH
0640 00157  -0 02 00152      LDA*  DODS
0641 00160   0 07 00736      SUB   =1
0642 00161   140500           SSM                   SET THE INDIRECT BIT
0643 00162   0 04 00100      STA   IP
0644 00163   0 02 00152      LDA   DODS
0645 00164   141206           AOA
0646 00165   0 01 00116      JMP   PUSH
0647                         EJCT
```

```
0648                    ****************************************************************
0649                    *
0650                    * CHARACTER HANDLING
0651                    *
0652                    ****************************************************************
0653 00166   0 000000   CHGT DAC   **                GET CHARACTER AT ADDRESS
0654 00167   0404 77          LGR   1                 LS BIT GOES TO CARRY
0655 00170   0 04 00111       STA   CADR              SAVE WORD ADDRESS
0656 00171  -0 02 00111       LDA*  CADR              LOAD THROUGH IT
0657 00172   101001           SSC
0658 00173   0404 70          LGR   8                 FIRST BYTE IN UPPER BYTE
0659 00174   0 03 00735       ANA   ='377             LOSE UPPER BYTE
0660 00175  -0 01 00166       JMP*  CHGT
0661                    *
0662 00176   0 000000   CHPT DAC   **                PUT CHARACTER IN B TO CHARACTER ADDRESS
0663 00177   0404 77          LGR   1                 LS BIT GOES TO CARRY
0664 00200   0 04 00111       STA   CADR              SAVE WORD ADDRESS
0665 00201   101001           SSC
0666 00202   0 01 00212       JMP   CHPU
0667                    *
0668                    * PLACE IN LOWER BYTE
0669 00203   000201           IAB                     GET CHARACTER BACK
0670 00204   0 03 00735       ANA   ='377             LOSE UPPER BYTE IF ANY
0671 00205  -0 13 00111       IMA*  CADR              GET EXISTING WORD
0672 00206   0 03 00734       ANA   ='177400          DISCARD LOWER BYTE
0673 00207  -0 05 00111 CHP1  ERA*  CADR
0674 00210  -0 04 00111       STA*  CADR
0675 00211  -0 01 00176       JMP*  CHPT
0676                    *
0677                    * PLACE IN UPPER BYTE
0678 00212   000201     CHPU  IAB                     GET CHARACTER BACK
0679 00213   0 03 00735       ANA   ='377             LOSE UPPER BYTE IF ANY
0680 00214   0414 70          LGL   8                 GET INTO UPPER BYTE
0681 00215  -0 13 00111       IMA*  CADR              GET EXISTING WORD
0682 00216   0 03 00735       ANA   ='377             DISCARD UPPER BYTE
0683 00217   0 01 00207       JMP   CHP1
0684                          EJCT
```

```
0685                    ****************************************************************
0686                    *
0687                    * RETURN STACK
0688                    *
0689                    ****************************************************************
0690                    *
0691                    * PUSH A VALUE ONTO THE RETURN STACK
0692                    *
0693 00220   0 000000   RPSH DAC   **
0694 00221   0 13 00101      IMA   RP               TEMP. SAVE VALUE
0695 00222   0 04 00102      STA   RP1              NEXT-ON-STACK-POINTER
0696 00223   0 07 00736      SUB   =1               DECREMENT POINTER
0697 00224   0 13 00101      IMA   RP               NEW POINTER, GET VALUE TO PUSH BACK
0698 00225  -0 04 00101      STA*  RP
0699 00226  -0 01 00220      JMP*  RPSH
0700                    *
0701                    * POP A VALUE FROM THE RETURN STACK
0702                    *
0703 00227   0 000000   RPOP DAC   **
0704 00230  -0 02 00101      LDA*  RP               GET VALUE
0705 00231   0 13 00102      IMA   RP1              SAVE AND GET RP+1
0706 00232   0 04 00101      STA   RP               WHICH IS NEW VALUE OF RP
0707 00233   141206         AOA                    INCREMENT TO NEW VALUE OF RP1
0708 00234   0 13 00102      IMA   RP1              UPDATE RP1, RETRIEVE VALUE
0709 00235  -0 01 00227      JMP*  RPOP
0710         000236     NXTW EQU   *
0711                         EJCT
```

```
0712                    ****************************************************************
0713                    *
0714                    *               START-UP TABLE
0715                    *
0716                    ****************************************************************
0717                    *
0718                    *  AT STARTUP, MOST OF THESE VALUES ARE MOVED INTO THE USER AREA
0719                    *  (STARTING AT 'XDP:'); THEY ARE NORMALLY ACCESSED THERE.  THE
0720                    *  VALUES HERE ARE  NOT USUALLY CHANGED, BUT THEY MAY BE CHANGED
0721                    *  E.G. TO CONTROL WHAT HAPPENS AT COLD START.  THIS TABLE COULD
0722                    *  BE MOVED OUT OF LOW MEMORY IF NECESSARY FOR ROM SYSTEMS.
0723                    *
0724                    *  LOCATED AT '1000 BECAUSE THAT'S THE TRADITIONAL STARTING
0725                    *  POINT FOR SERIES-16 PROGRAMS
0726                    *
0727                         ORG   '1000
0728 01000   0 01 01025 ORGN JMP   CENT              COLD START ENTRY POINT
0729 01001   0 01 01057      JMP   WENT              WARM START ENTRY ADDRESS
0730                    *
0731                    *  NOTE - COLD START WIPES OUT ANY NEW DICTIONARY DEFINITIONS, AND
0732                    *  THEN DOES A WARM START.  WARM START CLEANS UP STACKS, TERMINAL
0733                    *  BUFFER, ETC.
0734                    *
0735 01002   000020          DEC   16                    CPU
0736 01003   000000          DEC   0                     REVISION
0737 01004   0 005657   OTSK DAC   XTSK              '00 - POINTER TO LATEST WORD DEFINED
0738 01005   000010          OCT   10                '01 - BACKSPACE CHARACTER
0739 01006   0 006116   OUP  DAC   XUP               '02 - POINTER TO USER AREA
0740                    *  NOTE - THE USER AREA IS A HOOK IN THIS SYSTEM TO ALLOW
0741                    *  MULTITASKING TO BE ADDED LATER.
0742 01007   0 006416   OXS0 DAC   XS0               '03 - POINTER TO BEGINNING OF THE STACK
0743 01010   0 006116        DAC   XR0               '04 - POINTER TO BEGINNING OF RETURN STACK
0744 01011   0 005762   OXTB DAC   XTIB              '05 - POINTER TO TERMINAL INPUT BUFFER
0745 01012   000037          DEC   31                '06 - MAXIMUM NAME-FIELD WIDTH, NORMALLY 31
0746 01013   000000          DEC   0                 '07 - WARNING MODE; 0=ERROR, 1=DISK MESSAGE
0747                    *  NOTE - WARNING MODE INITIALIZED TO ZERO, IN CASE DISK ISN'T UP.
0748 01014   0 006216        DAC   XDP               '10 - FENCE TO PROTECT AGAINST ACCIDENTAL
0749                    *                                   'FORGET' OF THE SYSTEM.
0750 01015   0 006216        DAC   XDP               '11 - POINTER TO NEXT AVAILABLE DICTIONARY
0751                    *                                   LOCATION (RETURNED BY 'HERE').
0752 01016   0 005656        DAC   XXVC              '12 - POINTER TO INITIAL VOCABULARY LINK
0753                         IFN   DISK
0754                    O1ST DAC   DSKB              '13 - INITIALIZE 'FIRST'
0755                    OLMT DAC   ENDB              '14 - INITIALIZE 'LIMIT'
0756                         ELSE
0757 01017   000000     O1ST DEC   0                 '13 - INITIALIZE 'FIRST'
0758 01020   000000     OLMT DEC   0                 '14 - INITIALIZE 'LIMIT'
0759                         ENDC
0760 01021   000000          DEC   0                 '15 - AVAILABLE
0761 01022   000000          DEC   0                 '16 - AVAILABLE
```

```
0762                    *
0763 01023   0 001007   XXS0 DAC   OXS0              START OF AREA COPIED
0764 01024   0 005656   X4P4 DAC   FRTH+4
0765                    *
0766                    * ACTUAL COLD ENTRY POINT
0767                    *
0768                    * NOTE THAT THE DICTIONARY ENTRY FOR 'COLD' IS FURTHER
0769                    * ON. THIS CODE IS MOVED HERE SO CROSS-SECTOR LINKS FROM
0770                    * THE ENTRY POINT AT '1000, AND REFERENCES TO THE START-UP
0771                    * TABLE ARE NOT REQUIRED.
0772                    *
0773         001025     CENT EQU   *
0774                         IFN   XTND
0775                         EXA
0776                         ENDC
0777 01025   0 10 05762      JST   MSZ               ONCE ONLY - CALCULATE MEMORY SIZE
0778 01026   0 02 01004      LDA   OTSK              SET 'FORTH' VOCABULARY FROM STARTUP TABLE
0779 01027  -0 04 01024      STA*  X4P4
0780 01030   0 02 01006      LDA   OUP               INITIALIZE USER POINTER
0781 01031   0 04 00103      STA   UP
0782 01032   0 04 00000      STA   0                 AND BORROW INDEX REGISTER FOR INITIALISATION
0783                    *  NOTE - FOR SMALLER STAND-ALONE BOOT, INITIALIZE AREAS IN
0784                    *  HIGH MEMORY WHICH MUST BE INITIALIZED.
0785                    *  CLEAR DISK BUFFERS ON FIRST TIME THROUGH
0786 01033   0 02 01017      LDA   O1ST              GET POINTER TO START OF BUFFERS
0787 01034   101040         SNZ                     ANY DISK BUFFERS?
0788 01035   0 01 01046      JMP   CNT2
0789 01036   0 04 00105      STA   T2
0790 01037   0 07 01020      SUB   OLMT              SUBTRACT END TO GET -VE WORDS
0791 01040   0 04 00104      STA   T1                COUNTER
0792 01041   140040         CRA
0793 01042  -0 04 00105 CNT1 STA*  T2
0794 01043   0 12 00105      IRS   T2                STEP POINTER
0795 01044   0 12 00104      IRS   T1                STEP COUNTER
0796 01045   0 01 01042      JMP   CNT1              LOOP
0797                    *  NOW INITIALIZE 'OUT', 'OFFSET', 'USE' AND 'PREV'
0798                    *  NOTE INDEX REGISTER POINTING AT USER AREA (NOT STACK)
0799 01046   140040     CNT2 CRA
0800 01047   1 04 00021      STA   '21,1             CLEAR 'OUT'
0801 01050   1 04 00023      STA   '23,1             CLEAR 'OFFSET'
0802 01051   0 02 01017      LDA   O1ST
0803 01052   1 04 00035      STA   '35,1             TO 'USE'
0804 01053   1 04 00036      STA   '36,1             TO 'PREV'
0805                    *  END OF SPECIAL HIGH-MEMORY INITIALIZE
0806 01054   0 02 00732      LDA   =-12              ON COLD START, MOVE 12 WORDS
0807 01055   0 01 01061      JMP   WNT1
0808 01056   000000     STOP HLT                    'BYE' COMES HERE
0809                    * SO RESTART GOES TO WARM ENTRY
0810         001057     WENT EQU   *
0811                         IFZ   XTND
```

```
0812 01057   000011          DXA                    BECAUSE START-BUTTON INTERRUPT COME HERE
0813                      *                         AND MAY HAVE FORCED EXTENDED MODE
0814                           ELSE
0815                           EXA                    IN CASE MANUALLY STARTED HERE
0816                           ENDC
0817 01060   0 02 00731        LDA   =-5              ON WARM START, MOVE 5 WORDS
0818 01061   0 04 00104 WNT1   STA   T1
0819 01062   0 02 01023        LDA   XXS0             START WITH INITIAL STACK POINTER
0820 01063   0 04 00105        STA   T2
0821 01064   0 02 00730        LDA   =3               TO AREA 3 WORDS BEYOND
0822 01065   0 06 01006        ADD   OUP              USER POINTER
0823 01066   0 04 00106        STA   T3
0824 01067  -0 02 00105 WNT2   LDA*  T2
0825 01070  -0 04 00106        STA*  T3               COPY WORDS
0826 01071   0 12 00105        IRS   T2               STEP POINTERS
0827 01072   0 12 00106        IRS   T3
0828 01073   0 12 00104        IRS   T1               STEP COUNTER
0829 01074   0 01 01067        JMP   WNT2
0830                      *
0831                      * SET UP VECTOR SO THAT START-BUTTON INTERRUPT
0832                      * GOES TO WARM ENTRY
0833 01075   0 02 01102        LDA   XWNT
0834 01076   0 04 00063        STA   '63
0835 01077   000401           ENB                    ENABLE INTERRUPTS
0836                      *
0837                      *  NOW SET FORTH'S INSTRUCTION COUNTER, AND GO
0838 01100   0 02 01103        LDA   XGO
0839 01101   0 01 00136        JMP   NXT1
0840                      *
0841 01102   0 001057   XWNT   DAC   WENT             TO INITIALISE INTERRUPT VECTOR
0842                      *
0843                      *  NOTE - NORMALLY THE ABOVE INSTRUCTION WOULD JUMP STRAIGHT
0844                      *  TO THE ABORT ROUTINE. IT HAS BEEN CHANGED HERE TO ALLOW USER
0845                      *  TO PATCH A DIFFERENT START-UP.  BUT THE SYSTEM WON'T WORK
0846                      *  UNTIL SOME OF THE WORK OF 'ABORT' HAS BEEN DONE, SO THAT WORK
0847                      *  IS REPEATED.  THE USER CAN PATCH OVER THE 'ABORT' AND THE
0848                      *  ZEROS.
0849                      *
0850 01103  -0 001104   XGO   DAC*  GO
0851 01104   0 001345   GO    DAC   RPST               INITIALIZE RETURN STACK POINTER
0852 01105   0 001332         DAC   SPST               INITIALIZE DATA STACK POINTER
0853 01106   0 002651         DAC   DEC                SELECT DECIMAL
0854 01107   0 005652         DAC   FRTH               FORTH
0855 01110   0 004272         DAC   DFNS               DEFINITIONS
0856 01111   0 004347         DAC   ABRT               ABORT
0857 01112   0 000000         DAC   0
0858 01113   0 000000         DAC   0
0859 01114   0 000000         DAC   0
0860         001115      NXTX EQU   *
0861                           EJCT
```

```
0862                    ****************************************************************
0863                    *
0864                    * CODE DEFINITIONS
0865                    *
0866                    ****************************************************************
0867                         ORG   NXTW
0868         000000     LINK SET   0
0869         000000     FNUL EQU   0
0870         000001     FIMD EQU   1
0871                    *
0872                    * **** LIT ****
0873                    * USED ONLY BY THE COMPILER. PUSH FOLLOWING LITERAL ONTO THE STACK
0874                    *    HEAD  FNUL,LIT,LIT
0875         000236     TLNK SET   *
0876 00236   001714          VFD   1,0,1,FNUL,6,3,8,'314
0877                    *                              =<FNUL,3>,'L'
0878 00237   144524          VFD   8,'311,8,'124     ='IT'
0879 00240   0 000000        DAC   LINK
0880         000236     LINK SET   TLNK
0881         000241     LIT  EQU   *
0882 00241   0 12 00100     IRS   IP
0883 00242   0 02 00100     LDA   IP
0884 00243   140100          SSP                    LOSE INDIRECT BIT
0885 00244   0 04 00104     STA   T1
0886 00245  -0 02 00104     LDA*  T1
0887 00246   0 01 00116     JMP   PUSH
0888                    *
0889                    * **** EXEC ****
0890                    * USED ONLY BY THE COMPILER. EXECUTE FORTH WORD WHOSE ADDRESS IS
0891                    * ON THE STACK
0892                    *    HEAD  FNUL,EXECUTE,EXEC
0893         000247     TLNK SET   *
0894 00247   003705          VFD   1,0,1,FNUL,6,7,8,'305
0895                    *                              =<FNUL,7>,'E'
0896 00250   154305          BCI   2,XECU
     00251   141725
0897 00252   152105          VFD   8,'324,8,'105     ='TE'
0898 00253   0 000236        DAC   LINK
0899         000247     LINK SET   TLNK
0900         000254     EXEC EQU   *
0901 00254   0 12 00000     IRS   0                POP
0902                         IFZ   XTND
0903 00255  -1 01 00000     JMP*  0,1              JMP TO CFA
0904                         ELSE
0905                    * EXTENDED ADDRESSING DELAYS INDEXING UNTIL AFTER ALL INDIR.
0906                         LDA   0,1
0907                         STA   T1
0908                         JMP*  T1
0909                         ENDC
0910                         EJCT
```

```
0911                    * **** BRANCH ****
0912                    * USED ONLY BY THE COMPILER. BRANCH TO THE ADDRESS WHICH FOLLOWS
0913                    *    HEAD  FNUL,BRANCH,BRAN
0914          000256    TLNK SET   *
0915 00256    003302         VFD   1,0,1,FNUL,6,6,8,'302
0916                    *                              =<FNUL,6>,'B'
0917 00257    151301         BCI   2,RANC
     00260    147303
0918 00261    044000         VFD   8,'110             ='H'
0919 00262    0 000247       DAC   LINK
0920          000256    LINK SET   TLNK
0921          000263    BRAN EQU   *
0922 00263    0 02 00100     LDA   IP
0923 00264    140100         SSP
0924 00265    141206         AOA
0925 00266    0 04 00104     STA   T1
0926 00267   -0 02 00104     LDA*  T1
0927 00270    140500         SSM                      INDIRECT BIT
0928 00271    0 01 00136     JMP   NXT1
0929                    *
0930                    * **** 0BRANCH ****
0931                    * USED ONLY BY THE COMPILER. BRANCH TO THE ADDRESS WHICH FOLLOWS
0932                    * IF THE TOP OF STACK IS ZERO (FALSE)
0933                    *    HEAD  FNUL,0BRANCH,ZBRA
0934          000272    TLNK SET   *
0935 00272    003660         VFD   1,0,1,FNUL,6,7,8,'260
0936                    *                              =<FNUL,7>,'0'
0937 00273    141322         BCI   2,BRAN
     00274    140716
0938 00275    141510         VFD   8,'303,8,'110      ='CH'
0939 00276    0 000256       DAC   LINK
0940          000272    LINK SET   TLNK
0941          000277    ZBRA EQU   *
0942 00277    0 12 00000     IRS   0                  POP
0943 00300    1 02 00000     LDA   0,1
0944 00301    100040         SZE
0945 00302    0 01 00312     JMP   ZBR1
0946 00303    0 02 00100     LDA   IP
0947 00304    140100         SSP
0948 00305    141206         AOA
0949 00306    0 04 00104     STA   T1
0950 00307   -0 02 00104     LDA*  T1
0951 00310    140500         SSM
0952 00311    0 01 00136     JMP   NXT1
0953 00312    0 12 00100 ZBR1 IRS   IP
0954                    *
0955                    *    NEXT
0956 00313    0 12 00100     IRS   IP
0957 00314   -0 01 00100     JMP*  IP
0958                         EJCT
```

```
0959                    * **** (LOOP) ****
0960                    * USED ONLY BY THE COMPILER. INCREMENT LOOP INDEX BY 1
0961                    * BRANCH IF BELOW LIMIT
0962                    *    HEAD  FNUL,(LOOP),XLOP
0963          000315    TLNK SET   *
0964 00315    003250         VFD   1,0,1,FNUL,6,6,8,'250
0965                    *                          =<FNUL,6>,'('
0966 00316    146317         BCI   2,LOOP
     00317    147720
0967 00320    024400         VFD   8,'051           =')'
0968 00321    0 000272       DAC   LINK
0969          000315    LINK SET   TLNK
0970          000322    XLOP EQU   *
0971 00322   -0 02 00101     LDA*  RP
0972 00323    141206         AOA                    INCREMENT LOOP VARIABLE
0973 00324   -0 11 00102 XLL1 CAS*  RP1
0974 00325    0 01 00337     JMP   XLL3             [RP]+1 > [RP1]
0975 00326    0 01 00337     JMP   XLL3             [RP]+1 = [RP1]
0976 00327   -0 04 00101 XLL2 STA*  RP               [RP]+1 < [RP1]
0977 00330    0 12 00100     IRS   IP
0978 00331    0 02 00100     LDA   IP
0979 00332    140100         SSP
0980 00333    0 04 00104     STA   T1
0981 00334   -0 02 00104     LDA*  T1               GET JUMP ADDRESS
0982 00335    140500         SSM
0983 00336    0 01 00136     JMP   NXT1             AND LOOP
0984                    *
0985 00337    0 02 00102 XLL3 LDA   RP1              POP 2 VALUES FROM RETURN STACK
0986 00340    141206         AOA
0987 00341    0 04 00101     STA   RP
0988 00342    141206         AOA
0989 00343    0 04 00102     STA   RP1
0990 00344    0 12 00100     IRS   IP               AND SKIP THE BRANCH ADDRESS
0991                    *
0992                    *    NEXT  EXIT THE LOOP
0993 00345    0 12 00100     IRS   IP
0994 00346   -0 01 00100     JMP*  IP
0995                         EJCT
```

```
0996                    * **** (+LOOP) ****
0997                    * USED ONLY BY THE COMPILER. INCREMENT LOOP INDEX BY TOP-OF-STACK
0998                    * CONDITIONALLY BRANCH
0999                    *    HEAD  FNUL,(+LOOP),XPLO
1000          000347    TLNK SET   *
1001 00347    003650         VFD   1,0,1,FNUL,6,7,8,'250
1002                    *                              =<FNUL,7>,'('
1003 00350    125714         BCI   2,+LOO
     00351    147717
1004 00352    150051         VFD   8,'320,8,'051      ='P)'
1005 00353    0 000315       DAC   LINK
1006          000347    LINK SET   TLNK
1007          000354    XPLO EQU   *
1008 00354    1 02 00001     LDA   1,1                GET INCREMENT
1009 00355    100400         SPL                      +VE
1010 00356    0 01 00361     JMP   XLL4
1011 00357   -0 06 00101     ADD*  RP                 CURRENT LOOP COUNT
1012 00360    0 01 00324     JMP   XLL1               SAME COMPARISON AS (LOOP)
1013                    *
1014 00361   -0 06 00101 XLL4 ADD*  RP                CURRENT LOOP COUNT
1015 00362   -0 11 00102     CAS*  RP1
1016 00363    0 01 00327     JMP   XLL2               [RP]-N > [RP1]
1017 00364    0 01 00327     JMP   XLL2               [RP]-N = [RP1]
1018 00365    0 01 00337     JMP   XLL3               [RP]-N < [RP1]
1019                    *
1020                    * **** (DO) ****
1021                    * USED ONLY BY THE COMPILER. SET UP 'DO' LIMIT AND INDEX
1022                    *    HEAD  FNUL,(DO),XDO
1023          000366    TLNK SET   *
1024 00366    002250         VFD   1,0,1,FNUL,6,4,8,'250
1025                    *                              =<FNUL,4>,'('
1026 00367    142317         BCI   1,DO
1027 00370    024400         VFD   8,'051             =')'
1028 00371    0 000347       DAC   LINK
1029          000366    LINK SET   TLNK
1030          000372    XDO  EQU   *
1031 00372    1 02 00002     LDA   2,1
1032 00373    0 10 00220     JST   RPSH
1033 00374    1 02 00001     LDA   1,1
1034 00375    0 10 00220     JST   RPSH
1035 00376    0 01 00112     JMP   POP2
1036                         EJCT
```

```
1037                    * **** I ****
1038                    * RETURN CURRENT LOOP COUNTER TO THE STACK
1039                    *    HEAD  FNUL,I,I
1040          000377    TLNK SET   *
1041 00377    000511         VFD   1,0,1,FNUL,6,1,8,'111
1042                    *                              =<FNUL,1>,'I'
1043 00400    0 000366       DAC   LINK
1044          000377    LINK SET   TLNK
1045          000401    I    EQU   *
1046 00401   -0 02 00101      LDA*  RP
1047 00402    0 01 00116      JMP   PUSH
1048                    *
1049                    * **** DIGIT ****
1050                    * USED BY THE COMPILER
1051                    * (ASCII-DIGIT BASE ==> DIGIT-VALUE TRUE(OR FALSE))
1052                    *    HEAD  FNUL,DIGIT,DIGT
1053          000403    TLNK SET   *
1054 00403    002704         VFD   1,0,1,FNUL,6,5,8,'304
1055                    *                              =<FNUL,5>,'D'
1056 00404    144707         BCI   1,IG
1057 00405    144524         VFD   8,'311,8,'124     ='IT'
1058 00406    0 000377       DAC   LINK
1059          000403    LINK SET   TLNK
1060          000407    DIGT EQU   *
1061 00407    1 02 00002      LDA   2,1              GET ASCII VALUE
1062 00410    0 07 00424      SUB   XZRO
1063 00411    100400          SPL
1064 00412    0 01 00431      JMP   DIGX
1065 00413    0 11 00727      CAS   =9
1066 00414    0 01 00425      JMP   DIGA             A>9
1067 00415    101000          NOP                    A=9
1068 00416    1 11 00001 DIGY CAS   1,1              COMPARE BASE
1069 00417    0 01 00431      JMP   DIGX             A>BASE
1070 00420    0 01 00431      JMP   DIGX             A=BASE
1071 00421    1 04 00002      STA   2,1              SAVE DIGIT VALUE
1072 00422    0 02 00736      LDA   =1
1073 00423    0 01 00125      JMP   PUT
1074 00424    000260    XZRO VFD   16,CZRO
1075                    *
1076 00425    0 07 00726 DIGA SUB   =7               SUBTRACT 'A'-'0'
1077 00426    0 11 00727      CAS   =9               NOW EXPECT TO BE >9
1078 00427    0 01 00416      JMP   DIGY             A>9 - OK
1079 00430    101000          NOP                    A=9
1080 00431    140040    DIGX CRA                     A<9 - ERROR EXIT
1081 00432    0 01 00124      JMP   BINA
1082                         EJCT
```

```
1083                    * **** (FIND) ****
1084                    * USED BY THE COMPILER. FIND A WORD IN THE DICTIONARY
1085                    * (STRING-ADDRESS NFA ==> PFA LENGTH TRUE (OR FALSE))
1086                    * STRING-ADDRESS IS (WORD) ADDRESS OF THE WORD CONTAINING
1087                    * LENGTH BYTE, OF THE STRING BEING SOUGHT. NFA IS THE
1088                    * NAME FIELD ADDRESS OF THE WORD IN THE DICTIONARY WHERE
1089                    * THE SEARCH BEGINS. PFA IS THE PARAMETER FIELD ADDRESS
1090                    * OF THE DICTIONARY ENTRY WHICH IS FOUND. IF WORD NOT
1091                    * FOUND ONLY ONE RESULT (0, FALSE) IS RETURNED.
1092                    *
1093                    *    HEAD  FNUL,(FIND),PFND
1094          000433    TLNK SET   *
1095 00433   003250         VFD   1,0,1,FNUL,6,6,8,'250
1096                    *                              =<FNUL,6>,'('
1097 00434   143311         BCI   2,FIND
     00435   147304
1098 00436   024400         VFD   8,'051              =')'
1099 00437   0 000403       DAC   LINK
1100         000433    LINK SET   TLNK
1101         000440    PFND EQU   *
1102 00440   1 02 00002     LDA   2,1                 PICK UP STRING ADDRESS
1103 00441   0 04 00104     STA   T1
1104 00442   0 04 00105     STA   T2
1105 00443   1 02 00001     LDA   1,1                 PICK UP NFA
1106 00444   0 04 00106     STA   T3
1107                    *
1108 00445  -0 02 00106 FNDL LDA*  T3
1109 00446   0 04 00110     STA   T5                  SAVE FIRST WORD OF NFA
1110 00447  -0 05 00105     ERA*  T2
1111 00450   0 03 00725     ANA   ='037577            LOSE FLAG IN LENGTH BYTE, MS BITS
1112 00451   101040         SNZ
1113 00452   0 01 00467     JMP   FNDS
1114                    *
1115                    * WORDS DON'T MATCH - FIND END OF STRING
1116 00453  -0 02 00106 FNDX LDA*  T3
1117 00454   0 12 00106     IRS   T3                  STEP POINTER
1118 00455   0 03 00724     ANA   ='200               LOOK AT MS BIT OF LOWER BYTE
1119 00456   100040         SZE
1120 00457   0 01 00453     JMP   FNDX
1121 00460  -0 02 00106     LDA*  T3                  PICK UP LINK TO PREVIOUS NFA
1122 00461   101040         SNZ
1123 00462   0 01 00525     JMP   FNDN
1124 00463   0 04 00106     STA   T3
1125 00464   0 02 00104     LDA   T1
1126 00465   0 04 00105     STA   T2                  GO BACK TO START OF DESIRED STRING
1127 00466   0 01 00445     JMP   FNDL
1128                        EJCT
```

```
1129                    * FIRST WORD DOES MATCH - CHECK THE REST
1130 00467  -0 02 00105 FNDS LDA*  T2                GET LENGTH
1131 00470   0404 70         LGR   8                 MOVE TO LOWER BYTE
1132 00471   0 03 00723      ANA   =’077             LOSE FLAGS
1133 00472   140407          TCA                     WE COUNT -N TO -1
1134 00473   0 04 00107      STA   T4
1135 00474   0 01 00507      JMP   FNDZ
1136                    *
1137 00475   0 12 00105 FNDT IRS   T2                STEP POINTERS
1138 00476   0 12 00106      IRS   T3
1139 00477  -0 02 00105      LDA*  T2                PICK UP TWO BYTES FROM EACH STRING
1140 00500  -0 05 00106      ERA*  T3
1141 00501   0 12 00107      IRS   T4                FIRST OF TWO INCREMENTS IN LOOP
1142 00502   100000          SKP
1143 00503   0 01 00521      JMP   FNDY
1144 00504   0 03 00722      ANA   =’077577          LOSE MS BITS
1145 00505   100040          SZE
1146 00506   0 01 00453      JMP   FNDX              DON’T MATCH GO TO NEXT DIRECTORY ENTRY
1147 00507   0 12 00107 FNDZ IRS   T4
1148 00510   0 01 00475      JMP   FNDT
1149                    *
1150                    * MATCH - RETURN REQUIRED VALUES
1151 00511   0 02 00106 FNDM LDA   T3                POINTS AT LAST BYTES OF NAME
1152 00512   0 06 00730      ADD   =3                SKIP OVER LINK AND CODE ADDRESS
1153 00513   1 04 00002      STA   2,1
1154 00514   0 02 00110      LDA   T5                GET SAVED (NFA)
1155 00515   0404 70         LGR   8                 PUT LENGTH IN LOW BYTE
1156 00516   1 04 00001      STA   1,1
1157 00517   0 02 00736      LDA   =1
1158 00520   0 01 00116      JMP   PUSH
1159                    *
1160                    * FIRST IRS SKIPPED - SO ONLY THE TOP
1161                    * BYTE SHOULD BE COMPARED, NOT BOTH
1162 00521   0 03 00721 FNDY ANA   =’077400          LOSE MS BIT AND LOWER BYTE
1163 00522   100040          SZE                     MATCHED
1164 00523   0 01 00453      JMP   FNDX
1165 00524   0 01 00511      JMP   FNDM
1166                    *
1167                    * THE STRINGS DON’T MATCH
1168 00525   140040     FNDN CRA
1169 00526   0 01 00124      JMP   BINA
1170                        EJCT
```

```
1171                    * **** ENCLOSE ****
1172                    * USED BY THE COMPILER. BREAK NEXT WORD OUT OF THE INPUT BUFFER
1173                    * ( CADDR DELIMITER ==> CADDR OFFSET END-OFFSET NEXT-OFFSET)
1174                    *    HEAD  FNUL,ENCLOSE,ENCL
1175          000527    TLNK SET   *
1176 00527   003705         VFD   1,0,1,FNUL,6,7,8,'305
1177                    *                              =<FNUL,7>,'E'
1178 00530   147303         BCI   2,NCLO
     00531   146317
1179 00532   151505         VFD   8,'323,8,'105     ='SE'
1180 00533   0 000433       DAC   LINK
1181          000527    LINK SET   TLNK
1182          000534    ENCL EQU   *
1183 00534   0 02 00000     LDA   0
1184 00535   0 07 00720     SUB   =2
1185 00536   0 04 00000     STA   0                 CREATE SPACE FOR RESULTS
1186 00537   140040         CRA                     SET OFFSET TO ZERO
1187 00540   0 04 00104     STA   T1                USE A TEMPORARY TO COUNT CHARACTERS
1188 00541   1 13 00003     IMA   3,1               CLEAR OFFSET, GET DELIMITER
1189 00542   0 04 00105     STA   T2                BECAUSE STACK LOCATION WILL BE OVERWRITTEN
1190 00543   0 10 00566 ENC1 JST   ENCC
1191 00544   0 01 00561     JMP   ENC4              NULL
1192 00545   0 01 00543     JMP   ENC1              LOOP ON DELIMITERS
1193 00546   0 07 00736     SUB   =1
1194 00547   1 04 00003     STA   3,1               OFFSET
1195 00550   0 10 00566 ENC2 JST   ENCC
1196 00551   0 01 00561     JMP   ENC4              NULL
1197 00552   100000         SKP                     DELIMITER
1198 00553   0 01 00550     JMP   ENC2              LOOP UNTIL DELIMITER
1199 00554   1 04 00001     STA   1,1               NEXT-OFFSET
1200 00555   0 07 00736     SUB   =1
1201 00556   1 04 00002 ENC3 STA   2,1               END-OFFSET
1202                    *
1203                    *    NEXT
1204 00557   0 12 00100     IRS   IP
1205 00560  -0 01 00100     JMP*  IP
1206                    *
1207 00561   1 04 00001 ENC4 STA   1,1               NEXT-OFFSET
1208 00562   1 11 00003     CAS   3,1               CONTAINS THE START-OFFSET
1209 00563   0 01 00556     JMP   ENC3              NOT EQUAL
1210 00564   141206         AOA                     EQUAL - STEP POINTER, FALL THROUGH
1211 00565   0 01 00556     JMP   ENC3              NOT EQUAL
1212                        EJCT
```

```
1213                    * LOOK AT THE CHARACTER AT [CADDR+T1]
1214                    * RETURN TO ONE OF THE FOLLOWING THREE LOCATIONS
1215                    * IN PRIORITY ORDER:
1216                    * +1 - CHARACTER IS NULL (T1 NOT INCREMENTED)
1217                    * +2 - CHARACTER IS DELIMITER (T2)
1218                    * +3 - OTHER CHARACTER
1219                    *
1220                    * T1 IS RETURNED IN THE A REGISTER
1221 00566   0 000000   ENCC DAC   **
1222 00567   0 02 00104      LDA   T1               GET CHARACTER POINTER
1223 00570   1 06 00004      ADD   4,1              ADD BASE ADDRESS
1224 00571   0 10 00166      JST   CHGT             GET THE CHARACTER
1225 00572   101040          SNZ
1226 00573   0 01 00602      JMP   ENCX             NULL EXIT
1227 00574   0 12 00104      IRS   T1               NON-NULL EXITS STEP T1
1228 00575   0 12 00566      IRS   ENCC
1229 00576   0 11 00105      CAS   T2
1230 00577   100000          SKP
1231 00600   0 01 00602      JMP   ENCX             DELIMITER EXIT
1232 00601   0 12 00566      IRS   ENCC             FALL THROUGH FOR NORMAL EXIT
1233 00602   0 02 00104 ENCX LDA   T1
1234 00603  -0 01 00566      JMP*  ENCC
1235                         EJCT
```

