fig-FORTH for PDP-11

ASSEMBLY SOURCE LISTING

WITH COMPILER SECURITY

AND

VARIABLE LENGTH NAMES

VERSION 1.3

JANUARY 1980

Provided through the courtesy of the

P.O. Box 8231 San Jose, CA 95155 (408)277-0668

```
                        .TITLE F. I. G.
                        ;*****************************************************************
                        ;
                        ; PDP-11 FORTH                INTRODUCTION                PDP-11 FORTH
                        ;
                        ;*****************************************************************
                        ;
                        ;
                        ;
                        ;  PDP-11 FORTH      RT-11, RSX-11M, AND STAND-ALONE      JANUARY 1980
                        ;
                        ;
                        ;
                        ;       DEVELOPED BY THE
                        ;               FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
                        ;               P.O. BOX 1105
                        ;               SAN CARLOS, CA. 94070
                        ;
                        ;
                        ;       IMPLEMENTED BY
                        ;               JOHN S. JAMES
                        ;               P.O. BOX 348
                        ;               BERKELEY, CA. 94701
                        ;
                        ;
                        ;       THIS SYSTEM IS IN THE PUBLIC DOMAIN AND CAN BE USED
                        ;       WITHOUT RESTRICTION.  PLEASE CREDIT THE FORTH INTEREST
                        ;       GROUP IF YOU REPUBLISH SUBSTANTIAL PORTIONS.
                        ;
                        ;
                        ;       THE FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
                        ;       ALSO HAS DEVELOPED NEARLY IDENTICAL VERSIONS OF THIS
                        ;       SYSTEM FOR THE
                        ;                       8080        6502
                        ;                       6800        6809
                        ;                       PACE        ALPHA MICRO
                        ;
                        ;
                        ;
                        ;       FOR MORE INFORMATION, WRITE:
                        ;
                        ;               JOHN S. JAMES
                        ;               P.O. BOX 348
                        ;               BERKELEY, CA. 94701
                        ;
                        ;                       OR
                        ;
                        ;               FORTH INTEREST GROUP
                        ;               P.O. BOX 1105
                        ;               SAN CARLOS, CA. 94070
                        ;
                        ;
                        ;       'PDP' AND 'RSX' ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION.
```

NOTE: This system is available on diskette from the author, John James
PO Box 348
Berkeley, CA 94701
(415) 526-8815

```
                ;
                ;       THIS FORTH SYSTEM HAS
                ;               - FULL LENGTH NAMES
                ;               - EXTENSIVE COMPILE-TIME CHECKS AND ERROR MESSAGES
                ;               - DOUBLE INTEGER I/O
                ;               - A FORTH ASSEMBLER, PERMITTING STRUCTURED, INTERACTIVE
                ;                 DEVELOPMENT OF DEVICE HANDLERS, SPEED-CRITICAL
                ;                 ROUTINES, AND LINKAGE TO OPERATING SYSTEMS OR TO
                ;                 SUBROUTINE PACKAGES WRITTEN IN OTHER LANGUAGES.
                ;               - STRING-HANDLING ROUTINES
                ;               - A STRING-SEARCH EDITOR
                ;               - LINKED VOCABULARIES
                ;               - HOOKS FOR MULTITASKING/MULTIUSER (CURRENTLY SINGLE TASK)
                ;               - AND AS CURRENTLY CONFIGURED IT RUNS IN A 24K BYTE
                ;                 TASK IMAGE (THIS INCLUDES BUFFERS, OPERATING-SYSTEM
                ;                 AREA, AND ROOM FOR SUBSTANTIAL ADDITIONAL FORTH
                ;                 PROGRAMMING) ON ANY PDP-11 OR LSI-11 CPU, WITH OR
                ;                 WITHOUT HARDWARE MULTIPLY/DIVIDE.  THIS DISKETTE
                ;                 WILL BOOT AND RUN STAND-ALONE; ALSO IT CONTAINS A
                ;                 SOURCE PROGRAM WHICH CAN BE ASSEMBLED TO RUN
                ;                 UNDER RT-11, RSX-11M, OR STAND-ALONE.  THIS SYSTEM
                ;                 CAN BE MODIFIED TO INTERFACE WITH ANY OTHER OPERATING
                ;                 SYSTEM WHICH SUPPORTS READ AND WRITE A CHARACTER,
                ;                 DETECT A CHARACTER (OPTIONAL), AND READ AND WRITE
                ;                 A DISK BLOCK.