```
1236                    *
1237                    *  THE NEXT 4 HEADERS POINT TO INSTALLATION-DEPENDENT TERMINAL I/O
1238                    *  ROUTINES.
1239                    *
1240                    *    HEAD  FNUL,EMIT,EMIT    **** EMIT
1241           000604   TLNK SET   *
1242 00604     002305         VFD   1,0,1,FNUL,6,4,8,'305
1243                    *                            =<FNUL,4>,'E'
1244 00605     146711         BCI   1,MI
1245 00606     052000         VFD   8,'124            ='T'
1246 00607     0 000527       DAC   LINK
1247           000604   LINK SET   TLNK
1248           000610   EMIT EQU   *
1249 00610     0 01 05665     JMP   PEMT
1250                    *    HEAD  FNUL,KEY,KEY      **** KEY
1251           000611   TLNK SET   *
1252 00611     001713         VFD   1,0,1,FNUL,6,3,8,'313
1253                    *                            =<FNUL,3>,'K'
1254 00612     142531         VFD   8,'305,8,'131     ='EY'
1255 00613     0 000604       DAC   LINK
1256           000611   LINK SET   TLNK
1257           000614   KEY  EQU   *
1258 00614     0 01 05717     JMP   PKEY
1259                    *    HEAD  FNUL,?TERMINAL,QTRM **** ?TERMINAL
1260           000615   TLNK SET   *
1261 00615     004677         VFD   1,0,1,FNUL,6,9,8,'277
1262                    *                            =<FNUL,9>,'?'
1263 00616     152305         BCI   3,TERMIN
     00617     151315
     00620     144716
1264 00621     140514         VFD   8,'301,8,'114     ='AL'
1265 00622     0 000611       DAC   LINK
1266           000615   LINK SET   TLNK
1267           000623   QTRM EQU   *
1268 00623     0 01 05736     JMP   PQTR
1269                    *    HEAD  FNUL,CR,CR        **** CR
1270           000624   TLNK SET   *
1271 00624     001303         VFD   1,0,1,FNUL,6,2,8,'303
1272                    *                            =<FNUL,2>,'C'
1273 00625     051000         VFD   8,'122            ='R'
1274 00626     0 000615       DAC   LINK
1275           000624   LINK SET   TLNK
1276           000627   CR   EQU   *
1277 00627     0 01 05745     JMP   PCR
1278                          EJCT
```

```
                          IFN   PTW
                    *     HEAD  FNUL,PTRC,PTRC    **** PTRC - PAPERTAPE READER CHARACTER
          000630    TLNK  SET   *
00630     002320          VFD   1,0,1,FNUL,6,4,8,'320
                    *                           =<FNUL,4>,'P'
00631     152322          BCI   1,TR
00632     041400          VFD   8,'103            ='C'
00633     0 000624        DAC   LINK
          000630    LINK  SET   TLNK
          000634    PTRC  EQU   *
00634     0 01 05755      JMP   PPTC
                          ENDC
                    *
                    * **** CMOVE ****
                    * ( CADDR1 CADDR2 COUNT --- )
                    * ADDRESSES ARE BYTE ADDRESSES NOT CELL (WORD) ADDRESSES
                    *
                    * TODO - SHOULD PROBABLY OPTIMIZE TO MOVE WORDS WHERE POSSIBLE
                    *     HEAD  FNUL,CMOVE,CMOV
          000635    TLNK  SET   *
00635     002703          VFD   1,0,1,FNUL,6,5,8,'303
                    *                           =<FNUL,5>,'C'
00636     146717          BCI   1,MO
00637     153105          VFD   8,'326,8,'105     ='VE'
00640     0 000630        DAC   LINK
          000635    LINK  SET   TLNK
          000641    CMOV  EQU   *
00641     1 02 00001      LDA   1,1               GET COUNT
00642     101040          SNZ
00643     0 01 00663      JMP   CMVX
00644     140407          TCA
00645     0 04 00104      STA   T1
00646     1 02 00002      LDA   2,1               DESTINATION ADDRESS
00647     0 04 00105      STA   T2
00650     1 02 00003      LDA   3,1               SOURCE ADDRESS
00651     0 04 00106      STA   T3
                    *
00652     0 02 00106 CMVL LDA   T3                GET POINTER
00653     0 12 00106      IRS   T3                STEP
00654     0 10 00166      JST   CHGT              GET CHARACTER
00655     000201          IAB
00656     0 02 00105      LDA   T2                GET DEST POINTER
00657     0 12 00105      IRS   T2
00660     0 10 00176      JST   CHPT              PUT CHARACTER
00661     0 12 00104      IRS   T1
00662     0 01 00652      JMP   CMVL
                    *
00663     0 12 00000 CMVX IRS   0                 ALSO USED FOR MOVE EXIT
00664     0 01 00112      JMP   POP2
                          EJCT
```

```
1329                    * **** MOVE ****
1330                    * ( ADDR1 ADDR2 COUNT --- )
1331                    * ADDRESSES ARE CELL (WORD) ADDRESSES
1332                    *
1333                    *    HEAD  FNUL,MOVE,MOVE
1334          000665    TLNK SET   *
1335 00665   002315          VFD   1,0,1,FNUL,6,4,8,'315
1336                    *                              =<FNUL,4>,'M'
1337 00666   147726          BCI   1,OV
1338 00667   042400          VFD   8,'105             ='E'
1339 00670   0 000635        DAC   LINK
1340          000665    LINK SET   TLNK
1341          000671    MOVE EQU   *
1342 00671   1 02 00001      LDA   1,1               GET COUNT
1343 00672   101040          SNZ
1344 00673   0 01 00663      JMP   CMVX
1345 00674   140407          TCA
1346 00675   0 04 00104      STA   T1
1347 00676   1 02 00002      LDA   2,1               DESTINATION ADDRESS
1348 00677   0 04 00105      STA   T2
1349 00700   1 02 00003      LDA   3,1               SOURCE ADDRESS
1350 00701   0 04 00106      STA   T3
1351                    *
1352 00702  -0 02 00106 MOVL LDA*  T3
1353 00703   0 12 00106      IRS   T3
1354 00704  -0 04 00105      STA*  T2
1355 00705   0 12 00105      IRS   T2
1356 00706   0 12 00104      IRS   T1
1357 00707   0 01 00702      JMP   MOVL
1358 00710   0 01 00663      JMP   CMVX
1359                    *
1360                    * FIRST SECTOR FULL - LEAVING SPACE FOR CONSTANT POOL
1361                    * AND DESECTORIZING
1362                    *
1363          000711    NXTY EQU   *
1364                         EJCT
```

```
                         ORG   NXTX              AFTER START-UP CODE
                    * **** U* ****
                    * ( N1 N2 --- D ).  PRODUCT IS 32-BIT DOUBLE INTEGER,
                    *    HEAD  FNUL,U*,USTR
            001115  TLNK SET   *
01115       001325         VFD   1,0,1,FNUL,6,2,8,'325
                    *                           =<FNUL,2>,'U'
01116       025000         VFD   8,'052          ='*'
01117       0 000665       DAC   LINK
            001115  LINK SET   TLNK
            001120  USTR EQU   *
                         IFZ   HSA
                    * SOFTWARE UNSIGNED MULTIPLY
                         LDA   =-16              COUNTER
                         STA   T1
                         CRA                     CLEAR ACCUMULATOR
                         IAB
                         CRA
                         STA   T3                MS WORD OF MULTIPLIER
                    UST1 LLL   1                 SHIFT ACCUMULATOR LEFT
                         IMA   2,1               GET MULTIPLICAND
                         LGL   1                 TOP BIT TO CARRY
                         IMA   2,1               GET ACCUMULATOR BACK
                         SSC                     CARRY SET?
                         JMP   UST2              NO
                         IAB                     YES - LS WORD TO A
                         STA   T2                SAVE VALUE
                         ADD   1,1               ADD MULIPLIER
                         IMA   T2                SAVE SUM, GET FIRST INPUT
                         SRC                     CARRY (OVERFLOW!) SET?
                         CHS                     YES - XOR IT INTO MSB
                         ERA   1,1               XOR IN SECOND
                         ERA   T2                XOR IN SUM, CARRY NOW IN MSB
                         CSA                     PUT IN CBIT
                         LDA   T2                GET SUM BACK
                         IAB                     GET MS BITS BACK
                         ACA                     ADD IN ANY CARRY NEEDED
                         ADD   T3                ADD IN MS WORD (FOR SIGNED MULTIPLY)
                    UST2 IRS   T1
                         JMP   UST1
                         STA   1,1               MS WORD
                         IAB
                         STA   2,1               LS WORD
                    *
                    *    NEXT
                         IRS   IP
                         JMP*  IP
                         ELSE
                         EJCT
```

```
1414                    * HARDWARE UNSIGNED MULTIPLY
1415                    * ONLY HAVE SIGNED MULTIPLY INSTRUCTION SO THIS CODE WORKS
1416                    * BY BREAKING EACH 16-BIT UNSIGNED NUMBER INTO TWO FIELDS;
1417                    * THE LOWER 15 BITS, WHICH (BEING A VALID SIGNED NUMBER) CAN
1418                    * BE MULTIPLIED, AND THE TOP BIT, WHICH HAS SIGNIFICANCE
1419                    * 2^15. SO WE HAVE:
1420                    * (P1*2^15+P[2-16]) * (Q1*2^15+Q[2-16]) WHICH EQUALS...
1421                    * P1.Q1*2^30 + P1*2^15*Q[2-16] + Q1*2^15*P[2-16] + P[2-16]*Q[2-16]
1422 01120   1 02 00001       LDA   1,1              ALL OF P
1423 01121   140100            SSP
1424 01122   0 04 00104       STA   T1               BITS 2-16 OF P
1425 01123   000201            IAB
1426 01124   1 02 00002       LDA   2,1              ALL OF Q
1427 01125   140100            SSP
1428 01126   0 04 00105       STA   T2               BITS 2-16 OF Q
1429 01127   0 16 00104       MPY   T1               P[2-16]*Q[2-16]
1430 01130   0 04 00106       STA   T3               SAVE UPPER BITS
1431 01131   000201            IAB                   GET LOWER 15 BITS (B1=0)
1432 01132   0 04 00107       STA   T4               SAVE
1433 01133   140040            CRA                   CLEAR B REGISTER
1434 01134   000201            IAB
1435                    *
1436 01135   1 02 00001       LDA   1,1              CHECK TOP BITS
1437 01136   101400            SMI
1438 01137   0 01 01164       JMP   UST5
1439 01140   1 02 00002       LDA   2,1
1440 01141   101400            SMI
1441 01142   0 01 01161       JMP   UST4
1442                    *
1443                    * BOTH TOP BITS SET
1444 01143   0 02 00106       LDA   T3               UPPER BITS OF P[2-16]*Q[2-16]
1445 01144   0 06 00104       ADD   T1               2^15*P[2-16] (2^15 FREE, SINCE 15-BITS IN B)
1446                    * CAN'T OVERFLOW TO THIS POINTS SINCE BOTH
1447                    * INPUTS TO PREVIOUS ADD WERE 15-BITS, BUT NOW HAVE
1448                    * A 16-BIT VALUE AND ADDING A THIRD 15-BIT VALUE MAY
1449                    * OVERFLOW
1450 01145   100400            SPL
1451 01146   0 01 01152       JMP   UST2
1452                    * TOP BIT CLEAR - CANNOT OVERFLOW
1453 01147   0 06 00105       ADD   T2               2^15*Q[2-16]
1454 01150   0400 77    UST1 LRL   1                LS BIT OF A SHIFTS INTO B1, A1=0
1455 01151   0 01 01157       JMP   UST3
1456                    * TOP BIT WAS SET
1457 01152   0 06 00105 UST2 ADD   T2               2^15*Q[2-16]
1458 01153   100400            SPL
1459 01154   0 01 01150       JMP   UST1             IT STILL IS - NO OVERFLOW
1460 01155   0400 77          LRL   1
1461 01156   140500            SSM                   SET A1, DUE TO CARRY FROM ADD
1462 01157   0 06 00717 UST3 ADD   ='040000         2^30
1463 01160   0 01 01174       JMP   UST8
```

```
1464                    *
1465                    * ONLY TOP BIT OF P SET
1466 01161   0 02 00106 UST4 LDA   T3               UPPER BITS OF P[2-16]*Q[2-16]
1467 01162   0 06 00105      ADD   T2               2^15*Q[2-16]
1468 01163   0 01 01173      JMP   UST7
1469                    *
1470 01164   1 02 00002 UST5 LDA   2,1
1471 01165   101400           SMI
1472 01166   0 01 01172      JMP   UST6
1473                    *
1474                    * ONLY TOP BIT OF Q SET
1475 01167   0 02 00106      LDA   T3               UPPER BITS OF P[2-16]*Q[2-16]
1476 01170   0 06 00104      ADD   T1               2^15*P[2-16]
1477 01171   0 01 01173      JMP   UST7
1478                    *
1479                    * NEITHER UPPER BIT SET
1480 01172   0 02 00106 UST6 LDA   T3               UPPER BITS OF P[2-16]*Q[2-16]
1481 01173   0400 77    UST7 LRL   1
1482 01174   1 04 00001 UST8 STA   1,1              MS WORD
1483 01175   000201           IAB                   GET BACK 2^15 SIGNIFICANCE BIT IN MS BIT
1484 01176   0 05 00107      ERA   T4               OR IN THE LOWER 15 BITS
1485 01177   1 04 00002      STA   2,1              LS WORD
1486                    *
1487                    *    NEXT
1488 01200   0 12 00100      IRS   IP
1489 01201  -0 01 00100      JMP*  IP
1490                         ENDC
1491                         EJCT
```

```
1492                    * **** U/ ****
1493                    * ( D N --- N1 N2 )
1494                    * UNSIGNED DIVIDE YIELDING REMAINDER AND QUOTIENT
1495                    *    HEAD  FNUL,U/,USLA
1496           001202   TLNK SET   *
1497 01202     001325         VFD   1,0,1,FNUL,6,2,8,'325
1498                    *                          =<FNUL,2>,'U'
1499 01203     027400         VFD   8,'057           ='/'
1500 01204     0 001115       DAC   LINK
1501           001202   LINK SET   TLNK
1502           001205   USLA EQU   *
1503 01205     0 10 01210     JST   DIVU
1504                    *
1505                    *    NEXT
1506 01206     0 12 00100     IRS   IP
1507 01207    -0 01 00100     JMP*  IP
1508                    *
1509                    * UNSIGNED DIVIDE - SUBROUTINE SINCE CALLED
1510                    * BY SIGNED DIVIDE
1511                    *
1512 01210     0 000000 DIVU  DAC   **
1513 01211     0 02 00716     LDA   =-16
1514 01212     0 04 00104     STA   T1              COUNT
1515 01213     1 02 00001     LDA   1,1             DIVISOR
1516 01214     101040         SNZ
1517 01215     0 01 01267     JMP   DVU6            DIVIDE BY ZERO
1518 01216     1 07 00002     SUB   2,1             COMPARE TO DIVIDEND
1519 01217     101040         SNZ
1520 01220     0 01 01267     JMP   DVU6            BAD DIVIDE
1521                    * DON'T NEED SUM, WANT CARRY
1522 01221     100001         SRC                   CARRY (OVERFLOW!) SET?
1523 01222     140024         CHS                   YES - XOR IT INTO MSB
1524 01223     1 05 00001     ERA   1,1             XOR IN FIRST OPERAND
1525 01224     1 05 00002     ERA   2,1             XOR IN SECOND OPERAND, CARRY NOW IN MSB
1526 01225     100400         SPL
1527 01226     0 01 01267     JMP   DVU6            BAD DIVIDE
1528 01227     1 02 00003     LDA   3,1             LS WORD
1529 01230     000201         IAB
1530 01231     1 02 00002     LDA   2,1             MS WORD
1531 01232     0410 77  DVU1  LLL   1               SHIFT ACCUMULATOR LEFT
1532 01233     0 04 00105     STA   T2              SAVE IN CASE OF RESTORE
1533 01234     100001         SRC                   TOP BIT WAS SET?
1534 01235     0 01 01264     JMP   DVU5            YES
1535 01236     1 07 00001     SUB   1,1             DIVISOR
1536 01237     0 04 00106     STA   T3              SAVE SUM
1537 01240     100001         SRC                   CARRY (OVERFLOW!) SET?
1538 01241     140024         CHS                   YES - XOR IT INTO MSB
1539 01242     1 05 00001     ERA   1,1             XOR IN SECOND OPERAND
1540 01243     0 05 00105     ERA   T2              XOR IN FIRST OPERAND, CARRY NOW IN MSB
1541 01244     100400         SPL
```

```
1542 01245   0 01 01262      JMP   DVU4              CARRY SET
1543 01246   000201     DVU2 IAB
1544 01247   141206           AOA                    QUOTIENT BIT IS 1
1545 01250   000201           IAB
1546 01251   0 02 00106      LDA   T3
1547 01252   0 12 00104 DVU3 IRS   T1
1548 01253   0 01 01232      JMP   DVU1
1549 01254   140200           RCB                    NO ERROR
1550 01255   0 12 00000      IRS   0                 DISCARD DIVISOR
1551 01256   1 04 00002      STA   2,1               REMAINDER
1552 01257   000201          IAB
1553 01260   1 04 00001      STA   1,1               QUOTIENT
1554 01261  -0 01 01210      JMP*  DIVU
1555                  *
1556 01262   0 02 00105 DVU4 LDA   T2                RESTORE, QUOTIENT BIT ZERO
1557 01263   0 01 01252      JMP   DVU3
1558                  *
1559                  * HERE IF TOP BIT SHIFTED OUT WAS 1 SO
1560                  * SUBTRACTION CANNOT GENERATE CARRY
1561 01264   1 07 00001 DVU5 SUB   1,1               DIVISOR
1562 01265   0 04 00106      STA   T3
1563 01266   0 01 01246      JMP   DVU2
1564                  *
1565 01267   0 12 00000 DVU6 IRS   0                 ERROR EXIT - DISCARD DIVISOR
1566 01270   140040           CRA                    RETURN ZEROS
1567 01271   1 04 00002      STA   2,1               REMAINDER
1568 01272   1 04 00001      STA   1,1               QUOTIENT
1569 01273   140600          SCB                    SET CARRY SO CALLER CAN DETECT OVERFLOW
1570 01274  -0 01 01210      JMP*  DIVU
1571                         EJCT
```

```
1572                    * **** AND ****
1573                    * ( N1 N2 --- N3 ) BITWISE AND
1574                    *    HEAD  FNUL,AND,AND
1575          001275    TLNK SET   *
1576 01275    001701         VFD   1,0,1,FNUL,6,3,8,'301
1577                    *                              =<FNUL,3>,'A'
1578 01276    147104         VFD   8,'316,8,'104      ='ND'
1579 01277    0 001202       DAC   LINK
1580          001275    LINK SET   TLNK
1581          001300    AND  EQU   *
1582 01300    1 02 00001     LDA   1,1
1583 01301    1 03 00002     ANA   2,1
1584 01302    0 01 00124     JMP   BINA
1585                    *
1586                    * **** OR ****
1587                    * ( N1 N2 --- N3 ) BITWISE INCLUSIVE OR
1588                    *    HEAD  FNUL,OR,OR
1589          001303    TLNK SET   *
1590 01303    001317         VFD   1,0,1,FNUL,6,2,8,'317
1591                    *                              =<FNUL,2>,'O'
1592 01304    051000         VFD   8,'122             ='R'
1593 01305    0 001275       DAC   LINK
1594          001303    LINK SET   TLNK
1595          001306    OR   EQU   *
1596 01306    1 02 00001     LDA   1,1
1597 01307    0 05 00715     ERA   ='177777
1598 01310    1 03 00002     ANA   2,1
1599 01311    1 05 00001     ERA   1,1
1600 01312    0 01 00124     JMP   BINA
1601                    *
1602                    * **** XOR ****
1603                    * ( N1 N2 --- N3 ) BITWISE EXCLUSIVE OR
1604                    *    HEAD  FNUL,XOR,XOR
1605          001313    TLNK SET   *
1606 01313    001730         VFD   1,0,1,FNUL,6,3,8,'330
1607                    *                              =<FNUL,3>,'X'
1608 01314    147522         VFD   8,'317,8,'122      ='OR'
1609 01315    0 001303       DAC   LINK
1610          001313    LINK SET   TLNK
1611          001316    XOR  EQU   *
1612 01316    1 02 00001     LDA   1,1
1613 01317    1 05 00002     ERA   2,1
1614 01320    0 01 00124     JMP   BINA
1615                         EJCT
```

```
1616                    * **** SP@ ****
1617                    * ( --- N ) CURRENT STACK POINTER
1618                    *    HEAD  FNUL,SP@,SPAT
1619          001321    TLNK SET   *
1620 01321    001723         VFD   1,0,1,FNUL,6,3,8,'323
1621                    *                              =<FNUL,3>,'S'
1622 01322    150100         VFD   8,'320,8,'100      ='P@'
1623 01323    0 001313       DAC   LINK
1624          001321    LINK SET   TLNK
1625          001324    SPAT EQU   *
1626 01324    0 02 00000     LDA   0                  GET POINTER
1627 01325    141206         AOA                      INCREMENT BECAUSE POINTS TO FIRST FREE
1628 01326    0 01 00116     JMP   PUSH
1629                    *
1630                    * **** SP! ****
1631                    * ( --- ) INITIALISE STACK POINTER
1632                    *    HEAD  FNUL,SP!,SPST
1633          001327    TLNK SET   *
1634 01327    001723         VFD   1,0,1,FNUL,6,3,8,'323
1635                    *                              =<FNUL,3>,'S'
1636 01330    150041         VFD   8,'320,8,'041      ='P!'
1637 01331    0 001321       DAC   LINK
1638          001327    LINK SET   TLNK
1639          001332    SPST EQU   *
1640 01332    0 02 00103     LDA   UP
1641 01333    0 06 00730     ADD   ='3                OFFSET 3 IN USER AREA
1642 01334    0 04 00104     STA   T1                 POINT TO LOCATION IN TABLE
1643 01335   -0 02 00104     LDA*  T1                 GET VALUE
1644 01336    0 07 00736     SUB   =1                 BECAUSE POINTS TO FIRST FREE
1645 01337    0 04 00000     STA   0
1646                    *
1647                    *    NEXT
1648 01340    0 12 00100     IRS   IP
1649 01341   -0 01 00100     JMP*  IP
1650                         EJCT
```

```
1651                    * **** RP! ****
1652                    * ( --- ) INITIALISE RETURN STACK POINTER
1653                    *    HEAD  FNUL,RP!,RPST
1654          001342    TLNK SET   *
1655 01342   001722         VFD   1,0,1,FNUL,6,3,8,'322
1656                    *                              =<FNUL,3>,'R'
1657 01343   150041         VFD   8,'320,8,'041       ='P!'
1658 01344   0 001327       DAC   LINK
1659          001342    LINK SET   TLNK
1660          001345    RPST EQU   *
1661 01345   0 02 00103     LDA   UP
1662 01346   0 06 00714     ADD   ='4                 OFFSET 4 IN USER AREA
1663 01347   0 04 00104     STA   T1                  POINT TO LOCATION IN TABLE
1664 01350  -0 02 00104     LDA*  T1                  GET VALUE
1665 01351   0 04 00101     STA   RP
1666 01352   141206         AOA
1667 01353   0 04 00102     STA   RP1
1668                    *
1669                    *    NEXT
1670 01354   0 12 00100     IRS   IP
1671 01355  -0 01 00100     JMP*  IP
1672                    *
1673                    * **** ;S ****
1674                    * ( --- N ) RETURN?
1675                    *    HEAD  FNUL,;S,SMIS
1676          001356    TLNK SET   *
1677 01356   001273         VFD   1,0,1,FNUL,6,2,8,'273
1678                    *                              =<FNUL,2>,';'
1679 01357   051400         VFD   8,'123              ='S'
1680 01360   0 001342       DAC   LINK
1681          001356    LINK SET   TLNK
1682          001361    SMIS EQU   *
1683 01361   0 10 00227     JST   RPOP
1684 01362   0 07 00736     SUB   =1
1685 01363   140500         SSM
1686 01364   0 04 00100     STA   IP
1687                    *
1688                    *    NEXT
1689 01365   0 12 00100     IRS   IP
1690 01366  -0 01 00100     JMP*  IP
1691                         EJCT
```

```
1692                    * **** LEAVE ****
1693                    * ( --- )
1694                    *    HEAD  FNUL,LEAVE,LEAV
1695          001367    TLNK SET   *
1696 01367    002714         VFD   1,0,1,FNUL,6,5,8,'314
1697                    *                              =<FNUL,5>,'L'
1698 01370    142701         BCI   1,EA
1699 01371    153105         VFD   8,'326,8,'105      ='VE'
1700 01372    0 001356       DAC   LINK
1701          001367    LINK SET   TLNK
1702          001373    LEAV EQU   *
1703 01373   -0 02 00101     LDA*  RP
1704 01374   -0 04 00102     STA*  RP1
1705                    *
1706                    *    NEXT
1707 01375    0 12 00100     IRS   IP
1708 01376   -0 01 00100     JMP*  IP
1709                    *
1710                    * **** >R ****
1711                    * ( N --- )
1712                    *    HEAD  FNUL,>R,TOR
1713          001377    TLNK SET   *
1714 01377    001276         VFD   1,0,1,FNUL,6,2,8,'276
1715                    *                              =<FNUL,2>,'>'
1716 01400    051000         VFD   8,'122             ='R'
1717 01401    0 001367       DAC   LINK
1718          001377    LINK SET   TLNK
1719          001402    TOR  EQU   *
1720 01402    1 02 00001     LDA   1,1
1721 01403    0 12 00000     IRS   0
1722 01404    0 10 00220     JST   RPSH
1723                    *
1724                    *    NEXT
1725 01405    0 12 00100     IRS   IP
1726 01406   -0 01 00100     JMP*  IP
1727                         EJCT
```

```
1728                    *
1729                    * **** R> ****
1730                    * ( --- R )
1731                    *    HEAD  FNUL,R>,FRMR
1732          001407    TLNK SET   *
1733 01407   001322          VFD   1,0,1,FNUL,6,2,8,'322
1734                    *                              =<FNUL,2>,'R'
1735 01410   037000          VFD   8,'076             ='>'
1736 01411   0 001377        DAC   LINK
1737          001407    LINK SET   TLNK
1738          001412    FRMR EQU   *
1739 01412   0 10 00227      JST   RPOP
1740 01413   0 01 00116      JMP   PUSH
1741                    *
1742                    * **** R ****
1743                    * ( --- N )
1744                    *    HEAD  FNUL,R,R
1745          001414    TLNK SET   *
1746 01414   000522          VFD   1,0,1,FNUL,6,1,8,'122
1747                    *                              =<FNUL,1>,'R'
1748 01415   0 001407        DAC   LINK
1749          001414    LINK SET   TLNK
1750          001416    R    EQU   *
1751 01416  -0 02 00101      LDA*  RP
1752 01417   0 01 00116      JMP   PUSH
1753                    *
1754                    * **** 0= ****
1755                    * ( N --- N )
1756                    *    HEAD  FNUL,0=,ZEQU
1757          001420    TLNK SET   *
1758 01420   001260          VFD   1,0,1,FNUL,6,2,8,'260
1759                    *                              =<FNUL,2>,'0'
1760 01421   036400          VFD   8,'075             ='='
1761 01422   0 001414        DAC   LINK
1762          001420    LINK SET   TLNK
1763          001423    ZEQU EQU   *
1764 01423   1 02 00001      LDA   1,1
1765 01424   100040          SZE                      IS IT ZERO?
1766 01425   0 01 01430      JMP   ZEQ0
1767 01426   0 02 00736 ZEQ1 LDA   =1                 RETURN TRUE
1768 01427   0 01 00125      JMP   PUT
1769 01430   140040    ZEQ0 CRA                      RETURN FALSE
1770 01431   0 01 00125      JMP   PUT
1771                         EJCT
```

```
1772                    * **** 0< ****
1773                    * ( N --- N )
1774                    *    HEAD  FNUL,0<,ZLES
1775         001432     TLNK SET   *
1776 01432   001260          VFD   1,0,1,FNUL,6,2,8,'260
1777                    *                              =<FNUL,2>,'0'
1778 01433   036000          VFD   8,'074             ='<'
1779 01434   0 001420        DAC   LINK
1780         001432     LINK SET   TLNK
1781         001435     ZLES EQU   *
1782 01435   1 02 00001     LDA   1,1
1783 01436   101400          SMI
1784 01437   0 01 01430     JMP   ZEQ0
1785 01440   0 01 01426     JMP   ZEQ1
1786                    *
1787                    * **** + ****
1788                    * ( N N --- N )
1789                    *    HEAD  FNUL,+,PLUS
1790         001441     TLNK SET   *
1791 01441   000453          VFD   1,0,1,FNUL,6,1,8,'053
1792                    *                              =<FNUL,1>,'+'
1793 01442   0 001432        DAC   LINK
1794         001441     LINK SET   TLNK
1795         001443     PLUS EQU   *
1796 01443   1 02 00001     LDA   1,1
1797 01444   1 06 00002     ADD   2,1
1798 01445   0 01 00124     JMP   BINA
1799                    *
1800                    * **** D+ ****
1801                    * ( D D --- D )
1802                    *    HEAD  FNUL,D+,DPLS
1803         001446     TLNK SET   *
1804 01446   001304          VFD   1,0,1,FNUL,6,2,8,'304
1805                    *                              =<FNUL,2>,'D'
1806 01447   025400          VFD   8,'053             ='+'
1807 01450   0 001441        DAC   LINK
1808         001446     LINK SET   TLNK
1809         001451     DPLS EQU   *
1810 01451   1 02 00004     LDA   4,1                ADD LOWER WORDS
1811 01452   1 06 00002     ADD   2,1
1812 01453   1 13 00004     IMA   4,1                GET BACK FIRST INPUT, SAVE SUM
1813 01454   100001          SRC                      CARRY (OVERFLOW!) SET?
1814 01455   140024          CHS                      YES - XOR IT INTO MSB
1815 01456   1 05 00002     ERA   2,1                XOR IN SECOND
1816 01457   1 05 00004     ERA   4,1                XOR IN SUM
1817                    * TOP BIT IS NOW CARRY
1818 01460   140320          CSA                      PLACE IN CARRY
1819 01461   1 02 00003     LDA   3,1                ADD UPPER WORDS
1820 01462   141216          ACA                      CARRY FROM LOWER
1821 01463   1 06 00001     ADD   1,1
```

```
1822 01464   0 12 00000      IRS   0                DISCARD A WORD
1823 01465   0 01 00124      JMP   BINA
1824                      *
1825                      * **** MINUS ****
1826                      * ( N --- N )
1827                      *    HEAD  FNUL,MINUS,MINS
1828         001466       TLNK SET   *
1829 01466   002715            VFD   1,0,1,FNUL,6,5,8,'315
1830                      *                           =<FNUL,5>,'M'
1831 01467   144716            BCI   1,IN
1832 01470   152523            VFD   8,'325,8,'123      ='US'
1833 01471   0 001446          DAC   LINK
1834         001466       LINK SET   TLNK
1835         001472       MINS EQU   *
1836 01472   1 02 00001        LDA   1,1
1837 01473   140407            TCA
1838 01474   0 01 00125       JMP   PUT
1839                      * **** DMINUS ****
1840                      * ( D --- D )
1841                      *    HEAD  FNUL,DMINUS,DMNS
1842         001475       TLNK SET   *
1843 01475   003304            VFD   1,0,1,FNUL,6,6,8,'304
1844                      *                           =<FNUL,6>,'D'
1845 01476   146711            BCI   2,MINU
     01477   147325
1846 01500   051400            VFD   8,'123            ='S'
1847 01501   0 001466          DAC   LINK
1848         001475       LINK SET   TLNK
1849         001502       DMNS EQU   *
1850 01502   1 02 00002        LDA   2,1
1851 01503   140407            TCA
1852 01504   1 04 00002        STA   2,1
1853 01505   140200            RCB
1854 01506   101040            SNZ                    ZERO?
1855 01507   140600            SCB                    YES - SET CARRY
1856 01510   1 02 00001        LDA   1,1              MS WORD
1857 01511   140401            CMA
1858 01512   141216            ACA                    ADD ANY CARRY IN
1859 01513   0 01 00125       JMP   PUT
1860                          EJCT
```

```
1861                    * **** OVER ****
1862                    * ( N1 N2 --- N1 N2 N1 )
1863                    *    HEAD  FNUL,OVER,OVER
1864          001514    TLNK SET   *
1865 01514    002317         VFD   1,0,1,FNUL,6,4,8,'317
1866                    *                              =<FNUL,4>,'O'
1867 01515    153305         BCI   1,VE
1868 01516    051000         VFD   8,'122             ='R'
1869 01517    0 001475       DAC   LINK
1870          001514    LINK SET   TLNK
1871          001520    OVER EQU   *
1872 01520    1 02 00002     LDA   2,1
1873 01521    0 01 00116     JMP   PUSH
1874                    * **** DROP ****
1875                    * ( N --- )
1876                    *    HEAD  FNUL,DROP,DROP
1877          001522    TLNK SET   *
1878 01522    002304         VFD   1,0,1,FNUL,6,4,8,'304
1879                    *                              =<FNUL,4>,'D'
1880 01523    151317         BCI   1,RO
1881 01524    050000         VFD   8,'120             ='P'
1882 01525    0 001514       DAC   LINK
1883          001522    LINK SET   TLNK
1884          001526    DROP EQU   *
1885 01526    0 01 00113     JMP   POP
1886                    * **** SWAP ****
1887                    * ( N1 N2 --- N2 N1 )
1888                    *    HEAD  FNUL,SWAP,SWAP
1889          001527    TLNK SET   *
1890 01527    002323         VFD   1,0,1,FNUL,6,4,8,'323
1891                    *                              =<FNUL,4>,'S'
1892 01530    153701         BCI   1,WA
1893 01531    050000         VFD   8,'120             ='P'
1894 01532    0 001522       DAC   LINK
1895          001527    LINK SET   TLNK
1896          001533    SWAP EQU   *
1897 01533    1 02 00001     LDA   1,1
1898 01534    1 13 00002     IMA   2,1
1899 01535    0 01 00125     JMP   PUT
1900                         EJCT
```