                ;
                ;
                ;       IT IS ALIGNED WITH THE 1978 STANDARD OF THE FORTH INTERNATIONAL
                ;       STANDARDS TEAM.
                ;
                ;
                ;
                ;       RECOMMENDED DOCUMENTATION:
                ;               - A FORTH LANGUAGE MANUAL.  WE PARTICULARLY RECOMMEND EITHER
                ;                       (A) 'USING FORTH', BY FORTH, INC.
                ;                                       OR
                ;                       (B) 'A FORTH PRIMER', BY W. RICHARD STEVENS, KITT
                ;                           PEAK NATIONAL OBSERVATORY.
                ;                 EITHER IS AVAILABLE THROUGH THE FORTH INTEREST GROUP,
                ;                 P.O. BOX 1105, SAN CARLOS, CA. 94070.
                ;               - PDP-11 FORTH USER'S GUIDE, AVAILABLE FROM JOHN S. JAMES,
                ;                 ADDRESS ABOVE. Also available from FIG.
                ;               - FORTH REFERENCE CARD FOR THE FORTH IMPLEMENTATION TEAM
                ;                 COMMON MODEL, AVAILABLE FROM FIG.
                ;               - 'FIG-FORTH INSTALLATION MANUAL', ALSO FROM FIG.
                ;
                ;
                ;
                ;       ACKNOWLEDGMENTS:
                ;               THIS FORTH SYSTEM (IN 'FORTH.MAC') IS A GROUP PRODUCT
                ;               OF THE FORTH IMPLEMENTATION TEAM OF THE FORTH INTEREST
                ;               GROUP (P.O. BOX 1105, SAN CARLOS CA. 94070).  THE IMPLEMENTER
                ;               IS RESPONSIBLE FOR THIS PDP-11 VERSION OF THE MODEL, AND FOR
                ;               THE SOFTWARE ON SCREENS IN 'FORTH.DAT'.  ALTHOUGH THE LATTER
                ;               IS NOT AN OFFICIAL RELEASE OF THE F.I.G., THE CONTRIBUTIONS
                ;               FROM MEMBERS OF THE GROUP ARE TOO NUMEROUS TO CITE
                ;               INDIVIDUALLY.
                ;                    IN ADDITION WE APPRECIATE THE PDP-11 CODING
```

```
113                             ;       IMPROVEMENTS SUGGESTED BY STUART R. DOLE, DOLE & FARMER,
114                             ;       PO BOX 142, PETALUMA, CA. 94952; BY PAUL EDELSTEIN;
115                             ;       BY RICK STEVENS OF KITT PEAK; AND OTHERS.
```

```
                        ;*****************************************************************
                        ;
                        ;       BRINGING UP THE SYSTEM
                        ;
                        ;*****************************************************************
                        ;
                        ;
                        ;
                        ; TO RUN STAND-ALONE:
                        ;       - BOOT THE DISKETTE LIKE ANY OTHER SYSTEM DISK, FROM DX0.
                        ;         FORTH SHOULD COME UP AND TYPE 'FIG FORTH' AND THE VERSION
                        ;         NUMBER.  TEST AS DESCRIBED FOR RT-11 BELOW.
                        ;       - MAKE A COPY OF THE DISK; THIS STAND-ALONE SYSTEM DOES NOT
                        ;         PROTECT AGAINST ACCIDENTALLY OVERWRITING THE SYSTEM OR THE
                        ;         SOURCE PROGRAMS.  TO MAKE AN EXACT COPY OF THE ENTIRE DISK,
                        ;           1. PUT A BLANK DISK INTO THE SECOND DRIVE (DX1).  FOR
                        ;              SAFETY, SET THE WRITE-PROTECT SWITCH ON THE DRIVE
                        ;              WHICH CONTAINS THE ORIGINAL SYSTEM DISK.
                        ;           2. TYPE '38 LOAD', AND CARRIAGE RETURN.  THE SYSTEM SHOULD
                        ;              RESPOND 'OK'.  THEN TYPE 'COPY' AND RETURN.  EACH OF
                        ;              THE 77 TRACKS WILL BE READ FROM DX0 AND WRITTEN ON DX1.
                        ;       - NOTE THE LAYOUT OF THE DISKETTE.  IT IS SET UP TO BOOT AND
                        ;         RUN STAND-ALONE, BUT IT ALSO CONTAINS AN RT-11 DIRECTORY,
                        ;         AND A MACRO-11 SOURCE PROGRAM 'FORTH.MAC' (WHICH PRODUCED
                        ;         THIS LISTING).  THIS ALLOWS THE SAME DISK TO BE BOOTED
                        ;         AND RUN, OR TO PROVIDE SOURCE FOR MODIFICATION AND RE-ASSEMBLY.
                        ;              AS PROVIDED, THE FILE 'FORTH.DAT' CONTAINS FORTH SCREENS
                        ;         1-70.  YOU CAN USE LOCATIONS BEYOND 70, BUT THESE WILL
                        ;         OVERWRITE THE 'FORTH.MAC' SOURCE PROGRAM.  STAND-ALONE USERS
                        ;         MAY NEVER NEED TO USE THIS SOURCE, AND MAY WANT TO REMOVE IT
                        ;         AND USE THE SPACE FOR SOMETHING ELSE.  MAKE A COPY FIRST.
                        ;       - STAND-ALONE USERS CAN ADD THEIR OWN OPERATIONS AND THEN
                        ;         SAVE A BOOTABLE IMAGE OF THE NEW SYSTEM.  THE NEW OPERATIONS
                        ;         WILL BE AVAILABLE WHEN THE DISK IS BOOTED IN THE FUTURE.
                        ;         THE LOADER WHICH IS USED WILL ONLY LOAD IMAGES UP TO 7.9K;
                        ;         THIS LEAVES SEVERAL HUNDRED BYTES FOR NEW OPERATIONS, WHICH
                        ;         CAN INCLUDE EXTENDING THE SYSTEM BY BRINGING IN SOURCE OR
                        ;         OBJECT CODE.  TO SAVE THE CURRENT SYSTEM, EXECUTE 'FORTH DEFINITIONS'
                        ;         IF NECESSARY TO GET INTO THE FORTH VOCABULARY, THEN 'DECIMAL 34 LOAD'
                        ;         SOME WARNING MESSAGES WILL BE PRINTED (MSG #4); THEY CAN BE
                        ;         IGNORED.
                        ;       - IF YOU DO WANT TO RE-ASSEMBLE THE SYSTEM FOR STAND-ALONE
                        ;         USE (WHICH MOST USERS SHOULD NEVER FIND NECESSARY),
                        ;         YOU MUST USE RT-11 TO EDIT AND ASSEMBLE 'FORTH.MAC'.  NOTE
                        ;         THAT ALTHOUGH THIS LISTING IS ASSEMBLED FOR STAND-ALONE,
                        ;         THE SOURCE PROGRAM SUPPLIED IS SET FOR RT-11 ASSEMBLY;
                        ;         COMMENT OUT THE 'RT11' DEFINITION, AND REMOVE THE COMMENTING
                        ;         ON 'ALONE'.  ASSEMBLE, LINK, AND RUN, AND THE SYSTEM SHOULD
                        ;         COME UP STAND-ALONE.  IMMEDIATELY REMOVE THE RT-11 SYSTEM DISK
                        ;         AND PLACE THE FORTH DISK IN DRIVE ZERO.  TO REVISE
                        ;         THE BOOTABLE IMAGE ON THE FORTH DISK SO THAT YOUR NEW SYSTEM
                        ;         BOOTS STAND-ALONE, LIST SCREEN 34 (DECIMAL), AND FOLLOW THE
                        ;         INSTRUCTIONS THERE.  THE RUN TAKES ABOUT ONE MINUTE.
                        ;       - THE BOOTABLE SYSTEM DOES NOT USE HARDWARE MULTIPLY AND DIVIDE.
                        ;         IF YOU DON'T HAVE RT-11 TO EDIT AND RECOMPILE WITH 'EIS'
                        ;         CONDITIONAL ASSEMBLY, THE MULTIPLY/DIVIDE ROUTINES CAN BE
                        ;         PATCHED.  IF YOU PATCH FROM THE KEYBOARD MONITOR, THE
```

```
174                             ;         RESTART ADDRESS IS 1000 OCTAL ( COLD START) OR 1004 (WARM
175                             ;         START).  SAVE THE NEW VERSION AS A BOOTABLE SYSTEM, AS
176                             ;         DESCRIBED ABOVE.