```
1901                    * **** DUP ****
1902                    * ( N1 --- N1 N1 )
1903                    *    HEAD  FNUL,DUP,DUP
1904          001536    TLNK SET   *
1905 01536    001704         VFD   1,0,1,FNUL,6,3,8,'304
1906                    *                              =<FNUL,3>,'D'
1907 01537    152520         VFD   8,'325,8,'120       ='UP'
1908 01540    0 001527       DAC   LINK
1909          001536    LINK SET   TLNK
1910          001541    DUP  EQU   *
1911 01541    1 02 00001     LDA   1,1
1912 01542    0 01 00116     JMP   PUSH
1913                    *
1914                    * **** +! ****
1915                    * ( N A --- )
1916                    *    HEAD  FNUL,+!,PSTR
1917          001543    TLNK SET   *
1918 01543    001253         VFD   1,0,1,FNUL,6,2,8,'253
1919                    *                              =<FNUL,2>,'+'
1920 01544    020400         VFD   8,'041              ='!'
1921 01545    0 001536       DAC   LINK
1922          001543    LINK SET   TLNK
1923          001546    PSTR EQU   *
1924                         IFZ   XTND
1925 01546    1 02 00002     LDA   2,1
1926 01547   -1 06 00001     ADD*  1,1
1927 01550   -1 04 00001     STA*  1,1
1928                         ELSE
1929                         LDA   1,1
1930                         STA   T1
1931                         LDA   2,1
1932                         ADD*  T1
1933                         STA*  T1
1934                         ENDC
1935 01551    0 01 00112     JMP   POP2
1936                    *
1937                    * **** TOGGLE ****
1938                    * ( A N --- ) EXCLUSIVE OR INTO MEMORY WORD
1939                    *    HEAD  FNUL,TOGGLE,TOGL
1940          001552    TLNK SET   *
1941 01552    003324         VFD   1,0,1,FNUL,6,6,8,'324
1942                    *                              =<FNUL,6>,'T'
1943 01553    147707         BCI   2,OGGL
     01554    143714
1944 01555    042400         VFD   8,'105              ='E'
1945 01556    0 001543       DAC   LINK
1946          001552    LINK SET   TLNK
1947          001557    TOGL EQU   *
1948                         IFZ   XTND
1949 01557    1 02 00001     LDA   1,1                 GET PATTERN
```

```
1950 01560  -1 05 00002      ERA*  2,1
1951 01561  -1 04 00002      STA*  2,1
1952                         ELSE
1953                         LDA   2,1
1954                         STA   T1
1955                         LDA   1,1              GET PATTERN
1956                         ERA*  T1
1957                         STA*  T1
1958                         ENDC
1959 01562   0 01 00112      JMP   POP2
1960                         EJCT
```

```
1961                    * **** @ ****
1962                    * ( A --- N)
1963                    *    HEAD  FNUL,@,AT
1964          001563    TLNK SET   *
1965 01563    000500         VFD   1,0,1,FNUL,6,1,8,'100
1966                    *                              =<FNUL,1>,'@'
1967 01564    0 001552       DAC   LINK
1968          001563    LINK SET   TLNK
1969          001565    AT   EQU   *
1970                         IFZ   XTND
1971 01565   -1 02 00001     LDA*  1,1
1972                         ELSE
1973                         LDA   1,1
1974                         STA   T1
1975                         LDA*  T1
1976                         ENDC
1977 01566    0 01 00125     JMP   PUT
1978                    *
1979                    * **** C@ ****
1980                    * ( CADDR --- N)
1981                    *    HEAD  FNUL,C@,CAT
1982          001567    TLNK SET   *
1983 01567    001303         VFD   1,0,1,FNUL,6,2,8,'303
1984                    *                              =<FNUL,2>,'C'
1985 01570    040000         VFD   8,'100             ='@'
1986 01571    0 001563       DAC   LINK
1987          001567    LINK SET   TLNK
1988          001572    CAT  EQU   *
1989 01572    1 02 00001     LDA   1,1
1990 01573    0 10 00166     JST   CHGT
1991 01574    0 01 00125     JMP   PUT
1992                    *
1993                    * **** ! ****
1994                    * ( N A --- )
1995                    *    HEAD  FNUL,!,STOR
1996          001575    TLNK SET   *
1997 01575    000441         VFD   1,0,1,FNUL,6,1,8,'041
1998                    *                              =<FNUL,1>,'!'
1999 01576    0 001567       DAC   LINK
2000          001575    LINK SET   TLNK
2001          001577    STOR EQU   *
2002                         IFZ   XTND
2003 01577    1 02 00002     LDA   2,1
2004 01600   -1 04 00001     STA*  1,1
2005                         ELSE
2006                         LDA   1,1
2007                         STA   T1
2008                         LDA   2,1
2009                         STA*  T1
2010                         ENDC
```

```
2011 01601   0 01 00112      JMP   POP2
2012                        EJCT
```

```
2013                    * **** C! ****
2014                    * ( N CADDR --- )
2015                    *    HEAD  FNUL,C!,CSTR
2016          001602    TLNK SET   *
2017 01602    001303         VFD   1,0,1,FNUL,6,2,8,'303
2018                    *                              =<FNUL,2>,'C'
2019 01603    020400         VFD   8,'041             ='!'
2020 01604    0 001575       DAC   LINK
2021          001602    LINK SET   TLNK
2022          001605    CSTR EQU   *
2023 01605    1 02 00002     LDA   2,1
2024 01606    000201         IAB
2025 01607    1 02 00001     LDA   1,1
2026 01610    0 10 00176     JST   CHPT
2027 01611    0 01 00112     JMP   POP2
2028                    * **** BYTE ****
2029                    * ( A --- CADDR) GET BYTE ADDRESS FROM WORD ADDRESS
2030                    *    HEAD  FNUL,BYTE,BYTE
2031          001612    TLNK SET   *
2032 01612    002302         VFD   1,0,1,FNUL,6,4,8,'302
2033                    *                              =<FNUL,4>,'B'
2034 01613    154724         BCI   1,YT
2035 01614    042400         VFD   8,'105             ='E'
2036 01615    0 001602       DAC   LINK
2037          001612    LINK SET   TLNK
2038          001616    BYTE EQU   *
2039 01616    1 02 00001     LDA   1,1
2040 01617    0414 77        LGL   1
2041 01620    0 01 00125     JMP   PUT
2042                    * **** CELL ****
2043                    * ( CADDR --- A) GET WORD ADDRESS FROM BYTE ADDRESS
2044                    *    HEAD  FNUL,CELL,CELL
2045          001621    TLNK SET   *
2046 01621    002303         VFD   1,0,1,FNUL,6,4,8,'303
2047                    *                              =<FNUL,4>,'C'
2048 01622    142714         BCI   1,EL
2049 01623    046000         VFD   8,'114             ='L'
2050 01624    0 001612       DAC   LINK
2051          001621    LINK SET   TLNK
2052          001625    CELL EQU   *
2053 01625    1 02 00001     LDA   1,1
2054 01626    0404 77        LGR   1
2055 01627    0 01 00125     JMP   PUT
2056                         EJCT
```

```
2057                    ****************************************************************
2058                    *
2059                    *               PRE-COMPILED FORTH SECTION
2060                    *
2061                    ****************************************************************
2062                    *
2063                    *
2064                    *
2065                    *  NOTE - A FEW OF THE FOLLOWING OPERATIONS HAVE BEEN
2066                    *  CONVERTED TO CODE FOR SPEED.  HOWEVER, THE WORD ORDER
2067                    *  IN THE DICTIONARY HAS NOT BEEN CHANGED.
2068                    *
2069                    * **** : ****
2070                    *    HEAD  FIMD,:,COLN,DOCL
2071         001630     TLNK SET   *
2072 01630   040472          VFD   1,0,1,FIMD,6,1,8,'072
2073                    *                              =<FIMD,1>,':'
2074 01631   0 001621        DAC   LINK
2075         001630     LINK SET   TLNK
2076 01632   0 10 00130 COLN JST   DOCL
2077 01633   0 002506        DAC   QEXC
2078 01634   0 002443        DAC   SCSP
2079 01635   0 002136        DAC   CURR
2080 01636   0 001565        DAC   AT
2081 01637   0 002127        DAC   CONT
2082 01640   0 001577        DAC   STOR
2083 01641   0 003746        DAC   CRAT
2084 01642   0 002612        DAC   RBRC
2085 01643   0 002675        DAC   PSCD
2086 01644   0 10 00130      JST   DOCL             PICKED UP BY PSCD
2087                    *
2088                    * **** ; ****
2089                    *    HEAD  FIMD,;,SEMI,DOCL
2090         001645     TLNK SET   *
2091 01645   040473          VFD   1,0,1,FIMD,6,1,8,'073
2092                    *                              =<FIMD,1>,';'
2093 01646   0 001630        DAC   LINK
2094         001645     LINK SET   TLNK
2095 01647   0 10 00130 SEMI JST   DOCL
2096 01650   0 002534        DAC   QCSP
2097 01651   0 002570        DAC   COMP
2098 01652   0 001361        DAC   SMIS
2099 01653   0 002625        DAC   SMDG
2100 01654   0 002603        DAC   LBRC
2101 01655   0 001361        DAC   SMIS
2102                    *
2103                    * **** CONSTANT ****
2104                    *    HEAD  FNUL,CONSTANT,CON,DOCL
2105         001656     TLNK SET   *
2106 01656   004303          VFD   1,0,1,FNUL,6,8,8,'303
```

```
2107                    *                               =<FNUL,8>,'C'
2108 01657   147716           BCI   3,ONSTAN
     01660   151724
     01661   140716
2109 01662   052000           VFD   8,'124              ='T'
2110 01663   0 001645         DAC   LINK
2111         001656     LINK SET   TLNK
2112 01664   0 10 00130 CON  JST   DOCL
2113 01665   0 003746         DAC   CRAT
2114 01666   0 002625         DAC   SMDG
2115 01667   0 002256         DAC   COMA
2116 01670   0 002675         DAC   PSCD
2117 01671   0 10 00140       JST   DOCN               PICKED UP BY PSCD
2118                    *
2119                    * **** VARIABLE ****
2120                    * ( N --- )
2121                    *    HEAD  FNUL,VARIABLE,VAR,DOCL
2122         001672     TLNK SET   *
2123 01672   004326           VFD   1,0,1,FNUL,6,8,8,'326
2124                    *                               =<FNUL,8>,'V'
2125 01673   140722           BCI   3,ARIABL
     01674   144701
     01675   141314
2126 01676   042400           VFD   8,'105              ='E'
2127 01677   0 001656         DAC   LINK
2128         001672     LINK SET   TLNK
2129 01700   0 10 00130 VAR  JST   DOCL
2130 01701   0 001664         DAC   CON
2131 01702   0 002675         DAC   PSCD
2132 01703   0 10 00143       JST   DOVR               PICKED UP BY PSCD
2133                    *
2134                    * **** USER ****
2135                    *    HEAD  FNUL,USER,USER,DOCL
2136         001704     TLNK SET   *
2137 01704   002325           VFD   1,0,1,FNUL,6,4,8,'325
2138                    *                               =<FNUL,4>,'U'
2139 01705   151705           BCI   1,SE
2140 01706   051000           VFD   8,'122              ='R'
2141 01707   0 001672         DAC   LINK
2142         001704     LINK SET   TLNK
2143 01710   0 10 00130 USER JST   DOCL
2144 01711   0 001664         DAC   CON
2145 01712   0 002675         DAC   PSCD
2146 01713   0 10 00146       JST   DOUS               PICKED UP BY PSCD
2147                         EJCT
```

```
2148                    *
2149                    *  CONSTANTS
2150                    *
2151                    * **** 0
2152                    *    HEAD  FNUL,0,ZERO,DOCN
2153           001714   TLNK SET   *
2154 01714     000460         VFD   1,0,1,FNUL,6,1,8,'060
2155                    *                                =<FNUL,1>,'0'
2156 01715     0 001704       DAC   LINK
2157           001714   LINK SET   TLNK
2158 01716     0 10 00140 ZERO JST   DOCN
2159 01717     000000         DEC   0
2160                    *
2161                    * **** 1 ****
2162                    *    HEAD  FNUL,1,ONE,DOCN
2163           001720   TLNK SET   *
2164 01720     000461         VFD   1,0,1,FNUL,6,1,8,'061
2165                    *                                =<FNUL,1>,'1'
2166 01721     0 001714       DAC   LINK
2167           001720   LINK SET   TLNK
2168 01722     0 10 00140 ONE  JST   DOCN
2169 01723     000001         DEC   1
2170                    *
2171                    * **** 2 ****
2172                    *    HEAD  FNUL,2,TWO,DOCN
2173           001724   TLNK SET   *
2174 01724     000462         VFD   1,0,1,FNUL,6,1,8,'062
2175                    *                                =<FNUL,1>,'2'
2176 01725     0 001720       DAC   LINK
2177           001724   LINK SET   TLNK
2178 01726     0 10 00140 TWO  JST   DOCN
2179 01727     000002         DEC   2
2180                    *
2181                    * **** 3 ****
2182                    *    HEAD  FNUL,3,THRE,DOCN
2183           001730   TLNK SET   *
2184 01730     000463         VFD   1,0,1,FNUL,6,1,8,'063
2185                    *                                =<FNUL,1>,'3'
2186 01731     0 001724       DAC   LINK
2187           001730   LINK SET   TLNK
2188 01732     0 10 00140 THRE JST   DOCN
2189 01733     000003         DEC   3
2190                    *
2191                    * **** BL ****
2192                    *    HEAD  FNUL,BL,BL,DOCN
2193           001734   TLNK SET   *
2194 01734     001302         VFD   1,0,1,FNUL,6,2,8,'302
2195                    *                                =<FNUL,2>,'B'
2196 01735     046000         VFD   8,'114                ='L'
2197 01736     0 001730       DAC   LINK
```

```
2198          001734     LINK SET   TLNK
2199 01737    0 10 00140 BL   JST   DOCN
2200 01740    000240           VFD   16,CSPC
2201                     *
2202                     * **** C/L **** # OF CHARACTERS PER LINE
2203                     *    HEAD  FNUL,C/L,CL,DOCN
2204          001741     TLNK SET   *
2205 01741    001703           VFD   1,0,1,FNUL,6,3,8,'303
2206                     *                              =<FNUL,3>,'C'
2207 01742    127514           VFD   8,'257,8,'114     ='/L'
2208 01743    0 001734         DAC   LINK
2209          001741     LINK SET   TLNK
2210 01744    0 10 00140 CL   JST   DOCN
2211 01745    000100           OCT   100
2212                     *
2213                     * 'FIRST' AND 'LIMIT' MOVED TO USER AREA
2214                     *
2215                     * **** B/BUF **** BYTES PER DISK-BLOCK BUFFER.
2216                     *    HEAD  FNUL,B/BUF,BBUF,DOCN
2217          001746     TLNK SET   *
2218 01746    002702           VFD   1,0,1,FNUL,6,5,8,'302
2219                     *                              =<FNUL,5>,'B'
2220 01747    127702           BCI   1,/B
2221 01750    152506           VFD   8,'325,8,'106     ='UF'
2222 01751    0 001741         DAC   LINK
2223          001746     LINK SET   TLNK
2224 01752    0 10 00140 BBUF JST   DOCN
2225 01753    002000           DEC   1024
2226                     *
2227                     * **** B/SCR **** DISK BLOCKS PER FORTH SCREEN.
2228                     *    HEAD  FNUL,B/SCR,BSCR,DOCN
2229          001754     TLNK SET   *
2230 01754    002702           VFD   1,0,1,FNUL,6,5,8,'302
2231                     *                              =<FNUL,5>,'B'
2232 01755    127723           BCI   1,/S
2233 01756    141522           VFD   8,'303,8,'122     ='CR'
2234 01757    0 001746         DAC   LINK
2235          001754     LINK SET   TLNK
2236 01760    0 10 00140 BSCR JST   DOCN
2237 01761    000001           DEC   1
2238                     *
2239                     * **** +ORIGIN **** RETURNS ADDRESS, GIVEN OFFSET FROM ORIGIN.
2240                     *    HEAD  FNUL,+ORIGIN,PORG,DOCL
2241          001762     TLNK SET   *
2242 01762    003653           VFD   1,0,1,FNUL,6,7,8,'253
2243                     *                              =<FNUL,7>,'+'
2244 01763    147722           BCI   2,ORIG
     01764    144707
2245 01765    144516           VFD   8,'311,8,'116     ='IN'
2246 01766    0 001754         DAC   LINK
```

```
2247         001762     LINK SET   TLNK
2248 01767   0 10 00130 PORG JST   DOCL
2249 01770   0 000241         DAC   LIT
2250 01771   0 001000         DAC   ORGN
2251 01772   0 001443         DAC   PLUS
2252 01773   0 001361         DAC   SMIS
2253                    *
2254                    *  USER VARIABLES
2255                    *
2256                    * **** S0 **** STACK ORIGIN.
2257                    *    HEAD  FNUL,S0,SZRO,DOUS
2258         001774     TLNK SET   *
2259 01774   001323           VFD   1,0,1,FNUL,6,2,8,'323
2260                    *                              =<FNUL,2>,'S'
2261 01775   030000           VFD   8,'060           ='0'
2262 01776   0 001762         DAC   LINK
2263         001774     LINK SET   TLNK
2264 01777   0 10 00146 SZRO JST   DOUS
2265 02000   000003           OCT   3
2266                    *
2267                    * **** R0 **** RETURN STACK ORIGIN.
2268                    *    HEAD  FNUL,R0,RZRO,DOUS
2269         002001     TLNK SET   *
2270 02001   001322           VFD   1,0,1,FNUL,6,2,8,'322
2271                    *                              =<FNUL,2>,'R'
2272 02002   030000           VFD   8,'060           ='0'
2273 02003   0 001774         DAC   LINK
2274         002001     LINK SET   TLNK
2275 02004   0 10 00146 RZRO JST   DOUS
2276 02005   000004           OCT   4
2277                    *
2278                    * **** TIB **** TERMINAL INPUT BUFFER.
2279                    *    HEAD  FNUL,TIB,TIB,DOUS
2280         002006     TLNK SET   *
2281 02006   001724           VFD   1,0,1,FNUL,6,3,8,'324
2282                    *                              =<FNUL,3>,'T'
2283 02007   144502           VFD   8,'311,8,'102    ='IB'
2284 02010   0 002001         DAC   LINK
2285         002006     LINK SET   TLNK
2286 02011   0 10 00146 TIB  JST   DOUS
2287 02012   000005           OCT   5
2288                    *
2289                    * **** WIDTH **** MAXIMUM NAME LENGTH (DEFAULT, 31 CHARACTERS).
2290                    *    HEAD  FNUL,WIDTH,WDTH,DOUS
2291         002013     TLNK SET   *
2292 02013   002727           VFD   1,0,1,FNUL,6,5,8,'327
2293                    *                              =<FNUL,5>,'W'
2294 02014   144704           BCI   1,ID
2295 02015   152110           VFD   8,'324,8,'110    ='TH'
2296 02016   0 002006         DAC   LINK
```

```
2297         002013     LINK SET   TLNK
2298 02017   0 10 00146 WDTH JST   DOUS
2299 02020   000006          OCT   6
2300                     *
2301                     * **** WARNING **** WARNING MODE
2302                     *  (DEFAULT, GIVE MESSAGE NUMBER AT ERROR OR WARNING CONDITION,
2303                     *   DON'T GO TO DISK FOR MESSAGE).
2304                     *    HEAD  FNUL,WARNING,WARN,DOUS
2305         002021     TLNK SET   *
2306 02021   003727          VFD   1,0,1,FNUL,6,7,8,'327
2307                     *                              =<FNUL,7>,'W'
2308 02022   140722          BCI   2,ARNI
     02023   147311
2309 02024   147107          VFD   8,'316,8,'107     ='NG'
2310 02025   0 002013        DAC   LINK
2311         002021     LINK SET   TLNK
2312 02026   0 10 00146 WARN JST   DOUS
2313 02027   000007          OCT   7
2314                     *
2315                     * **** FENCE **** PREVENTS 'FORGET' BELOW THIS 'FENCE' SETTING.
2316                     *    HEAD  FNUL,FENCE,FENC,DOUS
2317         002030     TLNK SET   *
2318 02030   002706          VFD   1,0,1,FNUL,6,5,8,'306
2319                     *                              =<FNUL,5>,'F'
2320 02031   142716          BCI   1,EN
2321 02032   141505          VFD   8,'303,8,'105     ='CE'
2322 02033   0 002021        DAC   LINK
2323         002030     LINK SET   TLNK
2324 02034   0 10 00146 FENC JST   DOUS
2325 02035   000010          OCT   10
2326                     *
2327                     * **** DP **** DICTIONARY POINTER TO NEXT AVAILABLE SPACE.
2328                     *    HEAD  FNUL,DP,DP,DOUS
2329         002036     TLNK SET   *
2330 02036   001304          VFD   1,0,1,FNUL,6,2,8,'304
2331                     *                              =<FNUL,2>,'D'
2332 02037   050000          VFD   8,'120            ='P'
2333 02040   0 002030        DAC   LINK
2334         002036     LINK SET   TLNK
2335 02041   0 10 00146 DP   JST   DOUS
2336 02042   000011          OCT   11
2337                     *
2338                     * **** VOC-LINK **** VOCABULARY LINK (MAINLY FOR FUTURE USE).
2339                     *    HEAD  FNUL,VOC-LINK,VOCL,DOUS
2340         002043     TLNK SET   *
2341 02043   004326          VFD   1,0,1,FNUL,6,8,8,'326
2342                     *                              =<FNUL,8>,'V'
2343 02044   147703          BCI   3,OC-LIN
     02045   126714
     02046   144716
```

```
2344 02047   045400          VFD   8,'113           ='K'
2345 02050   0 002036        DAC   LINK
2346         002043     LINK SET   TLNK
2347 02051   0 10 00146 VOCL JST   DOUS
2348 02052   000012          OCT   12
2349                    *
2350                    * **** FIRST **** ADDRESS OF BEGINNING OF DISK BUFFER.
2351                    *    HEAD  FNUL,FIRST,FRST,DOUS
2352         002053     TLNK SET   *
2353 02053   002706          VFD   1,0,1,FNUL,6,5,8,'306
2354                    *                             =<FNUL,5>,'F'
2355 02054   144722          BCI   1,IR
2356 02055   151524          VFD   8,'323,8,'124     ='ST'
2357 02056   0 002043        DAC   LINK
2358         002053     LINK SET   TLNK
2359 02057   0 10 00146 FRST JST   DOUS
2360 02060   000013          OCT   13
2361                    *
2362                    * **** LIMIT **** ADDRESS JUST BEYOND END OF DISK BUFFERS.
2363                    *    HEAD  FNUL,LIMIT,LIMT,DOUS
2364         002061     TLNK SET   *
2365 02061   002714          VFD   1,0,1,FNUL,6,5,8,'314
2366                    *                             =<FNUL,5>,'L'
2367 02062   144715          BCI   1,IM
2368 02063   144524          VFD   8,'311,8,'124     ='IT'
2369 02064   0 002053        DAC   LINK
2370         002061     LINK SET   TLNK
2371 02065   0 10 00146 LIMT JST   DOUS
2372 02066   000014          OCT   14
2373                    *
2374                    * POSITIONS '15 AND '16 ARE AVAILABLE FOR EXPANSION.
2375                    * THEY ARE INITIALIZED FROM BOOT-UP TABLE, AT COLD START.
2376                    *
2377                    * **** BLK **** CURRENT DISK BLOCK BEING LOADED (0=TERMINAL)
2378                    *    HEAD  FNUL,BLK,BLK,DOUS
2379         002067     TLNK SET   *
2380 02067   001702          VFD   1,0,1,FNUL,6,3,8,'302
2381                    *                             =<FNUL,3>,'B'
2382 02070   146113          VFD   8,'314,8,'113     ='LK'
2383 02071   0 002061        DAC   LINK
2384         002067     LINK SET   TLNK
2385 02072   0 10 00146 BLK  JST   DOUS
2386 02073   000017          OCT   17
2387                    *
2388                    * **** IN **** OFFSET IN TERMINAL INPUT BUFFER.
2389                    *    HEAD  FNUL,IN,IN,DOUS
2390         002074     TLNK SET   *
2391 02074   001311          VFD   1,0,1,FNUL,6,2,8,'311
2392                    *                             =<FNUL,2>,'I'
2393 02075   047000          VFD   8,'116           ='N'
```

```
2394 02076   0 002067        DAC   LINK
2395         002074     LINK SET   TLNK
2396 02077   0 10 00146 IN   JST   DOUS
2397 02100   000020           OCT   20
2398                    *
2399                    * **** OUT **** OFFSET IN OUTPUT LINE.
2400                    *    HEAD  FNUL,OUT,OUT,DOUS
2401         002101     TLNK SET   *
2402 02101   001717           VFD   1,0,1,FNUL,6,3,8,'317
2403                    *                              =<FNUL,3>,'O'
2404 02102   152524           VFD   8,'325,8,'124      ='UT'
2405 02103   0 002074        DAC   LINK
2406         002101     LINK SET   TLNK
2407 02104   0 10 00146 OUT  JST   DOUS
2408 02105   000021           OCT   21
2409                    *
2410                    * **** SCR **** CURRENT FORTH DISK SCREEN.
2411                    *    HEAD  FNUL,SCR,SCR,DOUS
2412         002106     TLNK SET   *
2413 02106   001723           VFD   1,0,1,FNUL,6,3,8,'323
2414                    *                              =<FNUL,3>,'S'
2415 02107   141522           VFD   8,'303,8,'122      ='CR'
2416 02110   0 002101        DAC   LINK
2417         002106     LINK SET   TLNK
2418 02111   0 10 00146 SCR  JST   DOUS
2419 02112   000022           OCT   22
2420                    *
2421                    * **** OFFSET ****
2422                    *    HEAD  FNUL,OFFSET,OFST,DOUS
2423         002113     TLNK SET   *
2424 02113   003317           VFD   1,0,1,FNUL,6,6,8,'317
2425                    *                              =<FNUL,6>,'O'
2426 02114   143306           BCI   2,FFSE
     02115   151705
2427 02116   052000           VFD   8,'124             ='T'
2428 02117   0 002106        DAC   LINK
2429         002113     LINK SET   TLNK
2430 02120   0 10 00146 OFST JST   DOUS
2431 02121   000023           OCT   23
2432                    *
2433                    * **** CONTEXT ****
2434                    *    HEAD  FNUL,CONTEXT,CONT,DOUS
2435         002122     TLNK SET   *
2436 02122   003703           VFD   1,0,1,FNUL,6,7,8,'303
2437                    *                              =<FNUL,7>,'C'
2438 02123   147716           BCI   2,ONTE
     02124   152305
2439 02125   154124           VFD   8,'330,8,'124      ='XT'
2440 02126   0 002113        DAC   LINK
2441         002122     LINK SET   TLNK
```

```
2442 02127   0 10 00146 CONT JST   DOUS
2443 02130   000024          OCT   24
2444                      *
2445                      * **** CURRENT ****
2446                      *    HEAD  FNUL,CURRENT,CURR,DOUS
2447         002131     TLNK SET   *
2448 02131   003703          VFD   1,0,1,FNUL,6,7,8,'303
2449                      *                              =<FNUL,7>,'C'
2450 02132   152722          BCI   2,URRE
     02133   151305
2451 02134   147124          VFD   8,'316,8,'124     ='NT'
2452 02135   0 002122        DAC   LINK
2453         002131     LINK SET   TLNK
2454 02136   0 10 00146 CURR JST   DOUS
2455 02137   000025          OCT   25
2456                      *
2457                      * **** STATE ****
2458                      *    HEAD  FNUL,STATE,STAT,DOUS
2459         002140     TLNK SET   *
2460 02140   002723          VFD   1,0,1,FNUL,6,5,8,'323
2461                      *                              =<FNUL,5>,'S'
2462 02141   152301          BCI   1,TA
2463 02142   152105          VFD   8,'324,8,'105     ='TE'
2464 02143   0 002131        DAC   LINK
2465         002140     LINK SET   TLNK
2466 02144   0 10 00146 STAT JST   DOUS
2467 02145   000026          OCT   26
2468                      *
2469                      * **** BASE ****
2470                      *    HEAD  FNUL,BASE,BASE,DOUS
2471         002146     TLNK SET   *
2472 02146   002302          VFD   1,0,1,FNUL,6,4,8,'302
2473                      *                              =<FNUL,4>,'B'
2474 02147   140723          BCI   1,AS
2475 02150   042400          VFD   8,'105            ='E'
2476 02151   0 002140        DAC   LINK
2477         002146     LINK SET   TLNK
2478 02152   0 10 00146 BASE JST   DOUS
2479 02153   000027          OCT   27
2480                      *
2481                      * **** DPL **** OFFSET OF DECIMAL POINT AFTER DOUBLE-INTEGER INPUT.
2482                      *    HEAD  FNUL,DPL,DPL,DOUS
2483         002154     TLNK SET   *
2484 02154   001704          VFD   1,0,1,FNUL,6,3,8,'304
2485                      *                              =<FNUL,3>,'D'
2486 02155   150114          VFD   8,'320,8,'114     ='PL'
2487 02156   0 002146        DAC   LINK
2488         002154     LINK SET   TLNK
2489 02157   0 10 00146 DPL  JST   DOUS
2490 02160   000030          OCT   30
```

```
2491                    *
2492                    * **** FLD ****
2493                    *    HEAD  FNUL,FLD,FLD,DOUS
2494          002161    TLNK SET   *
2495 02161   001706           VFD   1,0,1,FNUL,6,3,8,'306
2496                    *                              =<FNUL,3>,'F'
2497 02162   146104           VFD   8,'314,8,'104      ='LD'
2498 02163   0 002154         DAC   LINK
2499          002161    LINK SET   TLNK
2500 02164   0 10 00146 FLD  JST   DOUS
2501 02165   000031           OCT   31
2502                    *
2503                    * **** CSP **** USED BY COMPILER TO HOLD CURRENT STACK POSITION,
2504                    *                FOR ERROR CHECKING.
2505                    *    HEAD  FNUL,CSP,CSP,DOUS
2506          002166    TLNK SET   *
2507 02166   001703           VFD   1,0,1,FNUL,6,3,8,'303
2508                    *                              =<FNUL,3>,'C'
2509 02167   151520           VFD   8,'323,8,'120      ='SP'
2510 02170   0 002161         DAC   LINK
2511          002166    LINK SET   TLNK
2512 02171   0 10 00146 CSP  JST   DOUS
2513 02172   000032           OCT   32
2514                    *
2515                    * **** R# **** CURSOR POSITION (FOR SOME EDITORS).
2516                    *    HEAD  FNUL,R#,RNUM,DOUS
2517          002173    TLNK SET   *
2518 02173   001322           VFD   1,0,1,FNUL,6,2,8,'322
2519                    *                              =<FNUL,2>,'R'
2520 02174   021400           VFD   8,'043             =' '
2521 02175   0 002166         DAC   LINK
2522          002173    LINK SET   TLNK
2523 02176   0 10 00146 RNUM JST   DOUS
2524 02177   000033           OCT   33
2525                    *
2526                    * **** HLD **** POINTS TO LAST CHARACTER HELD IN 'PAD'
2527                    *    HEAD  FNUL,HLD,HLD,DOUS
2528          002200    TLNK SET   *
2529 02200   001710           VFD   1,0,1,FNUL,6,3,8,'310
2530                    *                              =<FNUL,3>,'H'
2531 02201   146104           VFD   8,'314,8,'104      ='LD'
2532 02202   0 002173         DAC   LINK
2533          002200    LINK SET   TLNK
2534 02203   0 10 00146 HLD  JST   DOUS
2535 02204   000034           OCT   34
2536                    *
2537                    * **** USE ****
2538                    *    HEAD  FNUL,USE,USE,DOUS
2539          002205    TLNK SET   *
2540 02205   001725           VFD   1,0,1,FNUL,6,3,8,'325
```

```
2541                    *                               =<FNUL,3>,'U'
2542 02206   151505           VFD   8,'323,8,'105       ='SE'
2543 02207   0 002200         DAC   LINK
2544         002205     LINK SET   TLNK
2545 02210   0 10 00146 USE  JST   DOUS
2546 02211   000035           OCT   35
2547                    *
2548                    * **** PREV ****
2549                    *    HEAD  FNUL,PREV,PREV,DOUS
2550         002212     TLNK SET   *
2551 02212   002320           VFD   1,0,1,FNUL,6,4,8,'320
2552                    *                               =<FNUL,4>,'P'
2553 02213   151305           BCI   1,RE
2554 02214   053000           VFD   8,'126              ='V'
2555 02215   0 002205         DAC   LINK
2556         002212     LINK SET   TLNK
2557 02216   0 10 00146 PREV JST   DOUS
2558 02217   000036           OCT   36
2559                    *
2560                    *END OF USER AREA
2561                    *
2562                         EJCT
```

```
2563                    * **** 1+ ****
2564                    *    HEAD  FNUL,1+,ONEP
2565          002220    TLNK SET   *
2566 02220    001261         VFD   1,0,1,FNUL,6,2,8,'261
2567                    *                              =<FNUL,2>,'1'
2568 02221    025400         VFD   8,'053             ='+'
2569 02222    0 002212       DAC   LINK
2570          002220    LINK SET   TLNK
2571          002223    ONEP EQU   *
2572 02223    1 02 00001     LDA   1,1
2573 02224    141206         AOA
2574 02225    0 01 00125     JMP   PUT
2575                    *
2576                    * **** 2+ ****
2577                    *    HEAD  FNUL,2+,TWOP
2578          002226    TLNK SET   *
2579 02226    001262         VFD   1,0,1,FNUL,6,2,8,'262
2580                    *                              =<FNUL,2>,'2'
2581 02227    025400         VFD   8,'053             ='+'
2582 02230    0 002220       DAC   LINK
2583          002226    LINK SET   TLNK
2584          002231    TWOP EQU   *
2585 02231    1 02 00001     LDA   1,1
2586 02232    0 06 00720     ADD   =2
2587 02233    0 01 00125     JMP   PUT
2588                    *
2589                    * **** HERE ****
2590                    *    HEAD  FNUL,HERE,HERE,DOCL
2591          002234    TLNK SET   *
2592 02234    002310         VFD   1,0,1,FNUL,6,4,8,'310
2593                    *                              =<FNUL,4>,'H'
2594 02235    142722         BCI   1,ER
2595 02236    042400         VFD   8,'105             ='E'
2596 02237    0 002226       DAC   LINK
2597          002234    LINK SET   TLNK
2598 02240    0 10 00130 HERE JST   DOCL
2599 02241    0 002041       DAC   DP
2600 02242    0 001565       DAC   AT
2601 02243    0 001361       DAC   SMIS
2602                    *
2603                    * **** ALLOT ****
2604                    *    HEAD  FNUL,ALLOT,ALOT,DOCL
2605          002244    TLNK SET   *
2606 02244    002701         VFD   1,0,1,FNUL,6,5,8,'301
2607                    *                              =<FNUL,5>,'A'
2608 02245    146314         BCI   1,LL
2609 02246    147524         VFD   8,'317,8,'124      ='OT'
2610 02247    0 002234       DAC   LINK
2611          002244    LINK SET   TLNK
2612 02250    0 10 00130 ALOT JST   DOCL
```