177                             ;       - THE SKEWED DISK I/O OPERATIONS SKIP TRACK ZERO, FOR COMPATIBILITY
178                             ;         WITH STANDARD PDP-11 SECTOR SKEWING.  THE PHYSICAL READ
179                             ;         OPERATIONS ('RTS', 'WTS', 'NRTS', 'NWTS') CAN READ ANY SECTOR,
180                             ;         HOWEVER.
181                             ;       - ALSO THE SYSTEM AS DISTRIBUTED SKIPS THE FIRST 56 SECTORS
182                             ;         (7 SCREENS) IN ORDER TO SKIP THE BOOT BLOCK AND AN
183                             ;         RT-11 DIRECTORY.  THIS CAUSES THE SCREEN POSITIONS TO BE THE
184                             ;         SAME FOR STAND-ALONE AND FOR RT-11 (WHICH ACCESSES THE FILE
185                             ;         'FORTH.DAT').  YOU CAN CHANGE THIS BY CHANGING THE VALUE OF
186                             ;         THE VARIABLES 'S-SKIP' (NUMBER OF SCREENS SKIPPED) AND
187                             ;         'S-USE' (NUMBER OF SCREENS USED BEFORE ACCESSING THE
188                             ;         SECOND DISK).  THESE VARIABLES CAN BE CHANGED AT ANY TIME,
189                             ;         SO DISK SCREENS CAN BE READ INTO BUFFERS AND THEN FLUSHED
190                             ;         TO DIFFERENT LOCATIONS ON THE DISK.
191                             ;       - ADVANCED USERS MAY NOTE THAT THIS SYSTEM IS DESIGNED TO
192                             ;         ALLOW THE MEMORY LAYOUT - NUMBER AND LOCATION OF DISK
193                             ;         BUFFERS, LOCATION OF THE STACK, ETC. - TO BE CHANGED
194                             ;         DYNAMICALLY, WITHOUT REASSEMBLY.
195                             ;
196                             ;
197                             ;  TO BRING UP THIS SYSTEM UNDER RT-11:
198                             ;       - BE SURE THAT RT-11 IS SELECTED BELOW.  THE LINES DEFINING
199                             ;         'RSX11M' AND 'ALONE' SHOULD BE COMMENTED OUT; 'RT11' SHOULD
200                             ;         NOT BE.  NOTE THAT THIS DISK IS DISTRIBUTED READY FOR RT-11
201                             ;         ASSEMBLY (EVEN THOUGH THIS LISTING IS FOR STAND-ALONE).
202                             ;       - IF YOU HAVE HARDWARE MULTIPLY/DIVIDE, ALSO REMOVE THE
203                             ;         SEMICOLON FROM THE LINE DEFINING 'EIS'.
204                             ;       - IF YOU ARE USING AN OLDER VERSION OF RT-11 (VERSION 2),
205                             ;         YOU MAY NEED TO USE THE MACROS '..V2..' AND '.REQDEF'.
206                             ;       - ASSEMBLE, LINK, AND RUN.  THE SYSTEM SHOULD COME UP AND
207                             ;         TYPE 'FIG-FORTH' AND THE VERSION NUMBER.
208                             ;       - TEST THAT IT IS UP BY TRYING SOME ARITHMETIC OR DEFINITIONS, E.G.
209                             ;               88 88 * .                   (NOTE THAT THE '.' MEANS PRINT)
210                             ;               : SQUARE DUP * ;
211                             ;               25 SQUARE .
212                             ;         OR TYPE 'VLIST' FOR A LIST OF ALL THE FORTH OPERATIONS IN THE
213                             ;         DICTIONARY.
214                             ;       - THE DISK SHOULD WORK IF THE DISKETTE IS IN DRIVE 'DK'.
215                             ;         MAKE SURE THAT 'DK' IS ASSIGNED TO WHATEVER  PHYSICAL
216                             ;         DRIVE YOU ARE USING - OR CHANGE LINE 'RTFILE:' IN
217                             ;         'FORTH.MAC'.  TEST THE DISK BY TYPING
218                             ;               1 LIST
219                             ;         WHICH SHOULD LIST THE SCREEN WHICH LOADS THE EDITOR,
220                             ;         ASSEMBLER, AND STRING ROUTINES.
221                             ;       - IN CASE YOU NEED TO GET A LISTING FROM THE ASSEMBLY OF
222                             ;         'FORTH.MAC' (NOT USUALLY NECESSARY), AND YOUR SYSTEM HAS
223                             ;         ONLY DISKETTES (NO LARGER DISKS), THE 'ALLOCATE' OPTION
224                             ;         IS NECESSARY BECAUSE OF THE SIZE OF THE '.LST' FILE
225                             ;         (AROUND 230 BLOCKS).  FIRST COPY 'FORTH.MAC' ONTO A
226                             ;         SEPARATE DISKETTE BY ITSELF.  THEN EXECUTE
227                             ;               .MACRO /LIST:FORTH.LST /ALLOCATE:300. /NOOBJECT
228                             ;         AND REPLY 'FORTH.MAC' WHEN ASKED FOR 'FILES?'.
229                             ;
230                             ;
```

```
231                             ;
232                             ;  TO BRING UP THE SYSTEM UNDER RSX-11M:
233                             ;       - THE DISKETTE PROVIDED IS IN RT-11 FILE FORMAT.  THE TWO FILES
234                             ;         MUST BE COPIED OFF THE DISKETTE INTO AN RSX DIRECTORY.  THE
235                             ;         'FORTH.DAT' FILE MUST BE COPIED IN IMAGE MODE.  ANY RSX
236                             ;         DIRECTORY MAY BE USED.  ASSUMING THE DISKETTE IS IN DRIVE 0,
237                             ;         USE THE RSX COMMANDS:
238                             ;               >FLX =DX:FORTH.MAC/RT
239                             ;               >FLX =DX:FORTH.DAT/RT/IM
240                             ;         INCIDENTALLY, 'FORTH.DAT' IS THE SYSTEM'S 'VIRTUAL MEMORY'
241                             ;         FILE, USED FOR DISK I/O.  THE REST OF THE SYSTEM (THIS
242                             ;         PROGRAM ALONE) CAN RUN INDEPENDENTLY, EVEN IF 'FORTH.DAT'
243                             ;         IS NOT AVAILABLE.
244                             ;       - EDIT 'FORTH.MAC' TO SELECT RSX ASSEMBLY.  CHANGE THE SEMICOLON
245                             ;         TO COMMENT OUT 'RT11' NOT 'RSX11'.  LET 'EIS' BE DEFINED IF
246                             ;         YOU HAVE HARDWARE MULTIPLY/DIVIDE.
247                             ;       - ASSEMBLE, TASK BUILD, AND RUN.  TEST AS WITH RT11 ABOVE.
248                             ;       - THE DISK I/O SHOULD WORK IF 'FORTH.DAT' IS IN THE DEFAULT
249                             ;         DEVICE AND DIRECTORY.  TEST AS ABOVE.
250                             ;
251                             ;
252                             ;
253                             ;  THE SYSTEM AS SUPPLIED RESERVES 8000. BYTES FOR YOUR FORTH
254                             ;  PROGRAMMING AND STACK.  THIS IS ENOUGH FOR SUBSTANTIAL PROJECTS.
255                             ;  (NOTE THAT THE EDITOR, ASSEMBLER, AND STRING PACKAGE, IF LOADED,
256                             ;  USE MORE THAN 5K OF THIS.)  TO CHANGE THIS MEMORY SIZE, CHANGE
257                             ;  THE '8000.' WHICH IS IN THE LINES FOLLOWING THE LABEL 'DP:',
258                             ;  NEAR THE END OF THIS PROGRAM.  INCIDENTALLY, VERY FEW JOBS
259                             ;  (E.G. RECURSION) WILL EVER USE MORE THAN 100 WORDS OF THIS SPACE
260                             ;  FOR THE STACK; THE REST OF THE SPACE IS AVAILABLE FOR A STRING
261                             ;  STACK (IF USED) OR FOR YOUR PROGRAMS - AND FORTH OBJECT CODE IS
262                             ;  CONSIDERABLY MORE COMPACT THAN ASSEMBLY.