```
2613 02251   0 002041        DAC   DP
2614 02252   0 001546        DAC   PSTR
2615 02253   0 001361        DAC   SMIS
2616                     *
2617                     * **** , ****
2618                     * ( N  --- ) L0
2619                     * STORE N INTO THE NEXT AVAILABLE DICTIONARY MEMORY CELL, ADVANCING
2620                     * THE DICTIONARY POINTER.
2621                     *    HEAD  FNUL,$,,COMA,DOCL
2622         002254      TLNK SET   *
2623 02254   000454          VFD   1,0,1,FNUL,6,1,8,'054
2624                     *                              =<FNUL,1>,','
2625 02255   0 002244        DAC   LINK
2626         002254      LINK SET   TLNK
2627 02256   0 10 00130 COMA JST   DOCL
2628 02257   0 002240        DAC   HERE
2629 02260   0 001577        DAC   STOR
2630 02261   0 001722        DAC   ONE
2631 02262   0 002250        DAC   ALOT
2632 02263   0 001361        DAC   SMIS
2633                     *
2634                     *  THIS SYSTEM DOES NOT USE 'C,'
2635                     *
2636                     * **** - ****
2637                     *    HEAD  FNUL,-,SUB
2638         002264      TLNK SET   *
2639 02264   000455          VFD   1,0,1,FNUL,6,1,8,'055
2640                     *                              =<FNUL,1>,'-'
2641 02265   0 002254        DAC   LINK
2642         002264      LINK SET   TLNK
2643         002266      SUB  EQU   *
2644 02266   1 02 00002      LDA   2,1
2645 02267   1 07 00001      SUB   1,1
2646 02270   0 01 00124      JMP   BINA
2647                     *
2648                     * **** = ****
2649                     *    HEAD  FNUL,=,EQAL
2650         002271      TLNK SET   *
2651 02271   000475          VFD   1,0,1,FNUL,6,1,8,'075
2652                     *                              =<FNUL,1>,'='
2653 02272   0 002264        DAC   LINK
2654         002271      LINK SET   TLNK
2655         002273      EQAL EQU   *
2656 02273   1 02 00002      LDA   2,1
2657 02274   1 05 00001      ERA   1,1
2658 02275   100040          SZE                    EQUAL?
2659 02276   0 02 00715      LDA   =-1              NO: -1=>0
2660 02277   141206          AOA                    YES: 0=>1
2661 02300   0 01 00124      JMP   BINA
2662                     *
```

```
2663                    * **** < ****
2664                    *    HEAD  FNUL,<,LESS
2665          002301    TLNK SET   *
2666 02301    000474         VFD   1,0,1,FNUL,6,1,8,'074
2667                    *                              =<FNUL,1>,'<'
2668 02302    0 002271       DAC   LINK
2669          002301    LINK SET   TLNK
2670          002303    LESS EQU   *
2671 02303    1 02 00002     LDA   2,1
2672 02304    1 11 00001     CAS   1,1
2673 02305    0 01 02311     JMP   LES2
2674 02306    0 01 02311     JMP   LES2
2675 02307    0 02 00736 LES1 LDA   =1
2676 02310    0 01 00124     JMP   BINA
2677 02311    140040    LES2 CRA
2678 02312    0 01 00124     JMP   BINA
2679                    *
2680                    * **** > ****
2681                    *    HEAD  FNUL,>,GRTR
2682          002313    TLNK SET   *
2683 02313    000476         VFD   1,0,1,FNUL,6,1,8,'076
2684                    *                              =<FNUL,1>,'>'
2685 02314    0 002301       DAC   LINK
2686          002313    LINK SET   TLNK
2687          002315    GRTR EQU   *
2688 02315    1 02 00002     LDA   2,1
2689 02316    1 11 00001     CAS   1,1
2690 02317    0 01 02307     JMP   LES1
2691 02320    0 01 02311     JMP   LES2
2692 02321    0 01 02311     JMP   LES2
2693                    *
2694                    * **** ROT **** ( N1 N2 N3 --- N2 N3 N1 )
2695                    *    HEAD  FNUL,ROT,ROT
2696          002322    TLNK SET   *
2697 02322    001722         VFD   1,0,1,FNUL,6,3,8,'322
2698                    *                              =<FNUL,3>,'R'
2699 02323    147524         VFD   8,'317,8,'124     ='OT'
2700 02324    0 002313       DAC   LINK
2701          002322    LINK SET   TLNK
2702          002325    ROT  EQU   *
2703 02325    1 02 00001     LDA   1,1               N3
2704 02326    1 13 00002     IMA   2,1               STORE N3, GET N2
2705 02327    1 13 00003     IMA   3,1               STORE N2, GET N1
2706 02330    0 01 00125     JMP   PUT
2707                    *
2708                    * **** SPACE **** TYPE ONE SPACE
2709                    *    HEAD  FNUL,SPACE,SPCE,DOCL
2710          002331    TLNK SET   *
2711 02331    002723         VFD   1,0,1,FNUL,6,5,8,'323
2712                    *                              =<FNUL,5>,'S'
```

```
2713 02332   150301          BCI   1,PA
2714 02333   141505          VFD   8,'303,8,'105     ='CE'
2715 02334   0 002322        DAC   LINK
2716         002331     LINK SET   TLNK
2717 02335   0 10 00130 SPCE JST   DOCL
2718 02336   0 001737        DAC   BL
2719 02337   0 000610        DAC   EMIT
2720 02340   0 001361        DAC   SMIS
2721                    *
2722                    * **** -DUP **** ( N--- N (N) ) DUPLICATE ONLY IF NONZERO
2723                    *    HEAD  FNUL,-DUP,DDUP
2724         002341     TLNK SET   *
2725 02341   002255          VFD   1,0,1,FNUL,6,4,8,'255
2726                    *                             =<FNUL,4>,'-'
2727 02342   142325          BCI   1,DU
2728 02343   050000          VFD   8,'120            ='P'
2729 02344   0 002331        DAC   LINK
2730         002341     LINK SET   TLNK
2731         002345     DDUP EQU   *
2732 02345   1 02 00001      LDA   1,1
2733 02346   100040          SZE
2734 02347   0 01 00116      JMP   PUSH
2735                    *
2736                    *    NEXT
2737 02350   0 12 00100      IRS   IP
2738 02351  -0 01 00100      JMP*  IP
2739                    *
2740                    * THIS SYSTEM DOES NOT NEED TRAVERSE, NFA AND PFA ARE OK
2741                    *
2742                    * **** LATEST ****
2743                    *    HEAD  FNUL,LATEST,LTST,DOCL
2744         002352     TLNK SET   *
2745 02352   003314          VFD   1,0,1,FNUL,6,6,8,'314
2746                    *                             =<FNUL,6>,'L'
2747 02353   140724          BCI   2,ATES
     02354   142723
2748 02355   052000          VFD   8,'124            ='T'
2749 02356   0 002341        DAC   LINK
2750         002352     LINK SET   TLNK
2751 02357   0 10 00130 LTST JST   DOCL
2752 02360   0 002136        DAC   CURR
2753 02361   0 001565        DAC   AT
2754 02362   0 001565        DAC   AT
2755 02363   0 001361        DAC   SMIS
2756                    *
2757                    *  THE NEXT 4 OPERATORS CAN DEPEND ON COMPUTER WORD SIZE.
2758                    *  THEY CONVERT ADDRESSES WITHIN THE NAME FIELDS OF FORTH
2759                    *  DICTIONARY ENTRIES.
2760                    *
2761                    * **** LFA **** (PFA --- LFA) GET LINK FIELD ADDRESS
```

```
2762                    *    HEAD  FNUL,LFA,LFA,DOCL
2763          002364    TLNK SET   *
2764 02364    001714         VFD   1,0,1,FNUL,6,3,8,'314
2765                    *                              =<FNUL,3>,'L'
2766 02365    143101         VFD   8,'306,8,'101     ='FA'
2767 02366    0 002352       DAC   LINK
2768          002364    LINK SET   TLNK
2769 02367    0 10 00130 LFA  JST   DOCL
2770 02370    0 001726       DAC   TWO
2771 02371    0 002266       DAC   SUB
2772 02372    0 001361       DAC   SMIS
2773                    *
2774                    * **** CFA **** (PFA --- CFA) GET CODE FIELD ADDRESS
2775                    *    HEAD  FNUL,CFA,CFA,DOCL
2776          002373    TLNK SET   *
2777 02373    001703         VFD   1,0,1,FNUL,6,3,8,'303
2778                    *                              =<FNUL,3>,'C'
2779 02374    143101         VFD   8,'306,8,'101     ='FA'
2780 02375    0 002364       DAC   LINK
2781          002373    LINK SET   TLNK
2782 02376    0 10 00130 CFA  JST   DOCL
2783 02377    0 001722       DAC   ONE
2784 02400    0 002266       DAC   SUB
2785 02401    0 001361       DAC   SMIS
2786                    *
2787                    * **** NFA **** (PFA --- NFA) GET NAME FIELD ADDRESS
2788                    *    HEAD  FNUL,NFA,NFA,DOCL
2789          002402    TLNK SET   *
2790 02402    001716         VFD   1,0,1,FNUL,6,3,8,'316
2791                    *                              =<FNUL,3>,'N'
2792 02403    143101         VFD   8,'306,8,'101     ='FA'
2793 02404    0 002373       DAC   LINK
2794          002402    LINK SET   TLNK
2795 02405    0 10 00130 NFA  JST   DOCL
2796 02406    0 002367       DAC   LFA
2797 02407    0 001722   NFA1 DAC   ONE
2798 02410    0 002266       DAC   SUB               DECREMENT WORD POINTER
2799 02411    0 001541       DAC   DUP
2800 02412    0 001565       DAC   AT                GET WORD
2801 02413    0 001716       DAC   ZERO
2802 02414    0 002315       DAC   GRTR              LOOKING FOR POSITIVE WORD
2803 02415    0 000277       DAC   ZBRA
2804 02416    0 002407       DAC   NFA1
2805 02417    0 001361       DAC   SMIS
2806                    *
2807                    * **** PFA **** (NFA --- PFA) GET PARAMETER FIELD ADDRESS
2808                    *    HEAD  FNUL,PFA,PFA,DOCL
2809          002420    TLNK SET   *
2810 02420    001720         VFD   1,0,1,FNUL,6,3,8,'320
2811                    *                              =<FNUL,3>,'P'
```

```
2812 02421   143101          VFD   8,'306,8,'101     ='FA'
2813 02422   0 002402        DAC   LINK
2814         002420     LINK SET   TLNK
2815 02423   0 10 00130 PFA  JST   DOCL
2816 02424   0 001541        DAC   DUP
2817 02425   0 001616        DAC   BYTE
2818 02426   0 001572        DAC   CAT               GET FIRST BYTE
2819 02427   0 000241        DAC   LIT
2820 02430   000037          OCT   37
2821 02431   0 001300        DAC   AND               LOWER 5 BITS
2822 02432   0 001625        DAC   CELL              CONVERT TO WORD ADDRESS (/ BY 2)
2823 02433   0 002223        DAC   ONEP              ADD ONE (ROUND UP)
2824 02434   0 001443        DAC   PLUS              ADD NFA
2825 02435   0 002231        DAC   TWOP              SKIP LINK AND CODE FIELDS
2826 02436   0 001361        DAC   SMIS
2827                     *
2828                     *  THE NEXT 7 OPERATIONS ARE USED BY THE COMPILER, FOR
2829                     *  COMPILE-TIME SYNTAX-ERROR CHECKS.
2830                     *
2831                     * **** !CSP ****
2832                     *    HEAD  FNUL,!CSP,SCSP,DOCL
2833         002437      TLNK SET   *
2834 02437   002241          VFD   1,0,1,FNUL,6,4,8,'241
2835                     *                              =<FNUL,4>,'!'
2836 02440   141723          BCI   1,CS
2837 02441   050000          VFD   8,'120              ='P'
2838 02442   0 002420        DAC   LINK
2839         002437      LINK SET   TLNK
2840 02443   0 10 00130 SCSP JST   DOCL
2841 02444   0 001324        DAC   SPAT
2842 02445   0 002171        DAC   CSP
2843 02446   0 001577        DAC   STOR
2844 02447   0 001361        DAC   SMIS
2845                     *
2846                     * **** ?ERROR ****
2847                     *    HEAD  FNUL,?ERROR,QERR,DOCL
2848         002450      TLNK SET   *
2849 02450   003277          VFD   1,0,1,FNUL,6,6,8,'277
2850                     *                              =<FNUL,6>,'?'
2851 02451   142722          BCI   2,ERRO
     02452   151317
2852 02453   051000          VFD   8,'122              ='R'
2853 02454   0 002437        DAC   LINK
2854         002450      LINK SET   TLNK
2855 02455   0 10 00130 QERR JST   DOCL
2856 02456   0 001533        DAC   SWAP
2857 02457   0 000277        DAC   ZBRA
2858 02460   0 002464        DAC   QER1
2859 02461   0 003662        DAC   EROR
2860 02462   0 000263        DAC   BRAN
```

```
2861 02463   0 002465        DAC   QER2
2862 02464   0 001526   QER1 DAC   DROP
2863 02465   0 001361   QER2 DAC   SMIS
2864                    *
2865                    * **** ?COMP ****
2866                    *    HEAD  FNUL,?COMP,QCMP,DOCL
2867         002466     TLNK SET   *
2868 02466   002677          VFD   1,0,1,FNUL,6,5,8,'277
2869                    *                              =<FNUL,5>,'?'
2870 02467   141717          BCI   1,CO
2871 02470   146520          VFD   8,'315,8,'120     ='MP'
2872 02471   0 002450        DAC   LINK
2873         002466     LINK SET   TLNK
2874 02472   0 10 00130 QCMP JST   DOCL
2875 02473   0 002144        DAC   STAT
2876 02474   0 001565        DAC   AT
2877 02475   0 001423        DAC   ZEQU
2878 02476   0 000241        DAC   LIT
2879 02477   000021          OCT   21
2880 02500   0 002455        DAC   QERR
2881 02501   0 001361        DAC   SMIS
2882                    *
2883                    * **** ?EXEC ****
2884                    *    HEAD  FNUL,?EXEC,QEXC,DOCL
2885         002502     TLNK SET   *
2886 02502   002677          VFD   1,0,1,FNUL,6,5,8,'277
2887                    *                              =<FNUL,5>,'?'
2888 02503   142730          BCI   1,EX
2889 02504   142503          VFD   8,'305,8,'103     ='EC'
2890 02505   0 002466        DAC   LINK
2891         002502     LINK SET   TLNK
2892 02506   0 10 00130 QEXC JST   DOCL
2893 02507   0 002144        DAC   STAT
2894 02510   0 001565        DAC   AT
2895 02511   0 000241        DAC   LIT
2896 02512   000022          OCT   22
2897 02513   0 002455        DAC   QERR
2898 02514   0 001361        DAC   SMIS
2899                    *
2900                    * **** ?PAIRS ****
2901                    *    HEAD  FNUL,?PAIRS,QPRS,DOCL
2902         002515     TLNK SET   *
2903 02515   003277          VFD   1,0,1,FNUL,6,6,8,'277
2904                    *                              =<FNUL,6>,'?'
2905 02516   150301          BCI   2,PAIR
     02517   144722
2906 02520   051400          VFD   8,'123            ='S'
2907 02521   0 002502        DAC   LINK
2908         002515     LINK SET   TLNK
2909 02522   0 10 00130 QPRS JST   DOCL
```

```
2910 02523   0 002266        DAC   SUB
2911 02524   0 000241        DAC   LIT
2912 02525   000023          OCT   23
2913 02526   0 002455        DAC   QERR
2914 02527   0 001361        DAC   SMIS
2915                 *
2916                 * **** ?CSP ****
2917                 *    HEAD  FNUL,?CSP,QCSP,DOCL
2918         002530     TLNK SET   *
2919 02530   002277          VFD   1,0,1,FNUL,6,4,8,'277
2920                 *                            =<FNUL,4>,'?'
2921 02531   141723          BCI   1,CS
2922 02532   050000          VFD   8,'120           ='P'
2923 02533   0 002515        DAC   LINK
2924         002530     LINK SET   TLNK
2925 02534   0 10 00130 QCSP JST   DOCL
2926 02535   0 001324        DAC   SPAT
2927 02536   0 002171        DAC   CSP
2928 02537   0 001565        DAC   AT
2929 02540   0 002266        DAC   SUB
2930 02541   0 000241        DAC   LIT
2931 02542   000024          OCT   24
2932 02543   0 002455        DAC   QERR
2933 02544   0 001361        DAC   SMIS
2934                 *
2935                 * **** ?LOADING ****
2936                 *    HEAD  FNUL,?LOADING,QLDG,DOCL
2937         002545     TLNK SET   *
2938 02545   004277          VFD   1,0,1,FNUL,6,8,8,'277
2939                 *                            =<FNUL,8>,'?'
2940 02546   146317          BCI   3,LOADIN
     02547   140704
     02550   144716
2941 02551   043400          VFD   8,'107           ='G'
2942 02552   0 002530        DAC   LINK
2943         002545     LINK SET   TLNK
2944 02553   0 10 00130 QLDG JST   DOCL
2945 02554   0 002072        DAC   BLK
2946 02555   0 001565        DAC   AT
2947 02556   0 001423        DAC   ZEQU
2948 02557   0 000241        DAC   LIT
2949 02560   000026          OCT   26
2950 02561   0 002455        DAC   QERR
2951 02562   0 001361        DAC   SMIS
2952                 *
2953                 * **** COMPILE ****
2954                 *    HEAD  FNUL,COMPILE,COMP,DOCL
2955         002563     TLNK SET   *
2956 02563   003703          VFD   1,0,1,FNUL,6,7,8,'303
2957                 *                            =<FNUL,7>,'C'
```

```
2958 02564   147715          BCI   2,OMPI
     02565   150311
2959 02566   146105          VFD   8,'314,8,'105     ='LE'
2960 02567   0 002545        DAC   LINK
2961         002563     LINK SET   TLNK
2962 02570   0 10 00130 COMP JST   DOCL
2963 02571   0 002472        DAC   QCMP
2964 02572   0 001412        DAC   FRMR
2965 02573   0 001541        DAC   DUP
2966 02574   0 002223        DAC   ONEP
2967 02575   0 001402        DAC   TOR
2968 02576   0 001565        DAC   AT
2969 02577   0 002256        DAC   COMA
2970 02600   0 001361        DAC   SMIS
2971                     *
2972                     * **** [ **** STOP COMPILATION, ENTER EXECUTION STATE.
2973                     *    HEAD  FIMD,[,LBRC,DOCL
2974         002601     TLNK SET   *
2975 02601   040533          VFD   1,0,1,FIMD,6,1,8,'133
2976                     *                             =<FIMD,1>,'['
2977 02602   0 002563        DAC   LINK
2978         002601     LINK SET   TLNK
2979 02603   0 10 00130 LBRC JST   DOCL
2980 02604   0 001716        DAC   ZERO
2981 02605   0 002144        DAC   STAT
2982 02606   0 001577        DAC   STOR
2983 02607   0 001361        DAC   SMIS
2984                     *
2985                     * **** ] **** ENTER COMPILATION STATE.
2986                     *    HEAD  FNUL,],RBRC,DOCL
2987         002610     TLNK SET   *
2988 02610   000535          VFD   1,0,1,FNUL,6,1,8,'135
2989                     *                             =<FNUL,1>,']'
2990 02611   0 002601        DAC   LINK
2991         002610     LINK SET   TLNK
2992 02612   0 10 00130 RBRC JST   DOCL
2993 02613   0 000241        DAC   LIT
2994 02614   000100          OCT   100               NOT 300 BECAUSE MS BIT CLEAR ON H16
2995 02615   0 002144        DAC   STAT
2996 02616   0 001577        DAC   STOR
2997 02617   0 001361        DAC   SMIS
2998                     *
2999                     * **** SMUDGE **** ALTER LATEST WORD NAME (SO THAT DICTIONARY SEARCH
3000                     *                  WON'T FIND A PARTIALLY-COMPLETE ENTRY).
3001                     *    HEAD  FNUL,SMUDGE,SMDG,DOCL
3002         002620     TLNK SET   *
3003 02620   003323          VFD   1,0,1,FNUL,6,6,8,'323
3004                     *                             =<FNUL,6>,'S'
3005 02621   146725          BCI   2,MUDG
     02622   142307
```

```
3006 02623   042400          VFD   8,'105           ='E'
3007 02624   0 002610        DAC   LINK
3008         002620     LINK SET   TLNK
3009 02625   0 10 00130 SMDG JST   DOCL
3010 02626   0 002357        DAC   LTST
3011 02627   0 000241        DAC   LIT
3012 02630   020000          OCT   020000
3013 02631   0 001557        DAC   TOGL
3014 02632   0 001361        DAC   SMIS
3015                 *
3016                 * **** HEX ****
3017                 *    HEAD  FNUL,HEX,HEX,DOCL
3018         002633     TLNK SET   *
3019 02633   001710          VFD   1,0,1,FNUL,6,3,8,'310
3020                 *                             =<FNUL,3>,'H'
3021 02634   142530          VFD   8,'305,8,'130     ='EX'
3022 02635   0 002620        DAC   LINK
3023         002633     LINK SET   TLNK
3024 02636   0 10 00130 HEX  JST   DOCL
3025 02637   0 000241        DAC   LIT
3026 02640   000020          DEC   16
3027 02641   0 002152        DAC   BASE
3028 02642   0 001577        DAC   STOR
3029 02643   0 001361        DAC   SMIS
3030                 *
3031                 * **** DECIMAL ****
3032                 *    HEAD  FNUL,DECIMAL,DEC,DOCL
3033         002644     TLNK SET   *
3034 02644   003704          VFD   1,0,1,FNUL,6,7,8,'304
3035                 *                             =<FNUL,7>,'D'
3036 02645   142703          BCI   2,ECIM
     02646   144715
3037 02647   140514          VFD   8,'301,8,'114     ='AL'
3038 02650   0 002633        DAC   LINK
3039         002644     LINK SET   TLNK
3040 02651   0 10 00130 DEC  JST   DOCL
3041 02652   0 000241        DAC   LIT
3042 02653   000012          DEC   10
3043 02654   0 002152        DAC   BASE
3044 02655   0 001577        DAC   STOR
3045 02656   0 001361        DAC   SMIS
3046                 *
3047                 * **** OCT ****
3048                 *    HEAD  FNUL,OCT,OCT,DOCL
3049         002657     TLNK SET   *
3050 02657   001717          VFD   1,0,1,FNUL,6,3,8,'317
3051                 *                             =<FNUL,3>,'O'
3052 02660   141524          VFD   8,'303,8,'124     ='CT'
3053 02661   0 002644        DAC   LINK
3054         002657     LINK SET   TLNK
```

```
3055 02662   0 10 00130 OCT  JST   DOCL
3056 02663   0 000241        DAC   LIT
3057 02664   000010          DEC   8
3058 02665   0 002152        DAC   BASE
3059 02666   0 001577        DAC   STOR
3060 02667   0 001361        DAC   SMIS
3061                 *
3062                 * **** (;CODE) ****
3063                 *    HEAD  FNUL,(;CODE),PSCD,DOCL
3064         002670     TLNK SET   *
3065 02670   003650          VFD   1,0,1,FNUL,6,7,8,'250
3066                 *                              =<FNUL,7>,'('
3067 02671   135703          BCI   2,;COD
     02672   147704
3068 02673   142451          VFD   8,'305,8,'051     ='E)'
3069 02674   0 002657        DAC   LINK
3070         002670     LINK SET   TLNK
3071 02675   0 10 00130 PSCD JST   DOCL
3072 02676   0 001412        DAC   FRMR              SHOULD POINT AT JST INSTRUCTION
3073 02677   0 001565        DAC   AT                PICK UP THAT INSTRUCTION
3074 02700   0 002357        DAC   LTST
3075 02701   0 002423        DAC   PFA
3076 02702   0 002376        DAC   CFA
3077 02703   0 001577        DAC   STOR
3078 02704   0 001361        DAC   SMIS
3079                 *
3080                 *  ***** THE DEFINITION OF ';CODE' WAS MOVED TO THE END OF
3081                 *  THE DICTIONARY, BECAUSE IT IS NOT PURE CODE (IT IS PATCHED
3082                 *  WHEN A FORTH ASSEMBLER IS LOADED).
3083                 *
3084                 *
3085                 * **** <BUILDS **** CREATE NEW DATA TYPE WITH CODE ROUTINE IN
3086                 *                   HIGHER-LEVEL FORTH.
3087                 *    HEAD  FNUL,<BUILDS,BULD,DOCL
3088         002705     TLNK SET   *
3089 02705   003674          VFD   1,0,1,FNUL,6,7,8,'274
3090                 *                              =<FNUL,7>,'<'
3091 02706   141325          BCI   2,BUIL
     02707   144714
3092 02710   142123          VFD   8,'304,8,'123     ='DS'
3093 02711   0 002670        DAC   LINK
3094         002705     LINK SET   TLNK
3095 02712   0 10 00130 BULD JST   DOCL
3096 02713   0 001716        DAC   ZERO
3097 02714   0 001664        DAC   CON
3098 02715   0 001361        DAC   SMIS
3099                 *
3100                 * **** DOES> ****
3101                 *    HEAD  FNUL,DOES>,DOES,DOCL
3102         002716     TLNK SET   *
```

```
3103 02716   002704          VFD   1,0,1,FNUL,6,5,8,'304
3104                       *                           =<FNUL,5>,'D'
3105 02717   147705          BCI   1,OE
3106 02720   151476          VFD   8,'323,8,'076       ='S>'
3107 02721   0 002705        DAC   LINK
3108         002716     LINK SET   TLNK
3109 02722   0 10 00130 DOES JST   DOCL
3110 02723   0 001412        DAC   FRMR
3111 02724   0 002357        DAC   LTST
3112 02725   0 002423        DAC   PFA
3113 02726   0 001577        DAC   STOR
3114 02727   0 002675        DAC   PSCD
3115 02730   0 10 00152      JST   DODS              PICKED UP BY PSCD
3116                       *
3117                       * **** COUNT ****
3118                       * ( ADDR --- CADDR+1 COUNT )
3119                       *    HEAD  FNUL,COUNT,CNT,DOCL
3120         002731     TLNK SET   *
3121 02731   002703          VFD   1,0,1,FNUL,6,5,8,'303
3122                       *                           =<FNUL,5>,'C'
3123 02732   147725          BCI   1,OU
3124 02733   147124          VFD   8,'316,8,'124       ='NT'
3125 02734   0 002716        DAC   LINK
3126         002731     LINK SET   TLNK
3127 02735   0 10 00130 CNT  JST   DOCL
3128 02736   0 001616        DAC   BYTE
3129 02737   0 001541        DAC   DUP
3130 02740   0 002223        DAC   ONEP
3131 02741   0 001533        DAC   SWAP
3132 02742   0 001572        DAC   CAT
3133 02743   0 001361        DAC   SMIS
3134                       *
3135                       * **** TYPE ****
3136                       * ( CADDR COUNT --- ) TYPE STRING OF CHARACTERS
3137                       *    HEAD  FNUL,TYPE,TYPE,DOCL
3138         002744     TLNK SET   *
3139 02744   002324          VFD   1,0,1,FNUL,6,4,8,'324
3140                       *                           =<FNUL,4>,'T'
3141 02745   154720          BCI   1,YP
3142 02746   042400          VFD   8,'105              ='E'
3143 02747   0 002731        DAC   LINK
3144         002744     LINK SET   TLNK
3145 02750   0 10 00130 TYPE JST   DOCL
3146 02751   0 002345        DAC   DDUP
3147 02752   0 000277        DAC   ZBRA
3148 02753   0 002767        DAC   TYP2
3149 02754   0 001520        DAC   OVER
3150 02755   0 001443        DAC   PLUS
3151 02756   0 001533        DAC   SWAP
3152 02757   0 000372        DAC   XDO
```

```
3153 02760   0 000401   TYP1 DAC   I
3154 02761   0 001572        DAC   CAT
3155 02762   0 000610        DAC   EMIT
3156 02763   0 000322        DAC   XLOP
3157 02764   0 002760        DAC   TYP1
3158 02765   0 000263        DAC   BRAN
3159 02766   0 002770        DAC   TYP3
3160 02767   0 001526   TYP2 DAC   DROP
3161 02770   0 001361   TYP3 DAC   SMIS
3162                    *
3163                    * **** -TRAILING **** REDUCE CHARACTER COUNT OF STRING
3164                    *                     TO OMIT TRAILING SPACES
3165                    * ( CADDR COUNT2 --- CADDR COUNT2 )
3166                    *    HEAD  FNUL,-TRAILING,DTRA,DOCL
3167         002771     TLNK SET   *
3168 02771   004655          VFD   1,0,1,FNUL,6,9,8,'255
3169                    *                              =<FNUL,9>,'-'
3170 02772   152322          BCI   3,TRAILI
     02773   140711
     02774   146311
3171 02775   147107          VFD   8,'316,8,'107     ='NG'
3172 02776   0 002744        DAC   LINK
3173         002771     LINK SET   TLNK
3174 02777   0 10 00130 DTRA JST   DOCL
3175 03000   0 001541        DAC   DUP
3176 03001   0 001716        DAC   ZERO
3177 03002   0 000372        DAC   XDO
3178 03003   0 001520   DTR1 DAC   OVER
3179 03004   0 001520        DAC   OVER
3180 03005   0 001443        DAC   PLUS
3181 03006   0 001722        DAC   ONE
3182 03007   0 002266        DAC   SUB
3183 03010   0 001572        DAC   CAT
3184 03011   0 001737        DAC   BL
3185 03012   0 002266        DAC   SUB
3186 03013   0 000277        DAC   ZBRA
3187 03014   0 003020        DAC   DTR2
3188 03015   0 001373        DAC   LEAV
3189 03016   0 000263        DAC   BRAN
3190 03017   0 003022        DAC   DTR3
3191 03020   0 001722   DTR2 DAC   ONE
3192 03021   0 002266        DAC   SUB
3193 03022   0 000322   DTR3 DAC   XLOP
3194 03023   0 003003        DAC   DTR1
3195 03024   0 001361        DAC   SMIS
3196                    *
3197                    * **** (.") **** USED ONLY BY COMPILER.  COMPILED BY '."'
3198                    *    HEAD  FNUL,(."),PDTQ,DOCL
3199         003025     TLNK SET   *
3200 03025   002250          VFD   1,0,1,FNUL,6,4,8,'250
```

```
3201                    *                               =<FNUL,4>,'('
3202 03026   127242          BCI   1,."
3203 03027   024400          VFD   8,'051              =')'
3204 03030   0 002771        DAC   LINK
3205         003025     LINK SET   TLNK
3206 03031   0 10 00130 PDTQ JST   DOCL
3207 03032   0 001416        DAC   R
3208 03033   0 002735        DAC   CNT
3209 03034   0 001541        DAC   DUP
3210 03035   0 001625        DAC   CELL
3211 03036   0 002223        DAC   ONEP
3212 03037   0 001412        DAC   FRMR
3213 03040   0 001443        DAC   PLUS
3214 03041   0 001402        DAC   TOR
3215 03042   0 002750        DAC   TYPE
3216 03043   0 001361        DAC   SMIS
3217                    *
3218                    * **** ." **** TYPE ASCII MESSAGE.
3219                    *    HEAD  FIMD,.",DOTQ,DOCL
3220         003044     TLNK SET   *
3221 03044   041256          VFD   1,0,1,FIMD,6,2,8,'256
3222                    *                               =<FIMD,2>,'.'
3223 03045   021000          VFD   8,'042              ='""'
3224 03046   0 003025        DAC   LINK
3225         003044     LINK SET   TLNK
3226 03047   0 10 00130 DOTQ JST   DOCL
3227 03050   0 000241        DAC   LIT
3228 03051   000242          VFD   16,CDQT
3229 03052   0 002144        DAC   STAT
3230 03053   0 001565        DAC   AT
3231 03054   0 000277        DAC   ZBRA
3232 03055   0 003071        DAC   DTQ1
3233 03056   0 002570        DAC   COMP
3234 03057   0 003031        DAC   PDTQ
3235 03060   0 003363        DAC   WORD
3236 03061   0 002240        DAC   HERE
3237 03062   0 001616        DAC   BYTE
3238 03063   0 001572        DAC   CAT
3239 03064   0 001625        DAC   CELL
3240 03065   0 002223        DAC   ONEP
3241 03066   0 002250        DAC   ALOT
3242 03067   0 000263        DAC   BRAN
3243 03070   0 003075        DAC   DTQ2
3244 03071   0 003363   DTQ1 DAC   WORD
3245 03072   0 002240        DAC   HERE
3246 03073   0 002735        DAC   CNT
3247 03074   0 002750        DAC   TYPE
3248 03075   0 001361   DTQ2 DAC   SMIS
3249                    *
3250                    * **** EXPECT **** READ N CHARACTERS TO MEMORY
```

```
3251                    *                 (AND TERMINATE WITH NULLS).
3252                    * ( ADDR N --- ) WORD ADDRESS!
3253                    *    HEAD  FNUL,EXPECT,EXPC,DOCL
3254          003076    TLNK SET   *
3255 03076   003305          VFD   1,0,1,FNUL,6,6,8,'305
3256                    *                              =<FNUL,6>,'E'
3257 03077   154320          BCI   2,XPEC
     03100   142703
3258 03101   052000          VFD   8,'124               ='T'
3259 03102   0 003044        DAC   LINK
3260          003076    LINK SET   TLNK
3261 03103   0 10 00130 EXPC JST   DOCL
3262 03104   0 001533        DAC   SWAP
3263 03105   0 001616        DAC   BYTE
3264 03106   0 001533        DAC   SWAP
3265 03107   0 001520        DAC   OVER
3266 03110   0 001443        DAC   PLUS
3267 03111   0 001520        DAC   OVER
3268 03112   0 000372        DAC   XDO
3269          003113    EXP1 EQU   *
3270                          IFN   PTW
3271 03113   0 002072        DAC   BLK
3272 03114   0 001565        DAC   AT
3273 03115   0 001435        DAC   ZLES              -VE INDICATES PAPERTAPE
3274 03116   0 000277        DAC   ZBRA
3275 03117   0 003123        DAC   EXP2
3276 03120   0 004767        DAC   PTRK
3277 03121   0 000263        DAC   BRAN
3278 03122   0 003124        DAC   EXP3
3279                          ENDC
3280 03123   0 000614   EXP2 DAC   KEY
3281 03124   0 001541   EXP3 DAC   DUP
3282 03125   0 000241        DAC   LIT
3283 03126   000377          VFD   16,CDEL
3284 03127   0 002273        DAC   EQAL
3285 03130   0 000277        DAC   ZBRA
3286 03131   0 003151        DAC   EXP4
3287 03132   0 001526        DAC   DROP
3288 03133   0 000241        DAC   LIT
3289 03134   000210          VFD   16,CBS
3290 03135   0 001520        DAC   OVER
3291 03136   0 000401        DAC   I
3292 03137   0 002273        DAC   EQAL
3293 03140   0 001541        DAC   DUP
3294 03141   0 001412        DAC   FRMR
3295 03142   0 001726        DAC   TWO
3296 03143   0 002266        DAC   SUB
3297 03144   0 001443        DAC   PLUS
3298 03145   0 001402        DAC   TOR
3299 03146   0 002266        DAC   SUB
```

```
3300 03147   0 000263        DAC   BRAN
3301 03150   0 003200        DAC   EXP7
3302 03151   0 001541   EXP4 DAC   DUP
3303 03152   0 000241        DAC   LIT
3304 03153   000215          VFD   16,CCR
3305 03154   0 002273        DAC   EQAL
3306 03155   0 000277        DAC   ZBRA
3307 03156   0 003165        DAC   EXP5
3308 03157   0 001373        DAC   LEAV
3309 03160   0 001526        DAC   DROP
3310 03161   0 001737        DAC   BL
3311 03162   0 001716        DAC   ZERO
3312 03163   0 000263        DAC   BRAN
3313 03164   0 003166        DAC   EXP6
3314 03165   0 001541   EXP5 DAC   DUP
3315 03166   0 000401   EXP6 DAC   I
3316 03167   0 001605        DAC   CSTR
3317 03170   0 001716        DAC   ZERO
3318 03171   0 000401        DAC   I
3319 03172   0 002223        DAC   ONEP
3320 03173   0 001605        DAC   CSTR
3321 03174   0 001716        DAC   ZERO
3322 03175   0 000401        DAC   I
3323 03176   0 002231        DAC   TWOP
3324 03177   0 001605        DAC   CSTR
3325                         IFZ   ECHO
3326 03200   0 001526   EXP7 DAC   DROP
3327                         ELSE
3328                         IFN   PTW
3329                    EXP7 DAC   BLK
3330                         DAC   AT
3331                         DAC   ZEQU              ASR?
3332                         DAC   ZBRA
3333                         DAC   EXP8
3334                         DAC   EMIT
3335                         DAC   BRAN
3336                         DAC   EXP9
3337                    EXP8 DAC   DROP
3338                    EXP9 EQU   *
3339                         ELSE
3340                    EXP7 DAC   EMIT
3341                         ENDC
3342                         ENDC
3343 03201   0 000322        DAC   XLOP
3344 03202   0 003113        DAC   EXP1
3345 03203   0 001526        DAC   DROP
3346 03204   0 001361        DAC   SMIS
3347                    *
3348                    * **** QUERY ****
3349                    *    HEAD  FNUL,QUERY,QURY,DOCL
```

```
3350         003205     TLNK SET   *
3351 03205   002721          VFD   1,0,1,FNUL,6,5,8,'321
3352                        *                              =<FNUL,5>,'Q'
3353 03206   152705          BCI   1,UE
3354 03207   151131          VFD   8,'322,8,'131     ='RY'
3355 03210   0 003076        DAC   LINK
3356         003205     LINK SET   TLNK
3357 03211   0 10 00130 QURY JST   DOCL
3358 03212   0 002011        DAC   TIB
3359 03213   0 001565        DAC   AT
3360 03214   0 000241        DAC   LIT
3361 03215   000120          OCT   120
3362 03216   0 003103        DAC   EXPC
3363 03217   0 001716        DAC   ZERO
3364 03220   0 002077        DAC   IN
3365 03221   0 001577        DAC   STOR
3366 03222   0 001361        DAC   SMIS
3367                        *
3368                        * **** THE NULL ****
3369         003223     TLNK SET   *
3370 03223   040400          VFD   1,0,1,FIMD,6,1,1,0,7,0
3371                        *                              =<FIMD,1>,NULL
3372 03224   0 003205        DAC   LINK
3373         003223     LINK SET   TLNK
3374 03225   0 10 00130 NULL JST   DOCL
3375                        *  THE NULL OPERATION (ASCII 0) STOPS INTERPRETATION/COMPILATION
3376                        *  AT END OF A TERMINAL INPUT LINE, OR A DISK SCREEN.  ALL DISK
3377                        *  BUFFERS MUST TERMINATE WITH NULLS, AND 'EXPECT' PLACES NULLS
3378                        *  AFTER EACH TERMINAL INPUT LINE.
3379 03226   0 002072          DAC   BLK
3380 03227   0 001565          DAC   AT
3381                           IFN   PTW
3382 03230   0 002223          DAC   ONEP              -1 -> 0, 0 -> 1
3383 03231   0 000241          DAC   LIT
3384 03232   177776            OCT   177776
3385 03233   0 001300          DAC   AND               ASR OR PTR -> ZERO
3386                           ENDC
3387 03234   0 000277          DAC   ZBRA
3388 03235   0 003260          DAC   NUL2
3389 03236   0 001722          DAC   ONE
3390 03237   0 002072          DAC   BLK
3391 03240   0 001546          DAC   PSTR
3392 03241   0 001716          DAC   ZERO
3393 03242   0 002077          DAC   IN
3394 03243   0 001577          DAC   STOR
3395 03244   0 002072          DAC   BLK
3396 03245   0 001565          DAC   AT
3397 03246   0 001760          DAC   BSCR
3398 03247   0 004705          DAC   MOD
3399 03250   0 001423          DAC   ZEQU
```

```
3400 03251   0 000277        DAC   ZBRA
3401 03252   0 003256        DAC   NUL1
3402 03253   0 002506        DAC   QEXC
3403 03254   0 001412        DAC   FRMR              RETURN FROM INTERPRET
3404 03255   0 001526        DAC   DROP
3405 03256   0 000263   NUL1 DAC   BRAN
3406 03257   0 003262        DAC   NUL3
3407 03260   0 001412   NUL2 DAC   FRMR              RETURN FROM INTERPRET
3408 03261   0 001526        DAC   DROP
3409 03262   0 001361   NUL3 DAC   SMIS
3410                    *
3411                    * **** FILL **** FILL WORDS
3412                    * ( ADDR COUNT PATTERN --- )
3413                    *    HEAD  FNUL,FILL,FILL,DOCL
3414         003263     TLNK SET   *
3415 03263   002306          VFD   1,0,1,FNUL,6,4,8,'306
3416                    *                              =<FNUL,4>,'F'
3417 03264   144714          BCI   1,IL
3418 03265   046000          VFD   8,'114              ='L'
3419 03266   0 003223        DAC   LINK
3420         003263     LINK SET   TLNK
3421 03267   0 10 00130 FILL JST   DOCL
3422 03270   0 001533        DAC   SWAP
3423 03271   0 001402        DAC   TOR
3424 03272   0 001520        DAC   OVER
3425 03273   0 001577        DAC   STOR
3426 03274   0 001541        DAC   DUP
3427 03275   0 002223        DAC   ONEP
3428 03276   0 001412        DAC   FRMR
3429 03277   0 001722        DAC   ONE
3430 03300   0 002266        DAC   SUB
3431 03301   0 000671        DAC   MOVE
3432 03302   0 001361        DAC   SMIS
3433                    *
3434                    * **** ERASE **** ERASE WORDS
3435                    * ( ADDR COUNT --- )
3436                    *    HEAD  FNUL,ERASE,ERAS,DOCL
3437         003303     TLNK SET   *
3438 03303   002705          VFD   1,0,1,FNUL,6,5,8,'305
3439                    *                              =<FNUL,5>,'E'
3440 03304   151301          BCI   1,RA
3441 03305   151505          VFD   8,'323,8,'105       ='SE'
3442 03306   0 003263        DAC   LINK
3443         003303     LINK SET   TLNK
3444 03307   0 10 00130 ERAS JST   DOCL
3445 03310   0 001716        DAC   ZERO
3446 03311   0 003267        DAC   FILL
3447 03312   0 001361        DAC   SMIS
3448                    *
3449                    * **** BLANKS **** FILL WORDS WITH SPACE CHARACTERS
```

```
3450                    * ( ADDR COUNT --- )
3451                    *    HEAD  FNUL,BLANKS,BLKS,DOCL
3452           003313   TLNK SET   *
3453 03313     003302         VFD   1,0,1,FNUL,6,6,8,'302
3454                    *                              =<FNUL,6>,'B'
3455 03314     146301         BCI   2,LANK
     03315     147313
3456 03316     051400         VFD   8,'123             ='S'
3457 03317     0 003303       DAC   LINK
3458           003313   LINK SET   TLNK
3459 03320     0 10 00130 BLKS JST   DOCL
3460 03321     0 001737       DAC   BL
3461 03322     0 001541       DAC   DUP
3462 03323     0 000241       DAC   LIT
3463 03324     000400         DEC   256
3464 03325     0 001120       DAC   USTR              SHIFT TO UPPER BYTE
3465 03326     0 001306       DAC   OR                OR IN LOWER BYTE
3466 03327     0 003267       DAC   FILL
3467 03330     0 001361       DAC   SMIS
3468                    *
3469                    * **** HOLD ****
3470                    *    HEAD  FNUL,HOLD,HOLD,DOCL
3471           003331   TLNK SET   *
3472 03331     002310         VFD   1,0,1,FNUL,6,4,8,'310
3473                    *                              =<FNUL,4>,'H'
3474 03332     147714         BCI   1,OL
3475 03333     042000         VFD   8,'104             ='D'
3476 03334     0 003313       DAC   LINK
3477           003331   LINK SET   TLNK
3478 03335     0 10 00130 HOLD JST   DOCL
3479 03336     0 000241       DAC   LIT
3480 03337     177777         DEC   -1
3481 03340     0 002203       DAC   HLD
3482 03341     0 001546       DAC   PSTR
3483 03342     0 002203       DAC   HLD
3484 03343     0 001565       DAC   AT
3485 03344     0 001605       DAC   CSTR
3486 03345     0 001361       DAC   SMIS
3487                    *
3488                    * **** PAD ****
3489                    *    HEAD  FNUL,PAD,PAD,DOCL
3490           003346   TLNK SET   *
3491 03346     001720         VFD   1,0,1,FNUL,6,3,8,'320
3492                    *                              =<FNUL,3>,'P'
3493 03347     140504         VFD   8,'301,8,'104      ='AD'
3494 03350     0 003331       DAC   LINK
3495           003346   LINK SET   TLNK
3496 03351     0 10 00130 PAD  JST   DOCL
3497 03352     0 002240       DAC   HERE
3498 03353     0 000241       DAC   LIT
```

```
3499 03354   000042          VFD   16,KPAD
3500 03355   0 001443        DAC   PLUS
3501 03356   0 001361        DAC   SMIS
3502                        *
3503                        * **** WORD **** READ NEXT WORD FROM INPUT STREAM USING CHAR
3504                        *                AS DELIMITER
3505                        * ( CHAR --- )
3506                        *    HEAD  FNUL,WORD,WORD,DOCL
3507         003357     TLNK SET   *
3508 03357   002327          VFD   1,0,1,FNUL,6,4,8,'327
3509                        *                              =<FNUL,4>,'W'
3510 03360   147722          BCI   1,OR
3511 03361   042000          VFD   8,'104                   ='D'
3512 03362   0 003346        DAC   LINK
3513         003357     LINK SET   TLNK
3514 03363   0 10 00130 WORD JST   DOCL
3515 03364   0 002072        DAC   BLK                      DISK BLOCK
3516 03365   0 001565        DAC   AT
3517                        IFN   PTW
3518 03366   0 002223        DAC   ONEP                     -1 -> 0, 0 -> 1
3519 03367   0 000241        DAC   LIT
3520 03370   177776          OCT   177776
3521 03371   0 001300        DAC   AND                      ASR OR PTR -> ZERO
3522                        ENDC
3523 03372   0 000277        DAC   ZBRA                     ZERO IS TERMINAL
3524 03373   0 003400        DAC   WRD1
3525                        IFN   DISK
3526                        DAC   BLK
3527                        DAC   AT
3528                        DAC   BLCK
3529                        DAC   BRAN
3530                        DAC   WRD2
3531                        ELSE
3532 03374   0 001722        DAC   ONE                      TRUE - ALWAYS
3533 03375   0 000241        DAC   LIT
3534 03376   000006          OCT   6                        DISK RANGE
3535 03377   0 003662        DAC   EROR
3536                        ENDC
3537 03400   0 002011   WRD1 DAC   TIB
3538 03401   0 001565        DAC   AT
3539 03402   0 001616   WRD2 DAC   BYTE
3540 03403   0 002077        DAC   IN                       OFFSET IN INPUT BUFFER
3541 03404   0 001565        DAC   AT
3542 03405   0 001443        DAC   PLUS                     ADD TO BASE ADDRESS
3543 03406   0 001533        DAC   SWAP
3544 03407   0 000534        DAC   ENCL
3545 03410   0 002240        DAC   HERE
3546 03411   0 000241        DAC   LIT
3547 03412   000021          DEC   17                       34 BYTES
3548 03413   0 003307        DAC   ERAS
```

```
3549 03414   0 002077        DAC   IN
3550 03415   0 001546        DAC   PSTR
3551 03416   0 001520        DAC   OVER
3552 03417   0 002266        DAC   SUB
3553 03420   0 001402        DAC   TOR
3554 03421   0 001416        DAC   R
3555 03422   0 002240        DAC   HERE
3556 03423   0 001616        DAC   BYTE
3557 03424   0 001605        DAC   CSTR
3558 03425   0 001443        DAC   PLUS
3559 03426   0 002240        DAC   HERE
3560 03427   0 001616        DAC   BYTE
3561 03430   0 002223        DAC   ONEP
3562 03431   0 001412        DAC   FRMR
3563 03432   0 000641        DAC   CMOV
3564 03433   0 001361        DAC   SMIS
3565                     *
3566                     * **** (NUMBER) ****
3567                     * ( D1 CADDR1 --- D2 CADDR2 )
3568                     *    HEAD  FNUL,(NUMBER),PNUM,DOCL
3569         003434      TLNK SET   *
3570 03434   004250          VFD   1,0,1,FNUL,6,8,8,'250
3571                     *                              =<FNUL,8>,'('
3572 03435   147325          BCI   3,NUMBER
     03436   146702
     03437   142722
3573 03440   024400          VFD   8,'051               =')'
3574 03441   0 003357        DAC   LINK
3575         003434      LINK SET   TLNK
3576 03442   0 10 00130  PNUM JST   DOCL
3577 03443   0 002223    PNM1 DAC   ONEP
3578 03444   0 001541        DAC   DUP
3579 03445   0 001402        DAC   TOR
3580 03446   0 001572        DAC   CAT
3581 03447   0 002152        DAC   BASE
3582 03450   0 001565        DAC   AT
3583 03451   0 000407        DAC   DIGT
3584 03452   0 000277        DAC   ZBRA
3585 03453   0 003501        DAC   PNM3
3586 03454   0 001533        DAC   SWAP
3587 03455   0 002152        DAC   BASE
3588 03456   0 001565        DAC   AT
3589 03457   0 001120        DAC   USTR
3590 03460   0 001526        DAC   DROP
3591 03461   0 002325        DAC   ROT
3592 03462   0 002152        DAC   BASE
3593 03463   0 001565        DAC   AT
3594 03464   0 001120        DAC   USTR
3595 03465   0 001451        DAC   DPLS
3596 03466   0 002157        DAC   DPL
```

```
3597 03467   0 001565        DAC   AT
3598 03470   0 002223        DAC   ONEP
3599 03471   0 000277        DAC   ZBRA
3600 03472   0 003476        DAC   PNM2
3601 03473   0 001722        DAC   ONE
3602 03474   0 002157        DAC   DPL
3603 03475   0 001546        DAC   PSTR
3604 03476   0 001412   PNM2 DAC   FRMR
3605 03477   0 000263        DAC   BRAN
3606 03500   0 003443        DAC   PNM1
3607 03501   0 001412   PNM3 DAC   FRMR
3608 03502   0 001361        DAC   SMIS
3609                    *
3610                    * **** NUMBER ****
3611                    * ( CADDR --- D )
3612                    *    HEAD  FNUL,NUMBER,NUMB,DOCL
3613         003503     TLNK SET   *
3614 03503   003316          VFD   1,0,1,FNUL,6,6,8,'316
3615                    *                              =<FNUL,6>,'N'
3616 03504   152715          BCI   2,UMBE
     03505   141305
3617 03506   051000          VFD   8,'122               ='R'
3618 03507   0 003434        DAC   LINK
3619         003503     LINK SET   TLNK
3620 03510   0 10 00130 NUMB JST   DOCL
3621 03511   0 001716        DAC   ZERO
3622 03512   0 001716        DAC   ZERO
3623 03513   0 002325        DAC   ROT
3624 03514   0 001541        DAC   DUP
3625 03515   0 002223        DAC   ONEP
3626 03516   0 001572        DAC   CAT
3627 03517   0 000241        DAC   LIT
3628 03520   000255          VFD   16,CMNS
3629 03521   0 002273        DAC   EQAL
3630 03522   0 001541        DAC   DUP
3631 03523   0 001402        DAC   TOR
3632 03524   0 001443        DAC   PLUS
3633 03525   0 000241        DAC   LIT
3634 03526   0 177777        DAC   -1
3635 03527   0 002157   NUM1 DAC   DPL
3636 03530   0 001577        DAC   STOR
3637 03531   0 003442        DAC   PNUM
3638 03532   0 001541        DAC   DUP
3639 03533   0 001572        DAC   CAT
3640 03534   0 000277        DAC   ZBRA
3641 03535   0 003550        DAC   NUM2
3642 03536   0 001541        DAC   DUP
3643 03537   0 001572        DAC   CAT
3644 03540   0 000241        DAC   LIT
3645 03541   000256          VFD   16,CDOT
```

```
3646 03542   0 002266        DAC   SUB
3647 03543   0 001716        DAC   ZERO
3648 03544   0 002455        DAC   QERR
3649 03545   0 001716        DAC   ZERO
3650 03546   0 000263        DAC   BRAN
3651 03547   0 003527        DAC   NUM1
3652 03550   0 001526   NUM2 DAC   DROP
3653 03551   0 001412        DAC   FRMR
3654 03552   0 000277        DAC   ZBRA
3655 03553   0 003555        DAC   NUM3
3656 03554   0 001502        DAC   DMNS
3657 03555   0 001361   NUM3 DAC   SMIS
3658                    *
3659                    * **** -FIND ****
3660                    * ( --- PFA B TF ) (FOUND)
3661                    * ( --- FF ) (NOT FOUND)
3662                    *
3663                    * ACCEPTS THE NEXT TEXT WORD (DELIMITED BY BLANKS) IN THE
3664                    * INPUT STREAM TO HERE, AND SEARCHES THE CONTEXT AND THEN
3665                    * CURRENT VOCABULARIES FOR A MATCHING ENTRY. IF FOUND, THE
3666                    * DICTIONARY ENTRY'S PARAMETER FIELD ADDRESS, ITS LENGTH
3667                    * BYTE, AND A BOOLEAN TRUE IS LEFT. OTHERWISE, ONLY A
3668                    * BOOLEAN FALSE IS LEFT.
3669                    *
3670                    *    HEAD  FNUL,-FIND,DFND,DOCL
3671         003556     TLNK SET   *
3672 03556   002655          VFD   1,0,1,FNUL,6,5,8,'255
3673                    *                              =<FNUL,5>,'-'
3674 03557   143311          BCI   1,FI
3675 03560   147104          VFD   8,'316,8,'104     ='ND'
3676 03561   0 003503        DAC   LINK
3677         003556     LINK SET   TLNK
3678 03562   0 10 00130 DFND JST   DOCL
3679 03563   0 001737        DAC   BL
3680 03564   0 003363        DAC   WORD
3681 03565   0 002240        DAC   HERE
3682 03566   0 002735        DAC   CNT
3683 03567   0 003612        DAC   UPPR
3684 03570   0 002240        DAC   HERE
3685 03571   0 002127        DAC   CONT
3686 03572   0 001565        DAC   AT
3687 03573   0 001565        DAC   AT
3688 03574   0 000440        DAC   PFND
3689 03575   0 001541        DAC   DUP
3690 03576   0 001423        DAC   ZEQU
3691 03577   0 000277        DAC   ZBRA
3692 03600   0 003605        DAC   DFN1
3693 03601   0 001526        DAC   DROP
3694 03602   0 002240        DAC   HERE
3695 03603   0 002357        DAC   LTST
```

```
3696 03604   0 000440        DAC   PFND
3697 03605   0 001361   DFN1 DAC   SMIS
3698                     *
3699                     * **** UPPER **** SETS STRINGS TO UPPER CASE - TO ALLOW
3700                     *                 LOWER AS WELL AS UPPER CASE FROM TERMINAL.
3701                     * ( COUNT CADDR --- )
3702                     *    HEAD  FNUL,UPPER,UPPR,DOCL
3703         003606      TLNK SET   *
3704 03606   002725          VFD   1,0,1,FNUL,6,5,8,'325
3705                     *                              =<FNUL,5>,'U'
3706 03607   150320          BCI   1,PP
3707 03610   142522          VFD   8,'305,8,'122     ='ER'
3708 03611   0 003556        DAC   LINK
3709         003606      LINK SET   TLNK
3710 03612   0 10 00130 UPPR JST   DOCL
3711 03613   0 001520        DAC   OVER
3712 03614   0 001443        DAC   PLUS
3713 03615   0 001533        DAC   SWAP
3714 03616   0 000372        DAC   XDO
3715 03617   0 000401   UPR1 DAC   I
3716 03620   0 001572        DAC   CAT
3717 03621   0 000241        DAC   LIT
3718 03622   000340          OCT   340              =LOWERCASE-A-1
3719 03623   0 002315        DAC   GRTR
3720 03624   0 000401        DAC   I
3721 03625   0 001572        DAC   CAT
3722 03626   0 000241        DAC   LIT
3723 03627   000373          OCT   373              =LOWERCASE-Z+1
3724 03630   0 002303        DAC   LESS
3725 03631   0 001300        DAC   AND
3726 03632   0 000277        DAC   ZBRA
3727 03633   0 003643        DAC   UPR2
3728 03634   0 000401        DAC   I
3729 03635   0 001572        DAC   CAT
3730 03636   0 000241        DAC   LIT
3731 03637   000040          OCT   40
3732 03640   0 001316        DAC   XOR
3733 03641   0 000401        DAC   I
3734 03642   0 001605        DAC   CSTR
3735 03643   0 000322   UPR2 DAC   XLOP
3736 03644   0 003617        DAC   UPR1
3737 03645   0 001361        DAC   SMIS
3738                     *
3739                     * **** (ABORT) ****
3740                     *    HEAD  FNUL,(ABORT),PABT,DOCL
3741         003646      TLNK SET   *
3742 03646   003650          VFD   1,0,1,FNUL,6,7,8,'250
3743                     *                              =<FNUL,7>,'('
3744 03647   140702          BCI   2,ABOR
     03650   147722
```

```
3745 03651   152051          VFD   8,’324,8,’051     =’T)’
3746 03652   0 003606        DAC   LINK
3747         003646     LINK SET   TLNK
3748 03653   0 10 00130 PABT JST   DOCL
3749 03654   0 004347        DAC   ABRT
3750 03655   0 001361        DAC   SMIS
3751                    *
3752                    * **** ERROR ****
3753                    *    HEAD  FNUL,ERROR,EROR,DOCL
3754         003656     TLNK SET   *
3755 03656   002705          VFD   1,0,1,FNUL,6,5,8,’305
3756                    *                              =<FNUL,5>,’E’
3757 03657   151322          BCI   1,RR
3758 03660   147522          VFD   8,’317,8,’122     =’OR’
3759 03661   0 003646        DAC   LINK
3760         003656     LINK SET   TLNK
3761 03662   0 10 00130 EROR JST   DOCL
3762 03663   0 002026        DAC   WARN
3763 03664   0 001565        DAC   AT
3764 03665   0 001435        DAC   ZLES
3765 03666   0 000277        DAC   ZBRA
3766 03667   0 003671        DAC   ERR1
3767 03670   0 003653        DAC   PABT
3768 03671   0 002240   ERR1 DAC   HERE
3769 03672   0 002735        DAC   CNT
3770 03673   0 002750        DAC   TYPE
3771 03674   0 003031        DAC   PDTQ
3772                    *    STRG  $ ?$
3773 03675   001640          VFD   8,3,8,’240        =3,’ ’
3774 03676   137640          VFD   8,’277,8,’240     =’? ’
3775 03677   0 005027        DAC   MESS
3776 03700   0 001332        DAC   SPST
3777 03701   0 002077        DAC   IN
3778 03702   0 001565        DAC   AT
3779 03703   0 002072        DAC   BLK
3780 03704   0 001565        DAC   AT
3781 03705   0 004313        DAC   QUIT
3782 03706   0 001361        DAC   SMIS
3783                    *
3784                    * **** ID. ****
3785                    *    HEAD  FNUL,ID.,IDDT,DOCL
3786         003707     TLNK SET   *
3787 03707   001711          VFD   1,0,1,FNUL,6,3,8,’311
3788                    *                              =<FNUL,3>,’I’
3789 03710   142056          VFD   8,’304,8,’056     =’D.’
3790 03711   0 003656        DAC   LINK
3791         003707     LINK SET   TLNK
3792 03712   0 10 00130 IDDT JST   DOCL
3793 03713   0 003351        DAC   PAD
3794 03714   0 000241        DAC   LIT
```

```
3795 03715   000020          OCT   20                SINCE WORDS - THATS 32 CHARACTERS
3796 03716   0 000241        DAC   LIT
3797 03717   050521          VFD   8,337,8,337       TWO UNDERSCORES
3798 03720   0 003267        DAC   FILL
3799 03721   0 001541        DAC   DUP
3800 03722   0 002423        DAC   PFA
3801 03723   0 002367        DAC   LFA
3802 03724   0 001520        DAC   OVER
3803 03725   0 002266        DAC   SUB
3804 03726   0 003351        DAC   PAD
3805 03727   0 001533        DAC   SWAP
3806 03730   0 000671        DAC   MOVE
3807 03731   0 003351        DAC   PAD
3808 03732   0 002735        DAC   CNT
3809 03733   0 000241        DAC   LIT
3810 03734   000037          OCT   37
3811 03735   0 001300        DAC   AND
3812 03736   0 002750        DAC   TYPE
3813 03737   0 002335        DAC   SPCE
3814 03740   0 001361        DAC   SMIS
3815                     *
3816                     * **** CREATE ****
3817                     * MODIFIED TO PUT HLT (FOR NOW) AT CFA
3818                     *    HEAD  FNUL,CREATE,CRAT,DOCL
3819         003741      TLNK SET   *
3820 03741   003303          VFD   1,0,1,FNUL,6,6,8,'303
3821                     *                          =<FNUL,6>,'C'
3822 03742   151305          BCI   2,REAT
     03743   140724
3823 03744   042400          VFD   8,'105             ='E'
3824 03745   0 003707        DAC   LINK
3825         003741      LINK SET   TLNK
3826 03746   0 10 00130  CRAT JST   DOCL
3827 03747   0 003562        DAC   DFND
3828 03750   0 000277        DAC   ZBRA
3829 03751   0 003761        DAC   CRT1
3830 03752   0 001526        DAC   DROP
3831 03753   0 002405        DAC   NFA
3832 03754   0 003712        DAC   IDDT
3833 03755   0 000241        DAC   LIT
3834 03756   000004          DEC   4
3835 03757   0 005027        DAC   MESS
3836 03760   0 002335        DAC   SPCE
3837 03761   0 002240    CRT1 DAC   HERE
3838 03762   0 001541        DAC   DUP               ( HERE HERE )
3839 03763   0 001541        DAC   DUP               ( HERE HERE HERE )
3840 03764   0 001616        DAC   BYTE              ( HERE HERE C-HERE )
3841 03765   0 001572        DAC   CAT               ( HERE HERE FIRST-BYTE )
3842 03766   0 002017        DAC   WDTH
3843 03767   0 001565        DAC   AT
```

```
3844 03770   0 004473        DAC   MIN
3845 03771   0 001625        DAC   CELL
3846 03772   0 002223        DAC   ONEP            ( HERE HERE WORDS-REQUD )
3847 03773   0 001541        DAC   DUP             ( HERE HERE WORDS-REQUD WORDS-REQUD )
3848 03774   0 002250        DAC   ALOT            ( HERE HERE WORDS-REQUD )
3849 03775   0 001533        DAC   SWAP            ( HERE WORDS-REQUD HERE )
3850 03776   0 001541        DAC   DUP             ( HERE WORDS-REQUD HERE HERE )
3851 03777   0 000241        DAC   LIT
3852 04000   020000          OCT   020000          SET SMUDGE BIT, LEAVE TOP CLEAR
3853 04001   0 001557        DAC   TOGL            ( HERE WORDS-REQUD HERE )
3854                     *
3855 04002   0 001443        DAC   PLUS            ( HERE LFA )
3856 04003   0 001722        DAC   ONE             ( HERE LFA ONE )
3857 04004   0 002266        DAC   SUB             ( HERE LFA-1 )
3858 04005   0 001541        DAC   DUP             ( HERE LFA-1 LFA-1 )
3859 04006   0 001565        DAC   AT
3860 04007   0 000241        DAC   LIT
3861 04010   177577          OCT   177577          CLEAR TOP BIT
3862 04011   0 001300        DAC   AND
3863 04012   0 001533        DAC   SWAP
3864 04013   0 001577        DAC   STOR            ( HERE )
3865                     *
3866 04014   0 002357        DAC   LTST
3867 04015   0 002256        DAC   COMA
3868 04016   0 002136        DAC   CURR
3869 04017   0 001565        DAC   AT
3870 04020   0 001577        DAC   STOR            LINK FIELD
3871 04021   0 001716        DAC   ZERO            HLT INSTRUCTION
3872 04022   0 002256        DAC   COMA
3873 04023   0 001361        DAC   SMIS
3874                     *
3875                     * **** [COMPILE] ****
3876                     *    HEAD  FIMD,[COMPILE],BCMP,DOCL
3877         004024      TLNK SET   *
3878 04024   044733          VFD   1,0,1,FIMD,6,9,8,'333
3879                     *                          =<FIMD,9>,'['
3880 04025   141717          BCI   3,COMPIL
     04026   146720
     04027   144714
3881 04030   142535          VFD   8,'305,8,'135     ='E]'
3882 04031   0 003741        DAC   LINK
3883         004024      LINK SET   TLNK
3884 04032   0 10 00130  BCMP JST   DOCL
3885 04033   0 003562        DAC   DFND
3886 04034   0 001423        DAC   ZEQU
3887 04035   0 001716        DAC   ZERO
3888 04036   0 002455        DAC   QERR
3889 04037   0 001526        DAC   DROP
3890 04040   0 002376        DAC   CFA
3891 04041   0 002256        DAC   COMA
```

```
3892 04042   0 001361        DAC   SMIS
3893                     *
3894                     * **** LITERAL ****
3895                     *    HEAD  FIMD,LITERAL,LTRL,DOCL
3896         004043      TLNK SET   *
3897 04043   043714          VFD   1,0,1,FIMD,6,7,8,'314
3898                     *                              =<FIMD,7>,'L'
3899 04044   144724          BCI   2,ITER
     04045   142722
3900 04046   140514          VFD   8,'301,8,'114      ='AL'
3901 04047   0 004024        DAC   LINK
3902         004043      LINK SET   TLNK
3903 04050   0 10 00130  LTRL JST   DOCL
3904 04051   0 002144        DAC   STAT
3905 04052   0 001565        DAC   AT
3906 04053   0 000277        DAC   ZBRA
3907 04054   0 004060        DAC   LIT1
3908 04055   0 002570        DAC   COMP
3909 04056   0 000241        DAC   LIT
3910 04057   0 002256        DAC   COMA
3911 04060   0 001361    LIT1 DAC   SMIS
3912                     *
3913                     * **** DLITERAL ****
3914                     *    HEAD  FIMD,DLITERAL,DLIT,DOCL
3915         004061      TLNK SET   *
3916 04061   044304          VFD   1,0,1,FIMD,6,8,8,'304
3917                     *                              =<FIMD,8>,'D'
3918 04062   146311          BCI   3,LITERA
     04063   152305
     04064   151301
3919 04065   046000          VFD   8,'114             ='L'
3920 04066   0 004043        DAC   LINK
3921         004061      LINK SET   TLNK
3922 04067   0 10 00130  DLIT JST   DOCL
3923 04070   0 002144        DAC   STAT
3924 04071   0 001565        DAC   AT
3925 04072   0 000277        DAC   ZBRA
3926 04073   0 004077        DAC   DLT1
3927 04074   0 001533        DAC   SWAP
3928 04075   0 004050        DAC   LTRL
3929 04076   0 004050        DAC   LTRL
3930 04077   0 001361    DLT1 DAC   SMIS
3931                     *
3932                     * **** U< **** UNSIGNED LESS-THAN, NEEDED FOR '?STACK'
3933                     *  : U< >R 0 R> 0 DMINUS D+ SWAP DROP 0< ;
3934                     *    HEAD  FNUL,U<,ULES,DOCL
3935         004100      TLNK SET   *
3936 04100   001325          VFD   1,0,1,FNUL,6,2,8,'325
3937                     *                              =<FNUL,2>,'U'
3938 04101   036000          VFD   8,'074             ='<'
```

```
3939 04102   0 004061        DAC   LINK
3940         004100     LINK SET   TLNK
3941 04103   0 10 00130 ULES JST   DOCL
3942 04104   0 001402        DAC   TOR
3943 04105   0 001716        DAC   ZERO
3944 04106   0 001412        DAC   FRMR
3945 04107   0 001716        DAC   ZERO
3946 04110   0 001502        DAC   DMNS
3947 04111   0 001451        DAC   DPLS
3948 04112   0 001533        DAC   SWAP
3949 04113   0 001526        DAC   DROP
3950 04114   0 001435        DAC   ZLES
3951 04115   0 001361        DAC   SMIS
3952                     *
3953                     * **** ?STACK **** ERROR CHECK.
3954                     *    HEAD  FNUL,?STACK,QSTK,DOCL
3955         004116     TLNK SET   *
3956 04116   003277          VFD   1,0,1,FNUL,6,6,8,'277
3957                     *                              =<FNUL,6>,'?'
3958 04117   151724          BCI   2,STAC
     04120   140703
3959 04121   045400          VFD   8,'113               ='K'
3960 04122   0 004100        DAC   LINK
3961         004116     LINK SET   TLNK
3962 04123   0 10 00130 QSTK JST   DOCL
3963 04124   0 001777        DAC   SZRO
3964 04125   0 001565        DAC   AT
3965 04126   0 001324        DAC   SPAT
3966 04127   0 002223        DAC   ONEP
3967 04130   0 004103        DAC   ULES
3968 04131   0 001722        DAC   ONE
3969 04132   0 002455        DAC   QERR
3970 04133   0 001324        DAC   SPAT
3971 04134   0 002240        DAC   HERE
3972 04135   0 000241        DAC   LIT
3973 04136   000100          OCT   100
3974 04137   0 001443        DAC   PLUS
3975 04140   0 004103        DAC   ULES
3976 04141   0 001726        DAC   TWO
3977 04142   0 002455        DAC   QERR
3978 04143   0 001361        DAC   SMIS
3979                     *
3980                     * **** INTERPRET ****
3981                     *    HEAD  FNUL,INTERPRET,ITRP,DOCL
3982         004144     TLNK SET   *
3983 04144   004711          VFD   1,0,1,FNUL,6,9,8,'311
3984                     *                              =<FNUL,9>,'I'
3985 04145   147324          BCI   3,NTERPR
     04146   142722
     04147   150322
```