263                             ;
264                             ;
265                             ;
266                             ;  THE FORTH VIRTUAL FILE 'FORTH.DAT' IS USED FOR STORING SOURCE
267                             ;  PROGRAMS (OR DATA).  THIS FILE HAS 70 1-K SCREENS (1-70),
268                             ;  I.E. 140 PDP-11 DISK BLOCKS.  SCREENS 4 AND 5 ARE USED BY THE
269                             ;  SYSTEM FOR STORING ERROR AND WARNING MESSAGES.  SCREENS 6-30
270                             ;  CONTAIN A TEXT EDITOR, ASSEMBLER, STRING PACKAGE, AND MISCELLANEOUS
271                             ;  EXAMPLES.  SCREENS 40 THROUGH 47 CONTAIN A BINARY STAND-ALONE
272                             ;  SYSTEM (NOT USED UNDER RT-11 OR RSX-11M).  USERS MAY WANT
273                             ;  TO SAVE THEIR SOURCE PROGRAMS AND DATA IN THE BLANK SCREENS.
274                             ;  THE SIZE OF THIS FORTH SCREENS FILE ('FORTH.DAT') CAN BE INCREASED
275                             ;  IF NEEDED.  IF THE SYSTEM IS TO BE BOOTED STAND-ALONE, THE LOCATION
276                             ;  OF THE SYSTEM BINARY IMAGE ON THE DISK MUST NOT BE CHANGED;
277                             ;  THEREFORE, IF THE DISK IS TO BE USED TO RUN STAND-ALONE, DO NOT
278                             ;  USE RT-11 TO MOVE 'FORTH.DAT' TO ANOTHER PLACE ON THE DISK.
279                             ;
280                             ;
281                             ;
282                             ;
283                             ;
284                             ;  NOTE THAT THE RT-11 AND RSX-11M SYSTEMS DO NOT ECHO CHARACTERS
285                             ;  WHICH ARE INPUT FROM THE TERMINAL.  INSTEAD, THEY LET THE OPERATING
286                             ;  SYSTEM (RT-11 OR RSX-11M) ECHO THEM.  THIS IS DONE SO THAT TYPING
287                             ;  CONVENTIONS WILL BE THE SAME AS THE USER IS FAMILIAR WITH.  ALSO,
```

```
                        ;  TO AVOID SWAPPING DELAYS, THE RSX VERSION OF 'KEY' READS A LINE OF
                        ;  CHARACTERS AT A TIME.
                        ;
                        ;
                        ;
                        ;  CHANGE THESE LINES TO CONTROL CONDITIONAL ASSEMBLY:
                        ;
                        ;RT11=1          ; COMMENTED OUT UNLESS RT-11
                        ;RSX11=1         ; COMMENTED OUT UNLESS RSX11M
000001                  ALONE=1          ; COMMENTED OUT UNLESS STAND-ALONE
                        ;EIS=1           ; COMMENTED OUT UNLESS HARDWARE MULTIPLY-DIVIDE
                        ;LINKS=1         ; COMMENTED OUT UNLESS SUBROUTINE LINKAGE FROM
                        ;                       FORTH TO OTHER LANGUAGES
                        ;
                        ;.PAGE
                        ; ****************************************************************
                        ;
                        ;               VARIATIONS FROM F. I. G.  MODEL
                        ;
                        ; ****************************************************************
                        ;
                        ;
                        ;  'FIRST' AND 'LIMIT' HAVE BEEN MADE USER VARIABLES, NOT CONSTANTS.
                        ;  THEREFORE WHEN THEY ARE USED, 'FIRST @' AND 'LIMIT @' ARE
                        ;  REQUIRED.
                        ;
                        ;  ';CODE' AND 'FORTH' ARE NOT PURE CODE, SO THEY WERE MOVED TO THE
                        ;  END OF THE DICTIONARY.  THIS IS SO THE BULK OF THE DICTIONARY
                        ;  COULD BE PUT IN PROM OR USED RE-ENTRANTLY.
                        ;
                        ;  THE MACHINE-INDEPENDENT I/O SECTION WAS MOVED TO NEAR THE END OF
                        ;  THE DICTIONARY, BECAUSE IT IS NOT ALWAYS PURE CODE, AND ALSO TO
                        ;  ALLOW THE I/O TO BE REDEFINED WITHOUT REASSEMBLY.
                        ;
                        ;  THIS SYSTEM MUST TEST FOR FIRST-TIME-THROUGH TERMINAL AND DISK
                        ;  I/O, TO AVOID ERRONEOUS ATTEMPT TO OPEN FILES TWICE AT LATER COLD
                        ;  STARTS.  IT CLEARS DISK BUFFERS AT COLD START.