```
3986 04150   142524          VFD   8,'305,8,'124     ='ET'
3987 04151   0 004116        DAC   LINK
3988         004144     LINK SET   TLNK
3989 04152   0 10 00130 ITRP JST   DOCL
3990 04153   0 003562   ITR1 DAC   DFND              FIND NEXT WORD
3991 04154   0 000277        DAC   ZBRA              FOUND?
3992 04155   0 004174        DAC   ITR4              NO
3993 04156   0 002144        DAC   STAT              COMPARE STATE TO LENGTH BYTE
3994 04157   0 001565        DAC   AT
3995 04160   0 002303        DAC   LESS
3996 04161   0 000277        DAC   ZBRA
3997 04162   0 004167        DAC   ITR2
3998 04163   0 002376        DAC   CFA
3999 04164   0 002256        DAC   COMA
4000 04165   0 000263        DAC   BRAN
4001 04166   0 004171        DAC   ITR3
4002 04167   0 002376   ITR2 DAC   CFA
4003 04170   0 000254        DAC   EXEC
4004 04171   0 004123   ITR3 DAC   QSTK
4005 04172   0 000263        DAC   BRAN
4006 04173   0 004212        DAC   ITR7
4007 04174   0 002240   ITR4 DAC   HERE
4008 04175   0 001616        DAC   BYTE
4009 04176   0 003510        DAC   NUMB
4010 04177   0 002157        DAC   DPL
4011 04200   0 001565        DAC   AT
4012 04201   0 002223        DAC   ONEP
4013 04202   0 000277        DAC   ZBRA
4014 04203   0 004207        DAC   ITR5
4015 04204   0 004067        DAC   DLIT
4016 04205   0 000263        DAC   BRAN
4017 04206   0 004211        DAC   ITR6
4018 04207   0 001526   ITR5 DAC   DROP
4019 04210   0 004050        DAC   LTRL
4020 04211   0 004123   ITR6 DAC   QSTK
4021 04212   0 000263   ITR7 DAC   BRAN
4022 04213   0 004153        DAC   ITR1
4023                     *
4024                     * **** IMMEDIATE ****
4025                     *    HEAD  FNUL,IMMEDIATE,IMMD,DOCL
4026         004214     TLNK SET   *
4027 04214   004711          VFD   1,0,1,FNUL,6,9,8,'311
4028                     *                           =<FNUL,9>,'I'
4029 04215   146715          BCI   3,MMEDIA
     04216   142704
     04217   144701
4030 04220   152105          VFD   8,'324,8,'105     ='TE'
4031 04221   0 004144        DAC   LINK
4032         004214     LINK SET   TLNK
4033 04222   0 10 00130 IMMD JST   DOCL
```

```
4034 04223   0 002357        DAC   LTST
4035 04224   0 000241        DAC   LIT
4036 04225   040000          OCT   040000
4037 04226   0 001557        DAC   TOGL
4038 04227   0 001361        DAC   SMIS
4039                 *
4040                 * **** VOCABULARY ****
4041                 *    HEAD  FNUL,VOCABULARY,VCAB,DOCL
4042         004230     TLNK SET   *
4043 04230   005326          VFD   1,0,1,FNUL,6,10,8,'326
4044                 *                              =<FNUL,10>,'V'
4045 04231   147703          BCI   4,OCABULAR
     04232   140702
     04233   152714
     04234   140722
4046 04235   054400          VFD   8,'131           ='Y'
4047 04236   0 004214        DAC   LINK
4048         004230     LINK SET   TLNK
4049 04237   0 10 00130 VCAB JST   DOCL
4050 04240   0 002712        DAC   BULD
4051 04241   0 000241        DAC   LIT
4052 04242   120201          OCT   120201
4053 04243   0 002256        DAC   COMA
4054 04244   0 002136        DAC   CURR
4055 04245   0 001565        DAC   AT
4056 04246   0 002376        DAC   CFA
4057 04247   0 002256        DAC   COMA
4058 04250   0 002240        DAC   HERE
4059 04251   0 002051        DAC   VOCL
4060 04252   0 001565        DAC   AT
4061 04253   0 002256        DAC   COMA
4062 04254   0 002051        DAC   VOCL
4063 04255   0 001577        DAC   STOR
4064 04256   0 002722        DAC   DOES
4065 04257   0 002231   DOVC DAC   TWOP
4066 04260   0 002127        DAC   CONT
4067 04261   0 001577        DAC   STOR
4068 04262   0 001361        DAC   SMIS
4069                 *
4070                 *  ***** THE DEFINITION OF 'FORTH' WAS MOVED TO NEAR THE END OF THE
4071                 *  DICTIONARY, BECAUSE IT IS NOT PURE CODE.
4072                 *
4073                 *
4074                 * **** DEFINITIONS ****
4075                 *    HEAD  FNUL,DEFINITIONS,DFNS,DOCL
4076         004263     TLNK SET   *
4077 04263   005704          VFD   1,0,1,FNUL,6,11,8,'304
4078                 *                              =<FNUL,11>,'D'
4079 04264   142706          BCI   4,EFINITIO
     04265   144716
```

```
     04266   144724
     04267   144717
4080 04270   147123          VFD   8,'316,8,'123     ='NS'
4081 04271   0 004230        DAC   LINK
4082         004263     LINK SET   TLNK
4083 04272   0 10 00130 DFNS JST   DOCL
4084 04273   0 002127        DAC   CONT
4085 04274   0 001565        DAC   AT
4086 04275   0 002136        DAC   CURR
4087 04276   0 001577        DAC   STOR
4088 04277   0 001361        DAC   SMIS
4089                    *
4090                    * **** ( ****
4091                    *    HEAD  FIMD,(,PARN,DOCL
4092         004300     TLNK SET   *
4093 04300   040450          VFD   1,0,1,FIMD,6,1,8,'050
4094                    *                              =<FIMD,1>,'('
4095 04301   0 004263        DAC   LINK
4096         004300     LINK SET   TLNK
4097 04302   0 10 00130 PARN JST   DOCL
4098 04303   0 000241        DAC   LIT
4099 04304   000251          VFD   16,CRPR           =')'
4100 04305   0 003363        DAC   WORD
4101 04306   0 001361        DAC   SMIS
4102                    *
4103                    * **** QUIT ****
4104                    *    HEAD  FNUL,QUIT,QUIT,DOCL
4105         004307     TLNK SET   *
4106 04307   002321          VFD   1,0,1,FNUL,6,4,8,'321
4107                    *                              =<FNUL,4>,'Q'
4108 04310   152711          BCI   1,UI
4109 04311   052000          VFD   8,'124            ='T'
4110 04312   0 004300        DAC   LINK
4111         004307     LINK SET   TLNK
4112 04313   0 10 00130 QUIT JST   DOCL
4113 04314   0 001716        DAC   ZERO
4114 04315   0 002072        DAC   BLK
4115 04316   0 001577        DAC   STOR
4116 04317   0 002603        DAC   LBRC
4117 04320   0 001345   QUT1 DAC   RPST
4118 04321   0 000627        DAC   CR
4119 04322   0 003211   QUT2 DAC   QURY
4120 04323   0 004152        DAC   ITRP
4121                         IFN   PTW
4122 04324   0 002072        DAC   BLK
4123 04325   0 001565        DAC   AT
4124 04326   0 002223        DAC   ONEP              PAPERTAPE?
4125 04327   0 000277        DAC   ZBRA              YES - LOOP
4126 04330   0 004322        DAC   QUT2
4127                         ENDC
```

```
4128 04331   0 002144        DAC   STAT
4129 04332   0 001565        DAC   AT
4130 04333   0 001423        DAC   ZEQU
4131 04334   0 000277        DAC   ZBRA
4132 04335   0 004341        DAC   QUT3
4133 04336   0 003031        DAC   PDTQ
4134                     *    STRG  $ OK
4135 04337   001640          VFD   8,3,8,'240        =3,' '
4136 04340   147713          VFD   8,'317,8,'313     ='OK'
4137 04341   0 000263   QUT3 DAC   BRAN
4138 04342   0 004320        DAC   QUT1
4139                     *
4140                     * **** ABORT ****
4141                     *    HEAD  FNUL,ABORT,ABRT,DOCL
4142         004343     TLNK SET   *
4143 04343   002701          VFD   1,0,1,FNUL,6,5,8,'301
4144                     *                            =<FNUL,5>,'A'
4145 04344   141317          BCI   1,BO
4146 04345   151124          VFD   8,'322,8,'124     ='RT'
4147 04346   0 004307        DAC   LINK
4148         004343     LINK SET   TLNK
4149 04347   0 10 00130 ABRT JST   DOCL
4150 04350   0 001332        DAC   SPST
4151 04351   0 002651        DAC   DEC
4152 04352   0 002335        DAC   SPCE
4153 04353   0 000627        DAC   CR
4154 04354   0 003031        DAC   PDTQ
4155                     *    STRG  FIG-FORTH  V 0.9$
4156 04355   010706          VFD   8,17,8,'306       =17,'F'
4157 04356   144707          BCI   7,IG-FORTH  V 0.
     04357   126706
     04360   147722
     04361   152310
     04362   120240
     04363   153240
     04364   130256
4158 04365   134640          VFD   8,'271,8,'240     ='9 '
4159                         IFN   HSA
4160 04366   0 003031        DAC   PDTQ
4161                     *    STRG  HSA$
4162 04367   002310          VFD   8,4,8,'310        =4,'H'
4163 04370   151701          BCI   1,SA
4164 04371   120000          VFD   8,'240            =' '
4165                         ENDC
4166                         IFN   XTND
4167                         DAC   PDTQ
4168                     *    STRG  EXA$
4169                         VFD   8,4,8,'305        =4,'E'
4170                         BCI   1,XA
4171                         VFD   8,'240            =' '
```

```
4172                        ENDC
4173                        IFZ   DISK
4174 04372   0 003031       DAC   PDTQ
4175                    *   STRG  NO-
4176 04373   001716         VFD   8,3,8,'316        =3,'N'
4177 04374   147655         VFD   8,'317,8,'255     ='O-'
4178                        ENDC
4179 04375   0 003031       DAC   PDTQ
4180                    *   STRG  DISK$
4181 04376   002704         VFD   8,5,8,'304        =5,'D'
4182 04377   144723         BCI   1,IS
4183 04400   145640         VFD   8,'313,8,'240     ='K '
4184                    *
4185 04401   0 001777       DAC   SZRO
4186 04402   0 001565       DAC   AT
4187 04403   0 002240       DAC   HERE
4188 04404   0 002266       DAC   SUB
4189 04405   0 000241       DAC   LIT
4190 04406   000100         OCT   100
4191 04407   0 002266       DAC   SUB
4192 04410   0 005554       DAC   UDOT
4193 04411   0 003031       DAC   PDTQ
4194                    *   STRG  WORDS FREE
4195 04412   005327         VFD   8,10,8,'327       =10,'W'
4196 04413   147722         BCI   4,ORDS FRE
     04414   142323
     04415   120306
     04416   151305
4197 04417   142400         VFD   8,'305            ='E'
4198                    *
4199 04420   0 005652       DAC   FRTH
4200 04421   0 004272       DAC   DFNS
4201 04422   0 004313       DAC   QUIT
4202                    *
4203                    *  COLD START
4204                    *
4205                    * THE ACTUAL CODE IS DOWN NEAR ORGN AT '1000
4206                    * CLOSE TO THE ENTRY POINT
4207                    *
4208                    * **** COLD ****
4209                    *   HEAD  FNUL,COLD,COLD
4210         004423     TLNK SET  *
4211 04423   002303         VFD   1,0,1,FNUL,6,4,8,'303
4212                    *                          =<FNUL,4>,'C'
4213 04424   147714         BCI   1,OL
4214 04425   042000         VFD   8,'104            ='D'
4215 04426   0 004343       DAC   LINK
4216         004423     LINK SET  TLNK
4217         004427     COLD EQU  *
4218 04427   0 01 01025     JMP   CENT
```

```
4219                    *
4220                    * **** S->D ****
4221                    *    HEAD  FNUL,S->D,STOD
4222          004430    TLNK SET   *
4223 04430   002323          VFD   1,0,1,FNUL,6,4,8,'323
4224                    *                              =<FNUL,4>,'S'
4225 04431   126676          BCI   1,->
4226 04432   042000          VFD   8,'104             ='D'
4227 04433   0 004423        DAC   LINK
4228          004430    LINK SET   TLNK
4229          004434    STOD EQU   *
4230 04434   1 02 00001       LDA   1,1
4231 04435   140320          CSA                      COPY SIGN TO CARRY
4232 04436   140040          CRA
4233 04437   101001          SSC                      NEGATIVE?
4234 04440   0 01 00116       JMP   PUSH              NO - PUSH ZERO
4235 04441   0 02 00715       LDA   =-1
4236 04442   0 01 00116       JMP   PUSH              YES - SIGN EXTEND
4237                    *
4238                    *  NOTE - THIS SYSTEM DOESN'T NEED THE OPERATIONS '+-' AND 'D+-',
4239                    *    BECAUSE 'M*' AND 'M/' ARE DEFINED IN CODE.
4240                    *
4241                    * **** ABS ****
4242                    *    HEAD  FNUL,ABS,ABS,DOCL
4243          004443    TLNK SET   *
4244 04443   001701          VFD   1,0,1,FNUL,6,3,8,'301
4245                    *                              =<FNUL,3>,'A'
4246 04444   141123          VFD   8,'302,8,'123      ='BS'
4247 04445   0 004430        DAC   LINK
4248          004443    LINK SET   TLNK
4249 04446   0 10 00130 ABS  JST   DOCL
4250 04447   0 001541        DAC   DUP
4251 04450   0 001435        DAC   ZLES
4252 04451   0 000277        DAC   ZBRA
4253 04452   0 004454        DAC   ABS1
4254 04453   0 001472        DAC   MINS
4255 04454   0 001361   ABS1 DAC   SMIS
4256                    *
4257                    * **** DABS ****
4258                    *    HEAD  FNUL,DABS,DABS,DOCL
4259          004455    TLNK SET   *
4260 04455   002304          VFD   1,0,1,FNUL,6,4,8,'304
4261                    *                              =<FNUL,4>,'D'
4262 04456   140702          BCI   1,AB
4263 04457   051400          VFD   8,'123             ='S'
4264 04460   0 004443        DAC   LINK
4265          004455    LINK SET   TLNK
4266 04461   0 10 00130 DABS JST   DOCL
4267 04462   0 001541        DAC   DUP
4268 04463   0 001435        DAC   ZLES
```

```
4269 04464   0 000277        DAC   ZBRA
4270 04465   0 004467        DAC   DAB1
4271 04466   0 001502        DAC   DMNS
4272 04467   0 001361   DAB1 DAC   SMIS
4273                    *
4274                    * **** MIN ****
4275                    *    HEAD  FNUL,MIN,MIN,DOCL
4276         004470     TLNK SET   *
4277 04470   001715          VFD   1,0,1,FNUL,6,3,8,'315
4278                    *                              =<FNUL,3>,'M'
4279 04471   144516          VFD   8,'311,8,'116      ='IN'
4280 04472   0 004455        DAC   LINK
4281         004470     LINK SET   TLNK
4282 04473   0 10 00130 MIN  JST   DOCL
4283 04474   0 001520        DAC   OVER
4284 04475   0 001520        DAC   OVER
4285 04476   0 002315        DAC   GRTR
4286 04477   0 000277        DAC   ZBRA
4287 04500   0 004502        DAC   MIN1
4288 04501   0 001533        DAC   SWAP
4289 04502   0 001526   MIN1 DAC   DROP
4290 04503   0 001361        DAC   SMIS
4291                    *
4292                    * **** MAX ****
4293                    *    HEAD  FNUL,MAX,MAX,DOCL
4294         004504     TLNK SET   *
4295 04504   001715          VFD   1,0,1,FNUL,6,3,8,'315
4296                    *                              =<FNUL,3>,'M'
4297 04505   140530          VFD   8,'301,8,'130      ='AX'
4298 04506   0 004470        DAC   LINK
4299         004504     LINK SET   TLNK
4300 04507   0 10 00130 MAX  JST   DOCL
4301 04510   0 001520        DAC   OVER
4302 04511   0 001520        DAC   OVER
4303 04512   0 002303        DAC   LESS
4304 04513   0 000277        DAC   ZBRA
4305 04514   0 004516        DAC   MAX1
4306 04515   0 001533        DAC   SWAP
4307 04516   0 001526   MAX1 DAC   DROP
4308 04517   0 001361        DAC   SMIS
4309                    *
4310                    * **** M* ****
4311                    *    HEAD  FNUL,M*,MSTR
4312         004520     TLNK SET   *
4313 04520   001315          VFD   1,0,1,FNUL,6,2,8,'315
4314                    *                              =<FNUL,2>,'M'
4315 04521   025000          VFD   8,'052             ='*'
4316 04522   0 004504        DAC   LINK
4317         004520     LINK SET   TLNK
4318         004523     MSTR EQU   *
```

```
4319                        IFZ   HSA
4320                    * SOFTWARE SIGNED MULTIPLY
4321                        LDA   =-1
4322                        STA   T3              ASSUME -VE MULTIPLIER, SIGN EXTENSION
4323                        CRA
4324                        IAB                   CLEAR B
4325                        LDA   2,1             MULTIPLICAND
4326                        LGL   1               TOP BIT TO CARRY
4327                        STA   2,1             MULTIPLICAND
4328                        LDA   =-15            COUNTER
4329                        STA   T1
4330                        SSC                   WAS TOP BIT SET?
4331                        JMP   MST2            NO - DON'T SUBTRACT MUTIPLIER
4332                    * YES - THIS BIT WORTH -2^15
4333                        LDA   1,1             GET MULTIPLIER
4334                        CAS   ='100000        IS IT MIN INT. VALUE?
4335                        SKP
4336                        JMP   MST1            SO NEGATED IS +2^15
4337                        TCA                   NEGATE
4338                        SMI                   NEGATIVE RESULT?
4339                        JMP   MST1            NO, B = 0 ALREADY
4340                    * NEGATED MULTIPLIER IS -VE SO ORIGINAL IS +VE
4341                        IAB                   NO - PRESERVE A
4342                        CRA
4343                        STA   T3              SIGN EXTENSION
4344                        LDA   =-1             SET B TO -1
4345                        IAB
4346                    MST1 IAB                  SWAP ACCUM. WORDS TO RIGHT PLACE
4347                        JMP   UST1            JUMP INTO UNSIGNED ROUTINE FOR 15 BITS
4348                    *
4349                    * TOP BIT OF MULTIPLICAND CLEAR, SO ACCUMULATOR
4350                    * STARTS AT ZERO, BUT NEED TO SORT SIGN EXTEND
4351                    * OF MULTIPLIER
4352                    MST2 LDA   1,1            GET MULTIPLIER
4353                        CSA                   SIGN TO CARRY
4354                        CRA
4355                        SSC                   POSITIVE?
4356                        STA   T3              YES - CLEAR SIGN EXTENSION
4357                        JMP   MST1            OFF TO UNSIGNED ROUTINE
4358                        EJCT
4359                        ELSE
4360                    * HARDWARE SIGNED MULTIPLY
4361 04523   1 02 00002     LDA   2,1
4362 04524   1 16 00001     MPY   1,1             RESULT IN 16-0-15 FORMAT
4363                    * IT WOULD REALY HAVE HELPED IF OVERFLOW
4364                    * WENT TO C, BUT IT DOESN'T
4365 04525   0 11 00713     CAS   ='100000        OVERFLOW?
4366 04526   100000         SKP                   NO
4367 04527   0 01 04544     JMP   MST1            YES
4368 04530   000201         IAB
```

```
4369 04531   1 04 00002      STA   2,1              SAVE LOWER 15 BITS
4370 04532   140040          CRA                    CLEAR B
4371 04533   000201          IAB
4372 04534   0401 77         LRS   1                FORM CORRECT MS WORD
4373 04535   1 04 00001      STA   1,1
4374 04536   000201          IAB                    GET BACK 2^15 BIT FROM B2
4375 04537   0414 77         LGL   1                PUT INTO A1
4376 04540   1 05 00002      ERA   2,1              OR IN LOWER 15 BITS
4377 04541   1 04 00002      STA   2,1
4378                     *
4379                     *    NEXT
4380 04542   0 12 00100      IRS   IP
4381 04543  -0 01 00100      JMP*  IP
4382                     * OVERFLOW
4383                     * -2^15 * -2^15 => -2^30, SHOULD BE +2^30
4384 04544   0 02 00717 MST1 LDA   ='040000
4385 04545   1 04 00001      STA   1,1
4386 04546   140040          CRA
4387 04547   1 04 00002      STA   2,1
4388                     *
4389                     *    NEXT
4390 04550   0 12 00100      IRS   IP
4391 04551  -0 01 00100      JMP*  IP
4392                          ENDC
4393                          EJCT
```

```
4394                    *
4395                    * **** M/ ****
4396                    *    HEAD  FNUL,M/,MSLA
4397          004552    TLNK SET   *
4398 04552   001315          VFD   1,0,1,FNUL,6,2,8,'315
4399                    *                              =<FNUL,2>,'M'
4400 04553   027400          VFD   8,'057               ='/'
4401 04554   0 004520        DAC   LINK
4402          004552    LINK SET   TLNK
4403          004555    MSLA EQU   *
4404                          IFZ   HSA
4405                          JST   SSDV
4406                    *
4407                    *    NEXT
4408                          IRS   IP
4409                          JMP*  IP
4410                          ELSE
4411 04555   1 02 00002       LDA   2,1                 MS WORD OF DIVIDEND
4412 04556   0415 77          ALS   1                   SHIFT UP ONE BIT
4413 04557   100001           SRC                       OVERFLOW?
4414 04560   0 01 04604       JMP   DVS2                YES, DO SOFTWARE DIVIDE
4415 04561   000201           IAB
4416 04562   1 02 00003       LDA   3,1                 LS WORD OF DIVIDEND
4417 04563   140320           CSA                       TOP BIT TO CARRY
4418 04564   000201           IAB
4419 04565   141216           ACA                       2^15 BIT INTO A, LSB
4420                    *
4421                    * AT THIS POINT KNOW THAT WE HAVE A
4422                    * A VALID 31-BIT DIVIDEND, SO CAN USE H/W INSTRUCTION
4423 04566   1 17 00001       DIV   1,1
4424 04567   0 12 00000       IRS   0                   DISCARD DIVISOR
4425 04570   100001           SRC                       OVERFLOW?
4426 04571   0 01 04577       JMP   DVS1
4427 04572   1 04 00001       STA   1,1                 QUOTIENT
4428 04573   000201           IAB
4429 04574   1 04 00002       STA   2,1                 REMAINDER
4430                    *
4431                    *    NEXT
4432 04575    0 12 00100      IRS   IP
4433 04576   -0 01 00100      JMP*  IP
4434 04577   140040     DVS1  CRA
4435 04600   1 04 00001       STA   1,1                 QUOTIENT
4436 04601   1 04 00002       STA   2,1                 REMAINDER
4437                    *
4438                    *    NEXT
4439 04602    0 12 00100      IRS   IP
4440 04603   -0 01 00100      JMP*  IP
4441                    *
4442 04604    0 10 04607 DVS2  JST   SSDV
4443                    *
```

```
4444                    *    NEXT
4445 04605   0 12 00100      IRS   IP
4446 04606  -0 01 00100      JMP*  IP
4447                         EJCT
```

```
4448                        ENDC
4449                    * SOFTWARE SIGNED DIVIDE
4450 04607   0 000000   SSDV DAC   **
4451 04610   1 02 00002       LDA   2,1              MS WORD OF DIVIDEND
4452 04611   0 04 00107       STA   T4
4453 04612   101400           SMI                    NEGATIVE?
4454 04613   0 01 04627       JMP   SSD1
4455 04614   1 02 00003       LDA   3,1              LS WORD OF DIVIDEND
4456 04615   140407           TCA                    NEGATE
4457 04616   1 04 00003       STA   3,1
4458 04617   140200           RCB
4459 04620   0 05 00713       ERA   =’100000         MIN. INT VALUE?
4460 04621   101040           SNZ                    NO
4461 04622   140600           SCB                    YES - SET CARRY
4462 04623   1 02 00002       LDA   2,1              MS WORD OF DIVIDEND
4463 04624   0 05 00715       ERA   =-1              COMPLEMENT
4464 04625   141216           ACA                    ADD ANY CARRY IN
4465 04626   1 04 00002       STA   2,1
4466 04627   1 02 00001 SSD1  LDA   1,1              DIVISOR
4467 04630   0 04 00110       STA   T5
4468 04631   100400           SPL                    NEGATIVE?
4469 04632   140407           TCA                    YES - NEGATE
4470 04633   1 04 00001       STA   1,1
4471 04634   0 10 01210       JST   DIVU             UNSIGNED DIVIDE
4472 04635   0 02 00107       LDA   T4               ORIGINAL SIGN OF DIVIDEND
4473 04636   101400           SMI
4474 04637   0 01 04643       JMP   SSD2
4475 04640   1 02 00002       LDA   2,1              REMAINDER
4476 04641   140407           TCA
4477 04642   1 04 00002       STA   2,1
4478 04643   0 02 00107 SSD2  LDA   T4               ORIGINAL SIGN OF DIVIDEND
4479 04644   0 05 00110       ERA   T5               ORIGINAL SIGN OF DIVISOR
4480 04645   101400           SMI
4481 04646  -0 01 04607       JMP*  SSDV
4482 04647   1 02 00001       LDA   1,1              QUOTIENT
4483 04650   140407           TCA
4484 04651   1 04 00001       STA   1,1
4485 04652  -0 01 04607       JMP*  SSDV
4486                        EJCT
```

```
4487                    *
4488                    * **** * ****
4489                    *    HEAD  FNUL,*,STAR,DOCL
4490          004653    TLNK SET   *
4491 04653   000452          VFD   1,0,1,FNUL,6,1,8,'052
4492                    *                              =<FNUL,1>,'*'
4493 04654   0 004552        DAC   LINK
4494          004653    LINK SET   TLNK
4495 04655   0 10 00130 STAR JST   DOCL
4496 04656   0 004523        DAC   MSTR
4497 04657   0 001526        DAC   DROP
4498 04660   0 001361        DAC   SMIS
4499                    *
4500                    * **** /MOD ****
4501                    *    HEAD  FNUL,/MOD,SLMD,DOCL
4502          004661    TLNK SET   *
4503 04661   002257          VFD   1,0,1,FNUL,6,4,8,'257
4504                    *                              =<FNUL,4>,'/'
4505 04662   146717          BCI   1,MO
4506 04663   042000          VFD   8,'104             ='D'
4507 04664   0 004653        DAC   LINK
4508          004661    LINK SET   TLNK
4509 04665   0 10 00130 SLMD JST   DOCL
4510 04666   0 001402        DAC   TOR
4511 04667   0 004434        DAC   STOD
4512 04670   0 001412        DAC   FRMR
4513 04671   0 004555        DAC   MSLA
4514 04672   0 001361        DAC   SMIS
4515                    *
4516                    * **** / ****
4517                    *    HEAD  FNUL,/,SLSH,DOCL
4518          004673    TLNK SET   *
4519 04673   000457          VFD   1,0,1,FNUL,6,1,8,'057
4520                    *                              =<FNUL,1>,'/'
4521 04674   0 004661        DAC   LINK
4522          004673    LINK SET   TLNK
4523 04675   0 10 00130 SLSH JST   DOCL
4524 04676   0 004665        DAC   SLMD
4525 04677   0 001533        DAC   SWAP
4526 04700   0 001526        DAC   DROP
4527 04701   0 001361        DAC   SMIS
4528                    *
4529                    * **** MOD ****
4530                    *    HEAD  FNUL,MOD,MOD,DOCL
4531          004702    TLNK SET   *
4532 04702   001715          VFD   1,0,1,FNUL,6,3,8,'315
4533                    *                              =<FNUL,3>,'M'
4534 04703   147504          VFD   8,'317,8,'104      ='OD'
4535 04704   0 004673        DAC   LINK
4536          004702    LINK SET   TLNK
```

```
4537 04705   0 10 00130 MOD  JST   DOCL
4538 04706   0 004665        DAC   SLMD
4539 04707   0 001526        DAC   DROP
4540 04710   0 001361        DAC   SMIS
4541                      *
4542                      * **** */MOD ****
4543                      *    HEAD  FNUL,*/MOD,SSMD,DOCL
4544         004711       TLNK SET   *
4545 04711   002652            VFD   1,0,1,FNUL,6,5,8,'252
4546                      *                              =<FNUL,5>,'*'
4547 04712   127715           BCI   1,/M
4548 04713   147504           VFD   8,'317,8,'104     ='OD'
4549 04714   0 004702        DAC   LINK
4550         004711       LINK SET   TLNK
4551 04715   0 10 00130 SSMD JST   DOCL
4552 04716   0 001402        DAC   TOR
4553 04717   0 004523        DAC   MSTR
4554 04720   0 001412        DAC   FRMR
4555 04721   0 004555        DAC   MSLA
4556 04722   0 001361        DAC   SMIS
4557                      *
4558                      * **** */ ****
4559                      *    HEAD  FNUL,*/,SSLA,DOCL
4560         004723       TLNK SET   *
4561 04723   001252            VFD   1,0,1,FNUL,6,2,8,'252
4562                      *                              =<FNUL,2>,'*'
4563 04724   027400           VFD   8,'057            ='/'
4564 04725   0 004711        DAC   LINK
4565         004723       LINK SET   TLNK
4566 04726   0 10 00130 SSLA JST   DOCL
4567 04727   0 004715        DAC   SSMD
4568 04730   0 001533        DAC   SWAP
4569 04731   0 001526        DAC   DROP
4570 04732   0 001361        DAC   SMIS
4571                      *
4572                      * **** M/MOD ****
4573                      *    HEAD  FNUL,M/MOD,MSMD,DOCL
4574         004733       TLNK SET   *
4575 04733   002715            VFD   1,0,1,FNUL,6,5,8,'315
4576                      *                              =<FNUL,5>,'M'
4577 04734   127715           BCI   1,/M
4578 04735   147504           VFD   8,'317,8,'104     ='OD'
4579 04736   0 004723        DAC   LINK
4580         004733       LINK SET   TLNK
4581 04737   0 10 00130 MSMD JST   DOCL
4582 04740   0 001402        DAC   TOR
4583 04741   0 001716        DAC   ZERO
4584 04742   0 001416        DAC   R
4585 04743   0 001205        DAC   USLA
4586 04744   0 001412        DAC   FRMR
```

```
4587 04745   0 001533        DAC   SWAP
4588 04746   0 001402        DAC   TOR
4589 04747   0 001205        DAC   USLA
4590 04750   0 001412        DAC   FRMR
4591 04751   0 001361        DAC   SMIS
4592                         EJCT
```

```
4593                        IFN   PTW
4594                    ****************************************************************
4595                    *
4596                    *       PAPERTAPE I/O
4597                    *
4598                    ****************************************************************
4599                    *
4600                    * **** PTR ****
4601                    * TAKE SUBSEQUENT INPUT FROM PAPERTAPE READER
4602                    *    HEAD  FNUL,PTR,PTR,DOCL
4603           004752     TLNK SET   *
4604 04752 001720             VFD   1,0,1,FNUL,6,3,8,'320
4605                    *                              =<FNUL,3>,'P'
4606 04753 152122             VFD   8,'324,8,'122     ='TR'
4607 04754 0 004733           DAC   LINK
4608           004752     LINK SET   TLNK
4609 04755 0 10 00130 PTR  JST   DOCL
4610 04756 0 000241           DAC   LIT
4611 04757 177777             DEC   -1
4612 04760 0 002072           DAC   BLK
4613 04761 0 001577           DAC   STOR
4614 04762 0 001361           DAC   SMIS
4615                    *
4616                    * **** PTRK ****
4617                    * GET A CHARACTER FROM PAPERTAPE
4618                    *    HEAD  FNUL,PTRK,PTRK,DOCL
4619           004763     TLNK SET   *
4620 04763 002320             VFD   1,0,1,FNUL,6,4,8,'320
4621                    *                              =<FNUL,4>,'P'
4622 04764 152322             BCI   1,TR
4623 04765 045400             VFD   8,'113            ='K'
4624 04766 0 004752           DAC   LINK
4625           004763     LINK SET   TLNK
4626 04767 0 10 00130 PTRK JST   DOCL
4627 04770 0 000634   PTK1 DAC   PTRC
4628 04771 0 001541           DAC   DUP
4629 04772 0 000277           DAC   ZBRA              DISCARD NULLS
4630 04773 0 005002           DAC   PTK2
4631 04774 0 001541           DAC   DUP
4632 04775 0 000241           DAC   LIT
4633 04776 000212             VFD   16,CLF
4634 04777 0 002273           DAC   EQAL              LINE FEED?
4635 05000 0 000277           DAC   ZBRA              NO - RETAIN
4636 05001 0 005005           DAC   PTK3              ELSE FALL THROUGH AND DISCARD
4637                    *
4638 05002 0 001526   PTK2 DAC   DROP
4639 05003 0 000263           DAC   BRAN
4640 05004 0 004770           DAC   PTK1
4641                    *
4642 05005 0 001541   PTK3 DAC   DUP
```

```
4643 05006   0 000241        DAC   LIT
4644 05007   000204          VFD   16,CEOT          END OF TAPE?
4645 05010   0 002273        DAC   EQAL
4646 05011   0 000277        DAC   ZBRA             NO
4647 05012   0 005021        DAC   PTK4
4648                    *
4649 05013   0 001526        DAC   DROP
4650 05014   0 001716        DAC   ZERO
4651 05015   0 002072        DAC   BLK
4652 05016   0 001577        DAC   STOR
4653 05017   0 000241        DAC   LIT
4654 05020   000215          VFD   16,CCR           REPLACE EOT WITH CR
4655                    *
4656 05021   0 001361   PTK4 DAC   SMIS
4657                         ENDC
4658                         EJCT
```