                        ;
```

```
                        ;  ***************************************************************
                        ;
                        ;               SET UP REGISTERS AND MACROS.
                        ;
                        ;  ***************************************************************
                        ;
                        ;
000002                  W=%2            ; TEMPORARY USED BY 'NEXT' MACRO (THE INNER INTERPRETER)
000003                  U=%3            ; POINTER TO THE USER AREA
000004                  IP=%4           ; FORTH INSTRUCTION COUNTER
000005                  S=%5            ; FORTH STACK POINTER
000006                  RP=SP           ; FORTH RETURN-STACK POINTER
                        ;
                        ;  NOTE - CODE ROUTINES CAN USE REGISTERS 0, 1, 4, AND 5, WITHOUT
                        ;  RESTORING THEM.
                        ;
                        ;
                        ;  MACRO DEFINITIONS
                        ;
                        ;
                        ;
                        ;  THE 'HEAD' MACRO CREATES A FORTH DICTIONARY HEADER.  ITS ARGUMENTS ARE:
                        ;   (1) LENGTH BYTE - THE LENGTH OF THE NAME BEING DEFINED.  THE SIGN BIT
                        ;       OF THE LENGTH BYTE MUST BE SET, SO THAT THE SYSTEM WILL RECOGNIZE
                        ;       THE END OF A VARIABLE-LENGTH NAME FIELD; THEREFORE THE LENGTH BYTE
                        ;       IS GIVEN AS 200 OCTAL PLUS THE LENGTH.  IF THE OPERATION IS
                        ;       IMMEDIATE, THE BIT NEXT TO THE SIGN BIT IS ALSO SET, SO THE LENGTH
                        ;       BYTE IS GIVEN AS 300 OCTAL PLUS THE LENGTH.
                        ;   (2) NAME - THE NAME OF THE OPERATION BEING DEFINED.
                        ;   (3) LCHAR - THE ASCII VALUE OF THE LAST CHARACTER OF THE NAME, WITH THE
                        ;       SIGN BIT SET.  THE NAME FIELD MUST HAVE AN EVEN LENGTH (INCLUDING
                        ;       THE LENGTH BYTE), SO IF THE NUMBER OF CHARACTERS IN THE NAME IS
                        ;       EVEN, 'LCHAR' WILL BE GIVEN AS 240 (200 PLUS CODE FOR A SPACE).
                        ;   (4) LABEL - THE ASSEMBLY-LANGUAGE LABEL ASSOCIATED WITH THE 'CODE FIELD'
                        ;       OF THIS DICTIONARY HEADER.  THESE LABELS ARE USED IN THE PRECOMPILED-
                        ;       FORTH SECTION OF THE SYSTEM.  WHEN POSSIBLE, THE FORTH OPERATION
                        ;       NAME ITSELF IS USED AS THE ASSEMBLY LABEL; OTHERWISE AN ABBREVIATION
                        ;       IS USED.  BY CONVENTION, THESE NAMES ARE LIMITED TO FIVE CHARACTERS,
                        ;       FOR CONSISTENCY AMONG VARIOUS ASSEMBLERS FOR DIFFERENT MICROPROCESSORS.
                        ;       (THE FORTH IMPLEMENTATION TEAM USES THE SAME LABELS IN ALL OF ITS
                        ;       VERSIONS.)
                        ;   (5) CODE - POINTER TO THE MACHINE-LANGUAGE "CODE ROUTINE" ASSOCIATED
                        ;       WITH THIS OPERATION TYPE OR DATA TYPE.  E.G. FOR ANY COLON DEFINITION,
                        ;       THIS ARGUMENT IS 'DOCOL', THE LABEL OF A FIVE-INSTRUCTION ASSEMBLY
                        ;       ROUTINE WHICH USES THE RETURN STACK TO HANDLE THE NESTED EXECUTION
                        ;       OF ANOTHER LEVEL OF FORTH OPERATIONS.  FOR ANY CONSTANT, THIS CODE
                        ;       ROUTINE IS 'DOCON', AND SIMILARLY FOR ALL OTHER DATA TYPES.