```
4659                     IFN   DISK
4660                 ****************************************************************
4661                 *
4662                 *       DISK I/O  (SECTION COMMON TO ALL OPERATING SYSTEMS)
4663                 *       NOTE THAT EACH OPERATING SYSTEM DEFINED 'R/W' - READ
4664                 *       OR WRITE A 1024-BYTE RANDOM-ACCESS BLOCK.
4665                 *
4666                 ****************************************************************
4667                 *
4668                 * **** +BUF ****
4669                 *    HEAD  FNUL,+BUF,PBUF,DOCL
4670                 TLNK SET   *
4671                      VFD   1,0,1,FNUL,6,4,8,'253
4672                 *                            =<FNUL,4>,'+'
4673                      BCI   1,BU
4674                      VFD   8,'106           ='F'
4675                      DAC   LINK
4676                 LINK SET   TLNK
4677                 PBUF JST   DOCL
4678                      DAC   BBUF
4679                      DAC   LIT
4680                      OCT   4
4681                      DAC   PLUS
4682                      DAC   PLUS
4683                      DAC   DUP
4684                      DAC   LIMT
4685                      DAC   AT
4686                      DAC   EQAL
4687                      DAC   ZBRA
4688                      DAC   PBF1
4689                      DAC   DROP
4690                      DAC   FRST
4691                      DAC   AT
4692                 PBF1 DAC   DUP
4693                      DAC   PREV
4694                      DAC   AT
4695                      DAC   SUB
4696                      DAC   SMIS
4697                 *
4698                 * **** UPDATE ****
4699                 *    HEAD  FNUL,UPDATE,UPDT,DOCL
4700                 TLNK SET   *
4701                      VFD   1,0,1,FNUL,6,6,8,'325
4702                 *                            =<FNUL,6>,'U'
4703                      BCI   2,PDAT
4704                      VFD   8,'105           ='E'
4705                      DAC   LINK
4706                 LINK SET   TLNK
4707                 UPDT JST   DOCL
4708                      DAC   PREV
```

```
4709                      DAC   AT
4710                      DAC   AT
4711                      DAC   LIT
4712                      OCT   100000
4713                      DAC   OR
4714                      DAC   PREV
4715                      DAC   AT
4716                      DAC   STOR
4717                      DAC   SMIS
4718                    *
4719                    * **** EMPTY-BUFFERS ****
4720                    *     HEAD  FNUL,EMPTY-BUFFERS,MTBF,DOCL
4721                    TLNK  SET   *
4722                      VFD   1,0,1,FNUL,6,13,8,'305
4723                    *                             =<FNUL,13>,'E'
4724                      BCI   5,MPTY-BUFFE
4725                      VFD   8,'322,8,'123     ='RS'
4726                      DAC   LINK
4727                    LINK  SET   TLNK
4728                    MTBF  JST   DOCL
4729                      DAC   FRST
4730                      DAC   AT
4731                      DAC   LIMT
4732                      DAC   AT
4733                      DAC   OVER
4734                      DAC   SUB
4735                      DAC   ERAS
4736                      DAC   SMIS
4737                    *
4738                    * **** FLUSH ****
4739                    * SOME SYSTEMS DEFINE THIS IN THE EDITOR, NOT HERE.
4740                    *     HEAD  FNUL,FLUSH,FLSH,DOCL
4741                    TLNK  SET   *
4742                      VFD   1,0,1,FNUL,6,5,8,'306
4743                    *                             =<FNUL,5>,'F'
4744                      BCI   1,LU
4745                      VFD   8,'323,8,'110     ='SH'
4746                      DAC   LINK
4747                    LINK  SET   TLNK
4748                    FLSH  JST   DOCL
4749                      DAC   LIMT
4750                      DAC   AT
4751                      DAC   FRST
4752                      DAC   AT
4753                      DAC   XDO
4754                    FLS1  DAC   I
4755                      DAC   AT
4756                      DAC   ZLES
4757                      DAC   ZBRA
4758                      DAC   FLS2
```

```
4759                      DAC   I
4760                      DAC   TWOP
4761                      DAC   I
4762                      DAC   AT
4763                      DAC   LIT
4764                      OCT   77777
4765                      DAC   AND
4766                      DAC   ZERO
4767                      DAC   RW
4768                 FLS2 DAC   BBUF
4769                      DAC   LIT
4770                      OCT   4
4771                      DAC   PLUS
4772                      DAC   XPLO
4773                      DAC   FLS1
4774                      DAC   MTBF
4775                      DAC   SMIS
4776                 *
4777                 * **** DR0 ****
4778                 *  SELECT DRIVE #0
4779                 *    HEAD  FNUL,DR0,DR0,DOCL
4780                 TLNK SET   *
4781                      VFD   1,0,1,FNUL,6,3,8,'304
4782                 *                            =<FNUL,3>,'D'
4783                      VFD   8,'322,8,'060     ='R0'
4784                      DAC   LINK
4785                 LINK SET   TLNK
4786                 DR0  JST   DOCL
4787                      DAC   ZERO
4788                      DAC   OFST
4789                      DAC   STOR
4790                      DAC   SMIS
4791                 *
4792                 * **** DR1 ****
4793                 *  SELECT DRIVE #1
4794                 *    HEAD  FNUL,DR1,DR1,DOCL
4795                 TLNK SET   *
4796                      VFD   1,0,1,FNUL,6,3,8,'304
4797                 *                            =<FNUL,3>,'D'
4798                      VFD   8,'322,8,'061     ='R1'
4799                      DAC   LINK
4800                 LINK SET   TLNK
4801                 DR1  JST   DOCL
4802                      DAC   LIT
4803                      DEC   240
4804                      DAC   OFST
4805                      DAC   STOR
4806                      DAC   SMIS
4807                 *
4808                 * **** BUFFER ****
```

```
4809                *    HEAD  FNUL,BUFFER,BUFR,DOCL
4810                TLNK SET   *
4811                     VFD   1,0,1,FNUL,6,6,8,'302
4812                *                             =<FNUL,6>,'B'
4813                     BCI   2,UFFE
4814                     VFD   8,'122             ='R'
4815                     DAC   LINK
4816                LINK SET   TLNK
4817                BUFR JST   DOCL
4818                     DAC   USE
4819                     DAC   AT
4820                     DAC   DUP
4821                     DAC   TOR
4822                BUF1 DAC   PBUF
4823                     DAC   ZBRA
4824                     DAC   BUF1
4825                     DAC   USE
4826                     DAC   STOR
4827                     DAC   R
4828                     DAC   AT
4829                     DAC   ZLES
4830                     DAC   ZBRA
4831                     DAC   BUF2
4832                     DAC   R
4833                     DAC   TWOP
4834                     DAC   R
4835                     DAC   AT
4836                     DAC   LIT
4837                     OCT   77777
4838                     DAC   AND
4839                     DAC   ZERO
4840                     DAC   RW
4841                BUF2 DAC   R
4842                     DAC   STOR
4843                     DAC   R
4844                     DAC   PREV
4845                     DAC   STOR
4846                     DAC   FRMR
4847                     DAC   TWOP
4848                     DAC   SMIS
4849                *
4850                * **** BLOCK ****
4851                *    HEAD  FNUL,BLOCK,BLCK,DOCL
4852                TLNK SET   *
4853                     VFD   1,0,1,FNUL,6,5,8,'302
4854                *                             =<FNUL,5>,'B'
4855                     BCI   1,LO
4856                     VFD   8,'303,8,'113      ='CK'
4857                     DAC   LINK
4858                LINK SET   TLNK
```

```
4859                BLCK JST   DOCL
4860                * CHANGED TO MASK OFF THE UPDATE BIT WHEN COMPARING
4861                     DAC   OFST
4862                     DAC   AT
4863                     DAC   PLUS
4864                     DAC   TOR
4865                     DAC   PREV
4866                     DAC   AT
4867                     DAC   DUP
4868                     DAC   AT
4869                     DAC   LIT
4870                     OCT   077777
4871                     DAC   AND
4872                     DAC   R
4873                     DAC   SUB
4874                     DAC   ZBRA
4875                     DAC   BLK3
4876                BLK1 DAC   PBUF
4877                     DAC   ZEQU
4878                     DAC   ZBRA
4879                     DAC   BLK2
4880                     DAC   DROP
4881                     DAC   R
4882                     DAC   BUFR
4883                     DAC   DUP
4884                     DAC   R
4885                     DAC   ONE
4886                     DAC   RW
4887                     DAC   TWO
4888                     DAC   SUB
4889                BLK2 DAC   DUP
4890                     DAC   AT
4891                     DAC   LIT
4892                     OCT   077777
4893                     DAC   AND
4894                     DAC   R
4895                     DAC   SUB
4896                     DAC   ZEQU
4897                     DAC   ZBRA
4898                     DAC   BLK1
4899                     DAC   DUP
4900                     DAC   PREV
4901                     DAC   STOR
4902                BLK3 DAC   FRMR
4903                     DAC   DROP
4904                     DAC   TWOP
4905                     DAC   SMIS
4906                *
4907                * **** (LINE) ****
4908                *    HEAD  FNUL,(LINE),PLIN,DOCL
```

```
4909                    TLNK SET   *
4910                         VFD   1,0,1,FNUL,6,6,8,'250
4911                    *                              =<FNUL,6>,'('
4912                         BCI   2,LINE
4913                         VFD   8,'051              =')'
4914                         DAC   LINK
4915                    LINK SET   TLNK
4916                    PLIN JST   DOCL
4917                         DAC   TOR
4918                         DAC   CL
4919                         DAC   BBUF
4920                         DAC   SSMD
4921                         DAC   FRMR
4922                         DAC   BSCR
4923                         DAC   STAR
4924                         DAC   PLUS
4925                         DAC   BLCK
4926                         DAC   PLUS
4927                         DAC   CL
4928                         DAC   SMIS
4929                    *
4930                    * **** .LINE ****
4931                    *    HEAD  FNUL,.LINE,DLIN,DOCL
4932                    TLNK SET   *
4933                         VFD   1,0,1,FNUL,6,5,8,'256
4934                    *                              =<FNUL,5>,'.'
4935                         BCI   1,LI
4936                         VFD   8,'316,8,'105     ='NE'
4937                         DAC   LINK
4938                    LINK SET   TLNK
4939                    DLIN JST   DOCL
4940                         DAC   PLIN
4941                         DAC   DTRA
4942                         DAC   TYPE
4943                         DAC   SMIS
4944                         ENDC
4945                    *
4946                    * **** MESSAGE ****
4947                    *    HEAD  FNUL,MESSAGE,MESS,DOCL
4948         005022     TLNK SET   *
4949 05022   003715          VFD   1,0,1,FNUL,6,7,8,'315
4950                    *                              =<FNUL,7>,'M'
4951 05023   142723          BCI   2,ESSA
     05024   151701
4952 05025   143505          VFD   8,'307,8,'105     ='GE'
4953 05026   0 004763        DAC   LINK
4954         005022     LINK SET   TLNK
4955 05027   0 10 00130 MESS JST   DOCL
4956 05030   0 002026        DAC   WARN
4957 05031   0 001565        DAC   AT
```

```
4958 05032   0 000277        DAC   ZBRA
4959 05033   0 005040        DAC   MES2
4960                         IFN   DISK
4961                         DAC   DDUP
4962                         DAC   ZBRA
4963                         DAC   MES1
4964                         DAC   LIT
4965                         OCT   4
4966                         DAC   OFST
4967                         DAC   AT
4968                         DAC   BSCR
4969                         DAC   SLSH
4970                         DAC   SUB
4971                         DAC   DLIN
4972                   MES1  DAC   BRAN
4973                         DAC   MES3
4974                         ELSE
4975 05034   0 001722        DAC   ONE               TRUE - ALWAYS
4976 05035   0 000241        DAC   LIT
4977 05036   000006          OCT   6                 DISK RANGE
4978 05037   0 003662        DAC   EROR
4979                         ENDC
4980 05040   0 003031   MES2 DAC   PDTQ
4981                    *    STRG  MSG #$
4982 05041   003315          VFD   8,6,8,'315        =6,'M'
4983 05042   151707          VFD   8,'323,8,'307     ='SG'
4984 05043   120243          VFD   8,'240,8,'243     ='  '
4985 05044   120000          VFD   8,'240            =' '
4986 05045   0 005537        DAC   DOT
4987 05046   0 001361   MES3 DAC   SMIS
4988                         IFN   DISK
4989                    *
4990                    * **** LOAD ****
4991                    *    HEAD  FNUL,LOAD,LOAD,DOCL
4992                    TLNK SET   *
4993                         VFD   1,0,1,FNUL,6,4,8,'314
4994                    *                              =<FNUL,4>,'L'
4995                         BCI   1,OA
4996                         VFD   8,'104            ='D'
4997                         DAC   LINK
4998                    LINK SET   TLNK
4999                    LOAD JST   DOCL
5000                         DAC   BLK
5001                         DAC   AT
5002                         DAC   TOR
5003                         DAC   IN
5004                         DAC   AT
5005                         DAC   TOR
5006                         DAC   ZERO
5007                         DAC   IN
```

```
5008                      DAC   STOR
5009                      DAC   BSCR
5010                      DAC   STAR
5011                      DAC   BLK
5012                      DAC   STOR
5013                      DAC   ITRP
5014                      DAC   FRMR
5015                      DAC   IN
5016                      DAC   STOR
5017                      DAC   FRMR
5018                      DAC   BLK
5019                      DAC   STOR
5020                      DAC   SMIS
5021                  *
5022                  * **** --> ****
5023                  *    HEAD  FNUL,-->,AROW,DOCL
5024                  TLNK SET   *
5025                      VFD   1,0,1,FNUL,6,3,8,'255
5026                  *                            =<FNUL,3>,'-'
5027                      VFD   8,'255,8,'076      ='->'
5028                      DAC   LINK
5029                  LINK SET   TLNK
5030                  AROW JST   DOCL
5031                      DAC   QLDG
5032                      DAC   ZERO
5033                      DAC   IN
5034                      DAC   STOR
5035                      DAC   BSCR
5036                      DAC   BLK
5037                      DAC   AT
5038                      DAC   OVER
5039                      DAC   MOD
5040                      DAC   SUB
5041                      DAC   BLK
5042                      DAC   PSTR
5043                      DAC   SMIS
5044                  *
5045                  * **** R/W **** ( ADDRESS SCREEN# FLAG ==> )
5046                  *    HEAD  FNUL,R/W,RW,DOCL
5047                  TLNK SET   *
5048                      VFD   1,0,1,FNUL,6,3,8,'322
5049                  *                            =<FNUL,3>,'R'
5050                      VFD   8,'257,8,'127      ='/W'
5051                      DAC   LINK
5052                  LINK SET   TLNK
5053                  RW   JST   DOCL
5054                      DAC   DROP
5055                      DAC   DROP
5056                      DAC   DROP
5057                      DAC   SMIS
```

```
5058                    ENDC
5059                    EJCT
```

```
                    ****************************************************************
                    *
                    *       MISCELLANEOUS HIGHER LEVEL
                    *
                    ****************************************************************
                    *
                    * **** ' ****
                    *    HEAD  FIMD,',TICK,DOCL
         005047     TLNK SET   *
 05047   040447           VFD   1,0,1,FIMD,6,1,8,'047
                    *                              =<FIMD,1>,'''
 05050   0 005022         DAC   LINK
         005047     LINK SET   TLNK
 05051   0 10 00130 TICK JST   DOCL
 05052   0 003562         DAC   DFND
 05053   0 001423         DAC   ZEQU
 05054   0 001716         DAC   ZERO
 05055   0 002455         DAC   QERR
 05056   0 001526         DAC   DROP
 05057   0 004050         DAC   LTRL
 05060   0 001361         DAC   SMIS
                    *
                    * **** FORGET ****
                    *    HEAD  FNUL,FORGET,FRGT,DOCL
         005061     TLNK SET   *
 05061   003306           VFD   1,0,1,FNUL,6,6,8,'306
                    *                              =<FNUL,6>,'F'
 05062   147722           BCI   2,ORGE
 05063   143705
 05064   052000           VFD   8,'124             ='T'
 05065   0 005047         DAC   LINK
         005061     LINK SET   TLNK
 05066   0 10 00130 FRGT JST   DOCL
 05067   0 002136         DAC   CURR
 05070   0 001565         DAC   AT
 05071   0 002127         DAC   CONT
 05072   0 001565         DAC   AT
 05073   0 002266         DAC   SUB
 05074   0 000241         DAC   LIT
 05075   000030           OCT   30
 05076   0 002455         DAC   QERR
 05077   0 005051         DAC   TICK
 05100   0 001541         DAC   DUP
 05101   0 002034         DAC   FENC
 05102   0 001565         DAC   AT
 05103   0 002303         DAC   LESS
 05104   0 000241         DAC   LIT
 05105   000025           OCT   25
 05106   0 002455         DAC   QERR
 05107   0 001541         DAC   DUP
```

```
5109 05110   0 002405        DAC   NFA
5110 05111   0 002041        DAC   DP
5111 05112   0 001577        DAC   STOR
5112 05113   0 002367        DAC   LFA
5113 05114   0 001565        DAC   AT
5114 05115   0 002127        DAC   CONT
5115 05116   0 001565        DAC   AT
5116 05117   0 001577        DAC   STOR
5117 05120   0 001361        DAC   SMIS
5118                     *
5119                     *
5120                     *
5121                     * **** BACK ****
5122                     *    HEAD  FNUL,BACK,BACK,DOCL
5123         005121      TLNK SET   *
5124 05121   002302           VFD   1,0,1,FNUL,6,4,8,'302
5125                     *                              =<FNUL,4>,'B'
5126 05122   140703           BCI   1,AC
5127 05123   045400           VFD   8,'113           ='K'
5128 05124   0 005061        DAC   LINK
5129         005121      LINK SET   TLNK
5130 05125   0 10 00130 BACK JST   DOCL
5131                     * JUST COMPILE THE TARGET WORD
5132                     * NOT THE DIFFERENCE
5133 05126   0 002256        DAC   COMA
5134 05127   0 001361        DAC   SMIS
5135                     *
5136                     * **** BEGIN ****
5137                     *    HEAD  FIMD,BEGIN,BGIN,DOCL
5138         005130      TLNK SET   *
5139 05130   042702           VFD   1,0,1,FIMD,6,5,8,'302
5140                     *                              =<FIMD,5>,'B'
5141 05131   142707           BCI   1,EG
5142 05132   144516           VFD   8,'311,8,'116    ='IN'
5143 05133   0 005121        DAC   LINK
5144         005130      LINK SET   TLNK
5145 05134   0 10 00130 BGIN JST   DOCL
5146 05135   0 002472        DAC   QCMP
5147 05136   0 002240        DAC   HERE
5148 05137   0 001722        DAC   ONE
5149 05140   0 001361        DAC   SMIS
5150                     *
5151                     * **** ENDIF ****
5152                     *    HEAD  FIMD,ENDIF,ENDF,DOCL
5153         005141      TLNK SET   *
5154 05141   042705           VFD   1,0,1,FIMD,6,5,8,'305
5155                     *                              =<FIMD,5>,'E'
5156 05142   147304           BCI   1,ND
5157 05143   144506           VFD   8,'311,8,'106    ='IF'
5158 05144   0 005130        DAC   LINK
```

```
5159         005141     LINK SET   TLNK
5160 05145   0 10 00130 ENDF JST   DOCL
5161 05146   0 002472         DAC   QCMP
5162 05147   0 001726         DAC   TWO
5163 05150   0 002522         DAC   QPRS
5164 05151   0 002240         DAC   HERE
5165 05152   0 001533         DAC   SWAP
5166 05153   0 001577         DAC   STOR
5167 05154   0 001361         DAC   SMIS
5168                     *
5169                     * **** THEN ****
5170                     *    HEAD  FIMD,THEN,THEN,DOCL
5171         005155     TLNK SET   *
5172 05155   042324          VFD   1,0,1,FIMD,6,4,8,'324
5173                     *                              =<FIMD,4>,'T'
5174 05156   144305          BCI   1,HE
5175 05157   047000          VFD   8,'116             ='N'
5176 05160   0 005141        DAC   LINK
5177         005155     LINK SET   TLNK
5178 05161   0 10 00130 THEN JST   DOCL
5179 05162   0 005145        DAC   ENDF
5180 05163   0 001361        DAC   SMIS
5181                     *
5182                     * **** DO ****
5183                     *    HEAD  FIMD,DO,DO,DOCL
5184         005164     TLNK SET   *
5185 05164   041304          VFD   1,0,1,FIMD,6,2,8,'304
5186                     *                              =<FIMD,2>,'D'
5187 05165   047400          VFD   8,'117             ='O'
5188 05166   0 005155        DAC   LINK
5189         005164     LINK SET   TLNK
5190 05167   0 10 00130 DO   JST   DOCL
5191 05170   0 002570        DAC   COMP
5192 05171   0 000372        DAC   XDO
5193 05172   0 002240        DAC   HERE
5194 05173   0 000241        DAC   LIT
5195 05174   000003          DEC   3
5196 05175   0 001361        DAC   SMIS
5197                     *
5198                     * **** LOOP ****
5199                     *    HEAD  FIMD,LOOP,LOOP,DOCL
5200         005176     TLNK SET   *
5201 05176   042314          VFD   1,0,1,FIMD,6,4,8,'314
5202                     *                              =<FIMD,4>,'L'
5203 05177   147717          BCI   1,OO
5204 05200   050000          VFD   8,'120             ='P'
5205 05201   0 005164        DAC   LINK
5206         005176     LINK SET   TLNK
5207 05202   0 10 00130 LOOP JST   DOCL
5208 05203   0 000241        DAC   LIT
```

```
5209 05204   000003          DEC   3
5210 05205   0 002522        DAC   QPRS
5211 05206   0 002570        DAC   COMP
5212 05207   0 000322        DAC   XLOP
5213 05210   0 005125        DAC   BACK
5214 05211   0 001361        DAC   SMIS
5215                 *
5216                 * **** +LOOP ****
5217                 *    HEAD  FIMD,+LOOP,PLOP,DOCL
5218         005212      TLNK SET   *
5219 05212   042653          VFD   1,0,1,FIMD,6,5,8,'253
5220                 *                              =<FIMD,5>,'+'
5221 05213   146317          BCI   1,LO
5222 05214   147520          VFD   8,'317,8,'120     ='OP'
5223 05215   0 005176        DAC   LINK
5224         005212      LINK SET   TLNK
5225 05216   0 10 00130 PLOP JST   DOCL
5226 05217   0 000241        DAC   LIT
5227 05220   000003          DEC   3
5228 05221   0 002522        DAC   QPRS
5229 05222   0 002570        DAC   COMP
5230 05223   0 000354        DAC   XPLO
5231 05224   0 005125        DAC   BACK
5232 05225   0 001361        DAC   SMIS
5233                 *
5234                 * **** UNTIL ****
5235                 *    HEAD  FIMD,UNTIL,UNTL,DOCL
5236         005226      TLNK SET   *
5237 05226   042725          VFD   1,0,1,FIMD,6,5,8,'325
5238                 *                              =<FIMD,5>,'U'
5239 05227   147324          BCI   1,NT
5240 05230   144514          VFD   8,'311,8,'114     ='IL'
5241 05231   0 005212        DAC   LINK
5242         005226      LINK SET   TLNK
5243 05232   0 10 00130 UNTL JST   DOCL
5244 05233   0 001722        DAC   ONE
5245 05234   0 002522        DAC   QPRS
5246 05235   0 002570        DAC   COMP
5247 05236   0 000277        DAC   ZBRA
5248 05237   0 005125        DAC   BACK
5249 05240   0 001361        DAC   SMIS
5250                 *
5251                 * **** END ****
5252                 *    HEAD  FIMD,END,END,DOCL
5253         005241      TLNK SET   *
5254 05241   041705          VFD   1,0,1,FIMD,6,3,8,'305
5255                 *                              =<FIMD,3>,'E'
5256 05242   147104          VFD   8,'316,8,'104     ='ND'
5257 05243   0 005226        DAC   LINK
5258         005241      LINK SET   TLNK
```

```
5259 05244   0 10 00130 END  JST   DOCL
5260 05245   0 005232        DAC   UNTL
5261 05246   0 001361        DAC   SMIS
5262                     *
5263                     * **** AGAIN ****
5264                     *    HEAD  FIMD,AGAIN,AGAN,DOCL
5265         005247      TLNK SET   *
5266 05247   042701          VFD   1,0,1,FIMD,6,5,8,'301
5267                     *                              =<FIMD,5>,'A'
5268 05250   143701          BCI   1,GA
5269 05251   144516          VFD   8,'311,8,'116      ='IN'
5270 05252   0 005241        DAC   LINK
5271         005247      LINK SET   TLNK
5272 05253   0 10 00130 AGAN JST   DOCL
5273 05254   0 001722        DAC   ONE
5274 05255   0 002522        DAC   QPRS
5275 05256   0 002570        DAC   COMP
5276 05257   0 000263        DAC   BRAN
5277 05260   0 005125        DAC   BACK
5278 05261   0 001361        DAC   SMIS
5279                     *
5280                     * **** REPEAT ****
5281                     *    HEAD  FIMD,REPEAT,RPET,DOCL
5282         005262      TLNK SET   *
5283 05262   043322          VFD   1,0,1,FIMD,6,6,8,'322
5284                     *                              =<FIMD,6>,'R'
5285 05263   142720          BCI   2,EPEA
     05264   142701
5286 05265   052000          VFD   8,'124             ='T'
5287 05266   0 005247        DAC   LINK
5288         005262      LINK SET   TLNK
5289 05267   0 10 00130 RPET JST   DOCL
5290 05270   0 001402        DAC   TOR
5291 05271   0 001402        DAC   TOR
5292 05272   0 005253        DAC   AGAN
5293 05273   0 001412        DAC   FRMR
5294 05274   0 001412        DAC   FRMR
5295 05275   0 001726        DAC   TWO
5296 05276   0 002266        DAC   SUB
5297 05277   0 005145        DAC   ENDF
5298 05300   0 001361        DAC   SMIS
5299                     *
5300                     * **** IF ****
5301                     *    HEAD  FIMD,IF,IF,DOCL
5302         005301      TLNK SET   *
5303 05301   041311          VFD   1,0,1,FIMD,6,2,8,'311
5304                     *                              =<FIMD,2>,'I'
5305 05302   043000          VFD   8,'106             ='F'
5306 05303   0 005262        DAC   LINK
5307         005301      LINK SET   TLNK
```

```
5308 05304   0 10 00130 IF   JST   DOCL
5309 05305   0 002570        DAC   COMP
5310 05306   0 000277        DAC   ZBRA
5311 05307   0 002240        DAC   HERE
5312 05310   0 001716        DAC   ZERO
5313 05311   0 002256        DAC   COMA
5314 05312   0 001726        DAC   TWO
5315 05313   0 001361        DAC   SMIS
5316                 *
5317                 * **** ELSE ****
5318                 *    HEAD  FIMD,ELSE,ELSE,DOCL
5319         005314     TLNK SET   *
5320 05314   042305          VFD   1,0,1,FIMD,6,4,8,'305
5321                 *                              =<FIMD,4>,'E'
5322 05315   146323          BCI   1,LS
5323 05316   042400          VFD   8,'105           ='E'
5324 05317   0 005301        DAC   LINK
5325         005314     LINK SET   TLNK
5326 05320   0 10 00130 ELSE JST   DOCL
5327 05321   0 001726        DAC   TWO
5328 05322   0 002522        DAC   QPRS
5329 05323   0 002570        DAC   COMP
5330 05324   0 000263        DAC   BRAN
5331 05325   0 002240        DAC   HERE
5332 05326   0 001716        DAC   ZERO
5333 05327   0 002256        DAC   COMA
5334 05330   0 001533        DAC   SWAP
5335 05331   0 001726        DAC   TWO
5336 05332   0 005145        DAC   ENDF
5337 05333   0 001726        DAC   TWO
5338 05334   0 001361        DAC   SMIS
5339                 *
5340                 * **** WHILE ****
5341                 *    HEAD  FIMD,WHILE,WHIL,DOCL
5342         005335     TLNK SET   *
5343 05335   042727          VFD   1,0,1,FIMD,6,5,8,'327
5344                 *                              =<FIMD,5>,'W'
5345 05336   144311          BCI   1,HI
5346 05337   146105          VFD   8,'314,8,'105    ='LE'
5347 05340   0 005314        DAC   LINK
5348         005335     LINK SET   TLNK
5349 05341   0 10 00130 WHIL JST   DOCL
5350 05342   0 005304        DAC   IF
5351 05343   0 002231        DAC   TWOP
5352 05344   0 001361        DAC   SMIS
5353                 *
5354                 *
5355                 *
5356                 * **** SPACES ****
5357                 *    HEAD  FNUL,SPACES,SPCS,DOCL
```

```
5358         005345     TLNK SET   *
5359 05345   003323          VFD   1,0,1,FNUL,6,6,8,'323
5360                        *                              =<FNUL,6>,'S'
5361 05346   150301          BCI   2,PACE
     05347   141705
5362 05350   051400          VFD   8,'123                   ='S'
5363 05351   0 005335        DAC   LINK
5364         005345     LINK SET   TLNK
5365 05352   0 10 00130 SPCS JST   DOCL
5366 05353   0 001716        DAC   ZERO
5367 05354   0 004507        DAC   MAX
5368 05355   0 002345        DAC   DDUP
5369 05356   0 000277        DAC   ZBRA
5370 05357   0 005365        DAC   SPC2
5371 05360   0 001716        DAC   ZERO
5372 05361   0 000372        DAC   XDO
5373 05362   0 002335   SPC1 DAC   SPCE
5374 05363   0 000322        DAC   XLOP
5375 05364   0 005362        DAC   SPC1
5376 05365   0 001361   SPC2 DAC   SMIS
5377                        *
5378                        * **** <# ****
5379                        *    HEAD  FNUL,<#,BDGS,DOCL
5380         005366     TLNK SET   *
5381 05366   001274          VFD   1,0,1,FNUL,6,2,8,'274
5382                        *                              =<FNUL,2>,'<'
5383 05367   021400          VFD   8,'043                   =' '
5384 05370   0 005345        DAC   LINK
5385         005366     LINK SET   TLNK
5386 05371   0 10 00130 BDGS JST   DOCL
5387 05372   0 003351        DAC   PAD
5388 05373   0 001616        DAC   BYTE
5389 05374   0 002203        DAC   HLD
5390 05375   0 001577        DAC   STOR
5391 05376   0 001361        DAC   SMIS
5392                        *
5393                        * **** #> ****
5394                        *    HEAD  FNUL,#>,EDGS,DOCL
5395         005377     TLNK SET   *
5396 05377   001243          VFD   1,0,1,FNUL,6,2,8,'243
5397                        *                              =<FNUL,2>,'#'
5398 05400   037000          VFD   8,'076                   ='>'
5399 05401   0 005366        DAC   LINK
5400         005377     LINK SET   TLNK
5401 05402   0 10 00130 EDGS JST   DOCL
5402 05403   0 001526        DAC   DROP
5403 05404   0 001526        DAC   DROP
5404 05405   0 002203        DAC   HLD
5405 05406   0 001565        DAC   AT
5406 05407   0 003351        DAC   PAD
```

```
5407 05410   0 001616        DAC   BYTE
5408 05411   0 001520        DAC   OVER
5409 05412   0 002266        DAC   SUB
5410 05413   0 001361        DAC   SMIS
5411                     *
5412                     * **** SIGN ****
5413                     *    HEAD  FNUL,SIGN,SIGN,DOCL
5414         005414      TLNK SET   *
5415 05414   002323           VFD   1,0,1,FNUL,6,4,8,'323
5416                     *                              =<FNUL,4>,'S'
5417 05415   144707          BCI   1,IG
5418 05416   047000          VFD   8,'116            ='N'
5419 05417   0 005377        DAC   LINK
5420         005414      LINK SET   TLNK
5421 05420   0 10 00130  SIGN JST   DOCL
5422 05421   0 002325        DAC   ROT
5423 05422   0 001435        DAC   ZLES
5424 05423   0 000277        DAC   ZBRA
5425 05424   0 005430        DAC   SGN1
5426 05425   0 000241        DAC   LIT
5427 05426   000255          VFD   16,CMNS
5428 05427   0 003335        DAC   HOLD
5429 05430   0 001361    SGN1 DAC   SMIS
5430                     *
5431                     * **** # ****
5432                     *    HEAD  FNUL,#,DIG,DOCL
5433         005431      TLNK SET   *
5434 05431   000443          VFD   1,0,1,FNUL,6,1,8,'043
5435                     *                              =<FNUL,1>,'#'
5436 05432   0 005414        DAC   LINK
5437         005431      LINK SET   TLNK
5438 05433   0 10 00130  DIG  JST   DOCL
5439 05434   0 002152        DAC   BASE
5440 05435   0 001565        DAC   AT
5441 05436   0 004737        DAC   MSMD
5442 05437   0 002325        DAC   ROT
5443 05440   0 000241        DAC   LIT
5444 05441   000011          DEC   9
5445 05442   0 001520        DAC   OVER
5446 05443   0 002303        DAC   LESS
5447 05444   0 000277        DAC   ZBRA
5448 05445   0 005451        DAC   DIG1
5449 05446   0 000241        DAC   LIT
5450 05447   000007          OCT   7
5451 05450   0 001443        DAC   PLUS
5452 05451   0 000241    DIG1 DAC   LIT
5453 05452   000260          VFD   16,CZRO
5454 05453   0 001443        DAC   PLUS
5455 05454   0 003335        DAC   HOLD
5456 05455   0 001361        DAC   SMIS
```

```
5457                    *
5458                    * **** #S ****
5459                    *    HEAD  FNUL,#S,DIGS,DOCL
5460          005456    TLNK SET   *
5461 05456   001243           VFD   1,0,1,FNUL,6,2,8,'243
5462                    *                              =<FNUL,2>,'#'
5463 05457   051400           VFD   8,'123             ='S'
5464 05460   0 005431         DAC   LINK
5465          005456    LINK SET   TLNK
5466 05461   0 10 00130 DIGS JST   DOCL
5467 05462   0 005433   DGS1 DAC   DIG
5468 05463   0 001520         DAC   OVER
5469 05464   0 001520         DAC   OVER
5470 05465   0 001306         DAC   OR
5471 05466   0 001423         DAC   ZEQU
5472 05467   0 000277         DAC   ZBRA
5473 05470   0 005462         DAC   DGS1
5474 05471   0 001361         DAC   SMIS
5475                    *
5476                    * **** D.R ****
5477                    *    HEAD  FNUL,D.R,DDTR,DOCL
5478          005472    TLNK SET   *
5479 05472   001704           VFD   1,0,1,FNUL,6,3,8,'304
5480                    *                              =<FNUL,3>,'D'
5481 05473   127122           VFD   8,'256,8,'122      ='.R'
5482 05474   0 005456         DAC   LINK
5483          005472    LINK SET   TLNK
5484 05475   0 10 00130 DDTR JST   DOCL
5485 05476   0 001402         DAC   TOR
5486 05477   0 001533         DAC   SWAP
5487 05500   0 001520         DAC   OVER
5488 05501   0 004461         DAC   DABS
5489 05502   0 005371         DAC   BDGS
5490 05503   0 005461         DAC   DIGS
5491 05504   0 005420         DAC   SIGN
5492 05505   0 005402         DAC   EDGS
5493 05506   0 001412         DAC   FRMR
5494 05507   0 001520         DAC   OVER
5495 05510   0 002266         DAC   SUB
5496 05511   0 005352         DAC   SPCS
5497 05512   0 002750         DAC   TYPE
5498 05513   0 001361         DAC   SMIS
5499                    *
5500                    * **** .R ****
5501                    *    HEAD  FNUL,.R,DOTR,DOCL
5502          005514    TLNK SET   *
5503 05514   001256           VFD   1,0,1,FNUL,6,2,8,'256
5504                    *                              =<FNUL,2>,'.'
5505 05515   051000           VFD   8,'122             ='R'
5506 05516   0 005472         DAC   LINK
```

```
5507         005514     LINK SET   TLNK
5508 05517   0 10 00130 DOTR JST   DOCL
5509 05520   0 001402        DAC   TOR
5510 05521   0 004434        DAC   STOD
5511 05522   0 001412        DAC   FRMR
5512 05523   0 005475        DAC   DDTR
5513 05524   0 001361        DAC   SMIS
5514                     *
5515                     * **** D. ****
5516                     *    HEAD  FNUL,D.,DDOT,DOCL
5517         005525     TLNK SET   *
5518 05525   001304          VFD   1,0,1,FNUL,6,2,8,'304
5519                     *                              =<FNUL,2>,'D'
5520 05526   027000          VFD   8,'056                =','
5521 05527   0 005514        DAC   LINK
5522         005525     LINK SET   TLNK
5523 05530   0 10 00130 DDOT JST   DOCL
5524 05531   0 001716        DAC   ZERO
5525 05532   0 005475        DAC   DDTR
5526 05533   0 002335        DAC   SPCE
5527 05534   0 001361        DAC   SMIS
5528                     *
5529                     * **** . ****
5530                     *    HEAD  FNUL,.,DOT,DOCL
5531         005535     TLNK SET   *
5532 05535   000456          VFD   1,0,1,FNUL,6,1,8,'056
5533                     *                              =<FNUL,1>,'.'
5534 05536   0 005525        DAC   LINK
5535         005535     LINK SET   TLNK
5536 05537   0 10 00130 DOT  JST   DOCL
5537 05540   0 004434        DAC   STOD
5538 05541   0 005530        DAC   DDOT
5539 05542   0 001361        DAC   SMIS
5540                     *
5541                     * **** ? ****
5542                     *    HEAD  FNUL,?,QUST,DOCL
5543         005543     TLNK SET   *
5544 05543   000477          VFD   1,0,1,FNUL,6,1,8,'077
5545                     *                              =<FNUL,1>,'?'
5546 05544   0 005535        DAC   LINK
5547         005543     LINK SET   TLNK
5548 05545   0 10 00130 QUST JST   DOCL
5549 05546   0 001565        DAC   AT
5550 05547   0 005537        DAC   DOT
5551 05550   0 001361        DAC   SMIS
5552                     *
5553                     * **** U. ****
5554                     *    HEAD  FNUL,U.,UDOT,DOCL
5555         005551     TLNK SET   *
5556 05551   001325          VFD   1,0,1,FNUL,6,2,8,'325
```

```
5557                    *                               =<FNUL,2>,'U'
5558 05552   027000          VFD   8,'056              ='.'
5559 05553   0 005543        DAC   LINK
5560         005551     LINK SET   TLNK
5561 05554   0 10 00130 UDOT JST   DOCL
5562 05555   0 001716        DAC   ZERO
5563 05556   0 005530        DAC   DDOT
5564 05557   0 001361        DAC   SMIS
5565                    ****************************************************************
5566                    *
5567                    *  UTILITY SECTION.
5568                    *
5569                    ****************************************************************
5570                         IFN   DISK
5571                    *
5572                    * **** LIST ****
5573                    *    HEAD  FNUL,LIST,LIST,DOCL
5574                    TLNK SET   *
5575                         VFD   1,0,1,FNUL,6,4,8,'314
5576                    *                               =<FNUL,4>,'L'
5577                         BCI   1,IS
5578                         VFD   8,'124              ='T'
5579                         DAC   LINK
5580                    LINK SET   TLNK
5581                    LIST JST   DOCL
5582                         DAC   DEC
5583                         DAC   CR
5584                         DAC   DUP
5585                         DAC   SCR
5586                         DAC   STOR
5587                         DAC   PDTQ
5588                    *    STRG  SCR #$
5589                         VFD   8,6,8,'323          =6,'S'
5590                         VFD   8,'303,8,'322       ='CR'
5591                         VFD   8,'240,8,'243       ='  '
5592                         VFD   8,'240              =' '
5593                         DAC   DOT
5594                         DAC   LIT
5595                         OCT   20
5596                         DAC   ZERO
5597                         DAC   XDO
5598                    LST1 DAC   CR
5599                         DAC   I
5600                         DAC   THRE
5601                         DAC   DOTR
5602                         DAC   SPCE
5603                         DAC   I
5604                         DAC   SCR
5605                         DAC   AT
5606                         DAC   DLIN
```

```
5607                      DAC   XLOP
5608                      DAC   LST1
5609                      DAC   CR
5610                      DAC   SMIS
5611                 *
5612                 * **** INDEX **** LIST FIRST LINE OF A RANGE OF DISK SCREENS.
5613                 *    HEAD  FNUL,INDEX,INDX,DOCL
5614                 TLNK SET   *
5615                      VFD   1,0,1,FNUL,6,5,8,'311
5616                 *                              =<FNUL,5>,'I'
5617                      BCI   1,ND
5618                      VFD   8,'305,8,'130     ='EX'
5619                      DAC   LINK
5620                 LINK SET   TLNK
5621                 INDX JST   DOCL
5622                      DAC   CR
5623                      DAC   ONEP
5624                      DAC   SWAP
5625                      DAC   XDO
5626                 IDX1 DAC   CR
5627                      DAC   I
5628                      DAC   THRE
5629                      DAC   DOTR
5630                      DAC   SPCE
5631                      DAC   ZERO
5632                      DAC   I
5633                      DAC   DLIN
5634                      DAC   QTRM
5635                      DAC   ZBRA
5636                      DAC   IDX2
5637                      DAC   LEAV
5638                 IDX2 DAC   XLOP
5639                      DAC   IDX1
5640                      DAC   SMIS
5641                 *
5642                 * **** TRIAD **** LIST DISK SCREENS THREE PER PAGE.
5643                 *    HEAD  FNUL,TRIAD,TRAD,DOCL
5644                 TLNK SET   *
5645                      VFD   1,0,1,FNUL,6,5,8,'324
5646                 *                              =<FNUL,5>,'T'
5647                      BCI   1,RI
5648                      VFD   8,'301,8,'104     ='AD'
5649                      DAC   LINK
5650                 LINK SET   TLNK
5651                 TRAD JST   DOCL
5652                      DAC   LIT
5653                      OCT   214               FORM FEED
5654                      DAC   EMIT
5655                      DAC   THRE
5656                      DAC   SLSH
```

```
5657                        DAC   THRE
5658                        DAC   STAR
5659                        DAC   THRE
5660                        DAC   OVER
5661                        DAC   PLUS
5662                        DAC   SWAP
5663                        DAC   XDO
5664                   TRA1 DAC   CR
5665                        DAC   I
5666                        DAC   LIST
5667                        DAC   XLOP
5668                        DAC   TRA1
5669                        DAC   CR
5670                        DAC   LIT
5671                        OCT   17
5672                        DAC   MESS
5673                        DAC   CR
5674                        DAC   SMIS
5675                        ENDC
5676                   *
5677                   * **** VLIST ****
5678                   *    HEAD  FNUL,VLIST,VLST,DOCL
5679           005560  TLNK SET   *
5680 05560 002726           VFD   1,0,1,FNUL,6,5,8,'326
5681                   *                              =<FNUL,5>,'V'
5682 05561 146311           BCI   1,LI
5683 05562 151524           VFD   8,'323,8,'124     ='ST'
5684 05563 0 005551         DAC   LINK
5685           005560  LINK SET   TLNK
5686 05564 0 10 00130 VLST JST   DOCL
5687 05565 0 000241         DAC   LIT
5688 05566 000200           OCT   200
5689 05567 0 002104         DAC   OUT
5690 05570 0 001577         DAC   STOR
5691 05571 0 002127         DAC   CONT
5692 05572 0 001565         DAC   AT
5693 05573 0 001565         DAC   AT
5694 05574 0 002104   VLS1  DAC   OUT
5695 05575 0 001565         DAC   AT
5696 05576 0 000241         DAC   LIT
5697 05577 000100           OCT   100
5698 05600 0 002315         DAC   GRTR
5699 05601 0 000277         DAC   ZBRA
5700 05602 0 005607         DAC   VLS2
5701 05603 0 000627         DAC   CR
5702 05604 0 001716         DAC   ZERO
5703 05605 0 002104         DAC   OUT
5704 05606 0 001577         DAC   STOR
5705 05607 0 001541   VLS2  DAC   DUP
5706 05610 0 003712         DAC   IDDT
```

```
5707 05611   0 002335        DAC   SPCE
5708 05612   0 002335        DAC   SPCE
5709 05613   0 002423        DAC   PFA
5710 05614   0 002367        DAC   LFA
5711 05615   0 001565        DAC   AT
5712 05616   0 001541        DAC   DUP
5713 05617   0 001423        DAC   ZEQU
5714 05620   0 000623        DAC   QTRM
5715 05621   0 001306        DAC   OR
5716 05622   0 000277        DAC   ZBRA
5717 05623   0 005574        DAC   VLS1
5718 05624   0 001526        DAC   DROP
5719 05625   0 001361        DAC   SMIS
5720                     *
5721                     * **** BYE ****
5722                     *    HEAD  FNUL,BYE,BYE
5723         005626      TLNK SET   *
5724 05626   001702           VFD   1,0,1,FNUL,6,3,8,'302
5725                     *                              =<FNUL,3>,'B'
5726 05627   154505           VFD   8,'331,8,'105     ='YE'
5727 05630   0 005560        DAC   LINK
5728         005626      LINK SET   TLNK
5729         005631      BYE  EQU   *
5730 05631   0 01 01056      JMP   STOP
5731                     *
5732                          IFN   DBGW
5733                     *
5734                     * **** DEBUG ****
5735                     *    HEAD  FNUL,DEBUG,DBUG
5736                     TLNK SET   *
5737                          VFD   1,0,1,FNUL,6,5,8,'304
5738                     *                              =<FNUL,5>,'D'
5739                          BCI   1,EB
5740                          VFD   8,'325,8,'107     ='UG'
5741                          DAC   LINK
5742                     LINK SET   TLNK
5743                     DBUG EQU   *
5744                          JMP   PDBG
5745                          ENDC
5746                     ****************************************************************
5747                     *
5748                     *  THE FOLLOWING TWO DEFINITIONS ARE NOT PURE CODE, SO THEY WERE
5749                     *  MOVED HERE, NEAR THE END OF THE DICTIONARY.
5750                     *
5751                     ****************************************************************
5752                     *
5753                     * **** ;CODE **** CREATE NEW DATA TYPE WITH CODE ROUTINE WRITTEN
5754                     *                 IN ASSEMBLY.
5755                     *    HEAD  FIMD,;CODE,SEMC,DOCL
5756         005632      TLNK SET   *
```

```
5757 05632   042673          VFD   1,0,1,FIMD,6,5,8,'273
5758                      *                           =<FIMD,5>,';'
5759 05633   141717          BCI   1,CO
5760 05634   142105          VFD   8,'304,8,'105     ='DE'
5761 05635   0 005626        DAC   LINK
5762         005632     LINK SET   TLNK
5763 05636   0 10 00130 SEMC JST   DOCL
5764 05637   0 002534        DAC   QCSP
5765 05640   0 002570        DAC   COMP
5766 05641   0 002603        DAC   LBRC
5767 05642   0 002625        DAC   SMDG
5768 05643   0 001361        DAC   SMIS
5769 05644   0 002675        DAC   PSCD              WON'T WORK...
5770 05645   000000          HLT                     STOP EXECUTION
5771                      *  NOTE: LATER, THE ASSEMBLER WILL PATCH THIS DEFINITION.
5772                      *
5773                      * **** FORTH ****
5774                      *    HEAD  FIMD,FORTH,FRTH,DODS
5775         005646     TLNK SET   *
5776 05646   042706          VFD   1,0,1,FIMD,6,5,8,'306
5777                      *                           =<FIMD,5>,'F'
5778 05647   147722          BCI   1,OR
5779 05650   152110          VFD   8,'324,8,'110     ='TH'
5780 05651   0 005632        DAC   LINK
5781         005646     LINK SET   TLNK
5782 05652   0 10 00152 FRTH JST   DODS
5783 05653   0 004257        DAC   DOVC
5784                      *
5785                      *
5786 05654   120201          OCT   120201            DUMMY HEADER AT INTERSECTION
5787 05655   0 005657        DAC   XTSK
5788 05656   000000     XXVC OCT   0                 THE VOCABULARY LINK (FOR FUTURE USE)
5789                      *
5790                      * **** TASK ****
5791                      *TSK HEAD  FIMD,TASK,TASK,DOCL
5792         005657     TLNK SET   *
5793 05657   042324     XTSK VFD   1,0,1,FIMD,6,4,8,'324
5794                      *                           =<FIMD,4>,'T'
5795 05660   140723          BCI   1,AS
5796 05661   045400          VFD   8,'113            ='K'
5797 05662   0 005646        DAC   LINK
5798         005657     LINK SET   TLNK
5799 05663   0 10 00130 TASK JST   DOCL
5800 05664   0 001361        DAC   SMIS
5801                      *
5802                      ***************************************************************
5803                      * TERMINAL I/O
5804                      ***************************************************************
5805                      *
5806                      * **** EMIT ****
```

```
5807 05665   1 02 00001 PEMT LDA   1,1
5808 05666   0 03 00733      ANA   =’177            LOSE TOP BIT
5809 05667   0 11 00712      CAS   =’40
5810 05670   0 01 05675      JMP   EMT2             >’40
5811 05671   0 01 05675      JMP   EMT2             =’40
5812                    * IS A CONTROL CHARACTER
5813 05672   1 02 00001 EMT1 LDA   1,1              GET WHOLE CHARACTER BACK
5814 05673   0 10 05702      JST   OUT1
5815 05674   0 01 00113      JMP   POP
5816                    *
5817                    * INCREMENT ’OUT’, UNLESS A CONTROL CHARACTER BEING OUTPUT.
5818 05675   0 02 00103 EMT2 LDA   UP
5819 05676   0 06 00711      ADD   =’21
5820 05677   0 04 00104      STA   T1
5821 05700  -0 12 00104      IRS*  T1
5822 05701   0 01 05672      JMP   EMT1
5823                    *
5824 05702   0 000000   OUT1 DAC   **
5825 05703   34 0104         SKS   ’104
5826 05704   0 01 05703      JMP   *-1
5827 05705   14 0104         OCP   ’104
5828 05706   74 0004         OTA   ’4
5829 05707   0 01 05706      JMP   *-1
5830 05710  -0 01 05702      JMP*  OUT1
5831                    *
5832 05711   0 000000   OUT2 DAC   **
5833 05712   0406 70         ARR   8
5834 05713   0 10 05702      JST   OUT1
5835 05714   0416 70         ALR   8
5836 05715   0 10 05702      JST   OUT1
5837 05716  -0 01 05711      JMP*  OUT2
5838                    *
5839                    * **** KEY ****
5840 05717   34 0104    PKEY SKS   ’104            SKIP IF NOT BUSY
5841 05720   0 01 05717      JMP   *-1
5842 05721   14 0004         OCP   ’4               SELECT INPUT MODE
5843 05722   54 1004         INA   ’1004            INPUT
5844 05723   0 01 05722      JMP   *-1
5845                         IFZ   ECLF
5846                         JMP   PUSH
5847                         ELSE
5848 05724   0 11 05734      CAS   KCCR
5849 05725   0 01 00116      JMP   PUSH
5850 05726   100000          SKP
5851 05727   0 01 00116      JMP   PUSH
5852 05730   0 02 05735      LDA   KCLF
5853 05731   0 10 05702      JST   OUT1
5854 05732   0 02 05734      LDA   KCCR
5855 05733   0 01 00116      JMP   PUSH
5856                    *
```

```
5857 05734   000215     KCCR VFD   16,CCR
5858 05735   000212     KCLF VFD   16,CLF
5859                         ENDC
5860                    *
5861                    * **** ?TERMINAL ****
5862 05736   34 0104    PQTR SKS   '104              SKIP IF NOT BUSY
5863 05737   0 01 05736      JMP   *-1
5864 05740   14 0004         OCP   '4                SELECT INPUT MODE
5865 05741   0 02 00736      LDA   =1
5866 05742   34 0004         SKS   '4                SKIP IF READY
5867 05743   140040         CRA
5868 05744   0 01 00116      JMP   PUSH
5869                    *
5870                    * **** CR ****
5871 05745   0 10 05750 PCR  JST   CRLF
5872                    *
5873                    *    NEXT
5874 05746   0 12 00100      IRS   IP
5875 05747  -0 01 00100      JMP*  IP
5876                    *
5877 05750   0 000000   CRLF DAC   **
5878 05751   0 02 05754      LDA   CRL1
5879 05752   0 10 05711      JST   OUT2
5880 05753  -0 01 05750      JMP*  CRLF
5881 05754   106612     CRL1 VFD   8,CCR,8,CLF
5882                    *
5883                    * PAPERTAPE ROUTINES
5884                    *
5885                         IFN   PTW
5886 05755   14 0001    PPTC OCP   '1                START READER
5887 05756   54 1001         INA   '1001             INPUT
5888 05757   0 01 05756      JMP   *-1
5889 05760   14 0101         OCP   '101              STOP READER
5890 05761   0 01 00116      JMP   PUSH
5891                         ENDC
5892                    *
5893                    *
5894                    * DEBUG
5895                    *
5896                    *
5897                         IFN   DBGW
5898                    OCTC BSS   1
5899                    *
5900                    OCTL DAC   **
5901                         IAB
5902                         LDA   =-6
5903                         STA   OCTC
5904                         CRA
5905                         LLR   1                 MSB ROTATES INTO A
5906                    *
```

```
5907                OCT1 ADD   XZRO
5908                     JST   OUT1
5909                     CRA
5910                     LLR   3
5911                     IRS   OCTC
5912                     JMP   OCT1
5913                     JMP*  OCTL
5914                *
5915                DSPC DAC   **
5916                     LDA   XSPC
5917                     JST   OUT1
5918                     JMP*  DSPC
5919                XSPC VFD   16,CSPC
5920                *
5921                PDBG LDA   IP
5922                     JST   OCTL
5923                *
5924                     JST   DSPC
5925                     LDA   0
5926                     JST   OCTL
5927                *
5928                     JST   DSPC
5929                     LDA   1,1
5930                     JST   OCTL
5931                *
5932                     JST   DSPC
5933                     LDA   2,1
5934                     JST   OCTL
5935                *
5936                     JST   DSPC
5937                     LDA   3,1
5938                     JST   OCTL
5939                *
5940                     JST   DSPC
5941                     LDA   4,1
5942                     JST   OCTL
5943                *
5944                     JST   DSPC
5945                     LDA   5,1
5946                     JST   OCTL
5947                *
5948                     JST   DSPC
5949                     LDA*  RP
5950                     JST   OCTL
5951                *
5952                     JST   CRLF
5953                *
5954                *
5955                *    NEXT
5956                     IRS   IP
```

```
5957                         JMP*  IP
5958                         ENDC
5959                    ****************************************************************
5960                    *
5961                    *  STACKS AND BUFFERS
5962                    *
5963                    ****************************************************************
5964                    *
5965                    * 'XTIB', 'XR0', AND 'XUP' ARE ONLY USED IN BOOT-UP TABLE;
5966                    *  THEREFORE THE AREAS DEFINED HERE CAN BE MOVED AT RUN TIME.
5967 05762              XTIB BSS   42               TERMINAL INPUT BUFFER
5968 06116              XR0  BES   50               FOR RETURN STACK
5969 06116              XUP  BSS   '100             ROOM FOR '100 USER VARIABLES
5970                    *
5971                    *  NOTE - 'UP', 'OPENF', 'INTERM', AND DISK BUFFERS ARE
5972                    *  INITIALIZED AT COLD START, OR AT FIRST TIME THROUGH.
5973                    *
5974                    *
5975                         IFN   DISK
5976                    *
5977                    * ROOM  FOR 3 1K DISK BUFFERS
5978                    *
5979                    * INITIALIZE BUFFERS' UPDATE BITS, AND TERMINATING NULLS, TO ZERO.
5980                    * NOTE - THESE BUFFERS ARE CLEARED AT COLD START, ANYWAY,
5981                    * BECAUSE A STAND-ALONE BOOT MAY NOT INITIALIZE HIGH MEMORY;
5982                    * AND ALSO SO THAT THE NUMBER OR LOCATION OF BUFFERS CAN BE
5983                    * CHANGED AT RUN TIME.
5984                    DSKB OCT   0
5985                         BSS   512
5986                         OCT   0
5987                         OCT   0
5988                         BSS   512
5989                         OCT   0
5990                         OCT   0
5991                         BSS   512
5992                         OCT   0
5993                    ENDB EQU   *                CAUTION - 'ENDB' - 'DSKB' MUST BE EXACT MULTIPLE
5994                    *                           OF THE BUFFER LENGTH PLUS 4.
5995                    *
5996                         ENDC
5997                    *
5998                    * DICTIONARY STARTS HERE
5999 06216              XDP  BSS   128              FOR DICTIONARY AND COMP. STACK
6000                    * THIS IS JUST NOMINAL - SEE MSZ, BELOW
6001 06416              XS0  BSS   2                START OF COMPUTATION STACK
6002                    * 2 WORDS IN CASE OF EMPTY STACK
6003                    *
6004                    ****************************************************************
6005                    *
6006                    *  START-UP CODE - CALCULATE TOP OF MEMORY
```

```
6007                    *
6008                    ****************************************************************
6009                         ORG   XTIB              DELIBERATELY OVERLAY
6010 05762   0 000000   MSZ  DAC   **
6011 05763   140040          CRA
6012 05764   140500          SSM                    JUST TOP BIT
6013                         IFZ   XTND
6014 05765   0404 77         LGR   1                DIVIDE BY 2
6015                         ENDC
6016                    * A POINTS ONE BEYOND MAXIMUM MEMORY
6017 05766   0 01 05772      JMP   MSZ2
6018                    *
6019 05767   0 02 00105 MSZ1 LDA   T2
6020 05770  -0 04 00104      STA*  T1               PUT ORIGINAL DATA BACK (JUST IN CASE)
6021 05771   0 02 00104      LDA   T1
6022 05772   0 07 06034 MSZ2 SUB   MS4K
6023 05773   100400          SPL                    SHOULD ALWAYS BE POSITIVE
6024 05774   000000          HLT                    TRAP NEVER-ENDING LOOP
6025 05775   0 04 00104      STA   T1
6026 05776  -0 02 00104      LDA*  T1
6027 05777   0 04 00105      STA   T2               SAVE ORIGINAL CONTENTS
6028 06000   140040          CRA
6029 06001  -0 04 00104      STA*  T1
6030 06002   140401          CMA                    ALL ONES
6031 06003  -0 02 00104      LDA*  T1
6032 06004   100040          SZE
6033 06005   0 01 05767      JMP   MSZ1             DIDN'T CLEAR
6034 06006   140401          CMA                    GET ALL ONES
6035 06007  -0 04 00104      STA*  T1
6036 06010  -0 12 00104      IRS*  T1               SHOULD SKIP
6037 06011   0 01 05767      JMP   MSZ1             DIDN'T SKIP
6038                    *
6039                    * HAVE MEMORY HERE...
6040 06012   0 02 00105      LDA   T2
6041 06013  -0 04 00104      STA*  T1               PUT ORIGINAL DATA BACK
6042 06014   0 02 00104      LDA   T1
6043 06015   0 06 06034      ADD   MS4K             JUST ABOVE MEMORY
6044 06016   0 07 06035      SUB   MSRV             WORDS TO RESERVE
6045 06017   0 07 00720      SUB   =2               TWO WORDS FOR EMPTY STACK
6046 06020   0 11 06036      CAS   MXDB
6047 06021   101000          NOP                    OK - GREATER
6048 06022   100000          SKP                    OK - EQUAL
6049 06023   000000          HLT                    NO SPACE FOR DICTIONARY
6050                    *
6051                    * SAVE CALCULATED TOP OF STACK
6052 06024   0 04 01007      STA   OXS0
6053                    *
6054                    * PATCH SO NEVER CALL AGAIN
6055 06025   0 02 05762      LDA   MSZ              RETURN ADDRESS
6056 06026   0 07 00736      SUB   =1               POINT AT JST
```

```
6057 06027   0 04 00104      STA   T1
6058 06030   101000          NOP
6059 06031   0 02 06030      LDA   *-1
6060 06032  -0 04 00104      STA*  T1
6061 06033  -0 01 05762      JMP*  MSZ
6062                     *
6063 06034   010000     MS4K DEC   4096
6064 06035   000000     MSRV VFD   16,RSRV
6065 06036   0 006316   MXDB DAC   XDP+'100          MINIMUM DICTIONARY SPACE
6066                     *
6067                          ORG   NXTY             FOR CONSTANT POOL
6068 00711   000021          FIN
     00712   000040
     00713   100000
     00714   000004
     00715   177777
     00716   177760
     00717   040000
     00720   000002
     00721   077400
     00722   077577
     00723   000077
     00724   000200
     00725   037577
     00726   000007
     00727   000011
     00730   000003
     00731   177773
     00732   177764
     00733   000177
     00734   177400
     00735   000377
     00736   000001

6069         000737     NXTZ EQU   *
6070                          END   ORGN

ABRT   004347A   ABS    004446A   ABS1   004454A   AGAN   005253A
ALOT   002250A   AND    001300A   AT     001565A   BACK   005125A
BASE   002152A   BBUF   001752A   BCMP   004032A   BDGS   005371A
BGIN   005134A   BINA   000124A   BL     001737A   BLK    002072A
BLKS   003320A   BRAN   000263A   BSCR   001760A   BULD   002712A
BYE    005631A   BYTE   001616A   CADR   000111A   CAT    001572A
CBS    000210A   CCR    000215A   CDEL   000377A   CDOT   000256A
CDQT   000242A   CELL   001625A   CENT   001025A   CEOT   000204A
CFA    002376A   CHGT   000166A   CHP1   000207A   CHPT   000176A
CHPU   000212A   CL     001744A   CLF    000212A   CMNS   000255A
CMOV   000641A   CMVL   000652A   CMVX   000663A   CNT    002735A
CNT1   001042A   CNT2   001046A   COLD   004427A   COLN   001632A
COMA   002256A   COMP   002570A   CON    001664A   CONT   002127A
```

CR	000627A	CRAT	003746A	CRL1	005754A	CRLF	005750A
CRPR	000251A	CRT1	003761A	CSP	002171A	CSPC	000240A
CSTR	001605A	CURR	002136A	CZRO	000260A	DAB1	004467A
DABS	004461A	DBGW	000000A	DDOT	005530A	DDTR	005475A
DDUP	002345A	DEC	002651A	DFN1	003605A	DFND	003562A
DFNS	004272A	DGS1	005462A	DIG	005433A	DIG1	005451A
DIGA	000425A	DIGS	005461A	DIGT	000407A	DIGX	000431A
DIGY	000416A	DISK	000000A	DIVU	001210A	DLIT	004067A
DLT1	004077A	DMNS	001502A	DO	005167A	DOCL	000130A
DOCN	000140A	DODS	000152A	DOES	002722A	DOT	005537A
DOTQ	003047A	DOTR	005517A	DOUS	000146A	DOVC	004257A
DOVR	000143A	DP	002041A	DPL	002157A	DPLS	001451A
DROP	001526A	DTQ1	003071A	DTQ2	003075A	DTR1	003003A
DTR2	003020A	DTR3	003022A	DTRA	002777A	DUP	001541A
DVS1	004577A	DVS2	004604A	DVU1	001232A	DVU2	001246A
DVU3	001252A	DVU4	001262A	DVU5	001264A	DVU6	001267A
ECHO	000000A	ECLF	000001A	EDGS	005402A	ELSE	005320A
EMIT	000610A	EMT1	005672A	EMT2	005675A	ENC1	000543A
ENC2	000550A	ENC3	000556A	ENC4	000561A	ENCC	000566A
ENCL	000534A	ENCX	000602A	END	005244A	ENDF	005145A
EQAL	002273A	ERAS	003307A	EROR	003662A	ERR1	003671A
EXEC	000254A	EXP1	003113A	EXP2	003123A	EXP3	003124A
EXP4	003151A	EXP5	003165A	EXP6	003166A	EXP7	003200A
EXPC	003103A	FENC	002034A	FILL	003267A	FIMD	000001A
FLD	002164A	FNDL	000445A	FNDM	000511A	FNDN	000525A
FNDS	000467A	FNDT	000475A	FNDX	000453A	FNDY	000521A
FNDZ	000507A	FNUL	000000A	FRGT	005066A	FRMR	001412A
FRST	002057A	FRTH	005652A	GO	001104A	GRTR	002315A
HERE	002240A	HEX	002636A	HLD	002203A	HOLD	003335A
HSA	000001A	I	000401A	IDDT	003712A	IF	005304A
IMMD	004222A	IN	002077A	IP	000100A	ITR1	004153A
ITR2	004167A	ITR3	004171A	ITR4	004174A	ITR5	004207A
ITR6	004211A	ITR7	004212A	ITRP	004152A	KCCR	005734A
KCLF	005735A	KEY	000614A	KPAD	000042A	LBRC	002603A
LEAV	001373A	LES1	002307A	LES2	002311A	LESS	002303A
LFA	002367A	LIMT	002065A	LINK	005657A	LIT	000241A
LIT1	004060A	LOOP	005202A	LTRL	004050A	LTST	002357A
MAX	004507A	MAX1	004516A	MES2	005040A	MES3	005046A
MESS	005027A	MIN	004473A	MIN1	004502A	MINS	001472A
MOD	004705A	MOVE	000671A	MOVL	000702A	MS4K	006034A
MSLA	004555A	MSMD	004737A	MSRV	006035A	MST1	004544A
MSTR	004523A	MSZ	005762A	MSZ1	005767A	MSZ2	005772A
MXDB	006036A	NEXT	000122A	NFA	002405A	NFA1	002407A
NUL1	003256A	NUL2	003260A	NUL3	003262A	NULL	003225A
NUM1	003527A	NUM2	003550A	NUM3	003555A	NUMB	003510A
NXT1	000136A	NXTW	000236A	NXTX	001115A	NXTY	000711A
NXTZ	000737A	O1ST	001017A	OCT	002662A	OFST	002120A
OLMT	001020A	ONE	001722A	ONEP	002223A	OR	001306A
ORGN	001000A	OTSK	001004A	OUP	001006A	OUT	002104A
OUT1	005702A	OUT2	005711A	OVER	001520A	OXS0	001007A

```
OXTB   001011A   PABT   003653A   PAD    003351A   PARN   004302A
PCR    005745A   PDTQ   003031A   PEMT   005665A   PFA    002423A
PFND   000440A   PKEY   005717A   PLOP   005216A   PLUS   001443A
PNM1   003443A   PNM2   003476A   PNM3   003501A   PNUM   003442A
POP    000113A   POP2   000112A   PORG   001767A   PPTC   005755A
PQTR   005736A   PREV   002216A   PSCD   002675A   PSTR   001546A
PTK1   004770A   PTK2   005002A   PTK3   005005A   PTK4   005021A
PTR    004755A   PTRC   000634A   PTRK   004767A   PTW    000001A
PUSH   000116A   PUT    000125A   QCMP   002472A   QCSP   002534A
QER1   002464A   QER2   002465A   QERR   002455A   QEXC   002506A
QLDG   002553A   QPRS   002522A   QSTK   004123A   QTRM   000623A
QUIT   004313A   QURY   003211A   QUST   005545A   QUT1   004320A
QUT2   004322A   QUT3   004341A   R      001416A   RBRC   002612A
RNUM   002176A   ROT    002325A   RP     000101A   RP1    000102A
RPET   005267A   RPOP   000227A   RPSH   000220A   RPST   001345A
RSRV   000000A   RZRO   002004A   SCR    002111A   SCSP   002443A
SEMC   005636A   SEMI   001647A   SGN1   005430A   SIGN   005420A
SLMD   004665A   SLSH   004675A   SMDG   002625A   SMIS   001361A
SPAT   001324A   SPC1   005362A   SPC2   005365A   SPCE   002335A
SPCS   005352A   SPST   001332A   SSD1   004627A   SSD2   004643A
SSDV   004607A   SSLA   004726A   SSMD   004715A   STAR   004655A
STAT   002144A   STOD   004434A   STOP   001056A   STOR   001577A
SUB    002266A   SWAP   001533A   SZRO   001777A   T1     000104A
T2     000105A   T3     000106A   T4     000107A   T5     000110A
TASK   005663A   THEN   005161A   THRE   001732A   TIB    002011A
TICK   005051A   TLNK   005657A   TOGL   001557A   TOR    001402A
TWO    001726A   TWOP   002231A   TYP1   002760A   TYP2   002767A
TYP3   002770A   TYPE   002750A   UDOT   005554A   ULES   004103A
UNTL   005232A   UP     000103A   UPPR   003612A   UPR1   003617A
UPR2   003643A   USE    002210A   USER   001710A   USLA   001205A
UST1   001150A   UST2   001152A   UST3   001157A   UST4   001161A
UST5   001164A   UST6   001172A   UST7   001173A   UST8   001174A
USTR   001120A   VAR    001700A   VCAB   004237A   VLS1   005574A
VLS2   005607A   VLST   005564A   VOCL   002051A   WARN   002026A
WDTH   002017A   WENT   001057A   WHIL   005341A   WNT1   001061A
WNT2   001067A   WORD   003363A   WRD1   003400A   WRD2   003402A
X4P4   001024A   XDO    000372A   XDP    006216A   XGO    001103A
XLL1   000324A   XLL2   000327A   XLL3   000337A   XLL4   000361A
XLOP   000322A   XOR    001316A   XPLO   000354A   XR0    006116A
XS0    006416A   XTIB   005762A   XTND   000000A   XTSK   005657A
XUP    006116A   XWNT   001102A   XXS0   001023A   XXVC   005656A
XZRO   000424A   ZBR1   000312A   ZBRA   000277A   ZEQ0   001430A
ZEQ1   001426A   ZEQU   001423A   ZERO   001716A   ZLES   001435A

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2    REV. C    01-26-71
```