                        ;               THE CODE ARGUMENT MAY BE OMITTED.  IN THAT CASE, THE 'HEAD'
                        ;       MACRO LEAVES THE CODE FIELD POINTING TWO BYTES BEYOND ITSELF, WHERE
                        ;       MACHINE-LANGUAGE CODE MUST BEING - AND THE OPERATION SO DEFINED IS
                        ;       CALLED A "PRIMITIVE".  THE "NUCLEUS SECTION" OF THIS VERSION OF
                        ;       FORTH CONTAINS ABOUT 45 PRIMITIVES, FROM WHICH THE WHOLE SYSTEM
                        ;       IS BUILT; IN EFFECT, THESE PRIMITIVES DEFINE THE VIRTUAL FORTH
                        ;       MACHINE.  (A FEW OPERATIONS IN THE "PRECOMPILED FORTH" SECTION
                        ;       OF THE SYSTEM HAVE BEEN REPLACED WITH PRIMITIVES, TO OPTIMIZE
                        ;       EXECUTION SPEED.  AND WHEN A FORTH ASSEMBLER IS ADDED TO THIS
                        ;       SYSTEM, USERS WILL BE ABLE TO DEFINE THEIR OWN PRIMITIVES DIRECTLY
```

```
                        ;       IN FORTH,  IMMEDIATELY READY FOR EXECUTION. )
                        ;
                        ;  THE 'HEAD' MACRO CREATES A FORTH HEADER CONSISTING OF
                        ;       LENGTH BYTE - SIGN BIT SET
                        ;       NAME OF THE OPERATION - VARIABLE LENGTH - SIGN BIT SET ON LAST CHAR.
                        ;       LINK FIELD,  WHICH POINTS TO THE BEGINNING OF THE PREVIOUS DICTIONARY
                        ;               HEADER (USED AT COMPILE TIME)
                        ;       CODE POINTER.
                        ;
000000                  LINK=0          ; LAST LINK FIELD IS 0,  INDICATING END OF THE DICTIONARY.

                        ;
                                .MACRO  HEAD,LENGTH,NAME,LCHAR,LABEL,CODE
                        LINK2=.
                                .BYTE   LENGTH
                                .ASCII  ^NAME^
                                .EVEN
                        .=.-1
                                .BYTE   LCHAR   ; LAST CHARACTER OF NAME (OR BLANK FILL),
                                        ;  PASSED IN OCTAL,  WITH HIGH BIT SET.
                                .WORD   LINK
                        LINK=LINK2
                        LABEL:  .IF NB CODE
                                .WORD   CODE
                                .IFF
                                .WORD   .+2
                                .ENDC
                                .ENDM
                        ;
                        ;
                        ;
                        ;  THE 'NEXT' MACRO TRANSFERS CONTROL FROM ONE FORTH OPERATION TO THE
                        ;  'CODE ROUTINE' OF THE NEXT.  NOTICE THAT ONLY TWO INSTRUCTION
                        ;  EXECUTIONS ARE REQUIRED TO TRANSFER CONTROL FROM USEFUL OPERATIONS
                        ;  OF ONE FORTH PRIMITIVE TO THOSE OF THE NEXT.
                        ;
                                .MACRO  NEXT
                                MOV     (IP)+,W
                                JMP     @(W)+
                                .ENDM
                        ;
                        ;
                        ;  MACRO CALLS
                        ;
                        ;
                                .IFDF   RT11
                                .MCALL  .RCTRLO,.TTYIN,.TTINR,.TTYOUT,.EXIT,.TRPSET
                                .MCALL  .SETTOP,.DSTATUS,.FETCH,.LOOKUP,.READW,.WRITW
                                .ENDC
                        ;
                        ;
                                .IFDF   RSX11
                                .MCALL  GIOW$C,EXIT$S,ALUN$C,ASTX$S,SVTK$S
                                .MCALL  FDBDF$,FDRC$A,FDBK$A,FDOP$A,FSRSZ$
                                .MCALL  OPEN$M,READ$,WRITE$,WAIT$,CLOSE$
                                .MCALL  GIOW$
                                .ENDC
```

```
                                ;*****************************************************************
                                ;
                                ;               START-UP TABLE
                                ;
                                ;*****************************************************************
                                ;
                                ;  AT STARTUP, MOST OF THESE VALUES ARE MOVED INTO THE USER AREA
                                ;  (STARTING AT 'XDP:'); THEY ARE NORMALLY ACCESSED THERE.  THE VALUES
                                ;  HERE ARE  NOT USUALLY CHANGED, BUT THEY MAY BE CHANGED E.G. TO
                                ;  CONTROL WHAT HAPPENS AT COLD START.  THIS TABLE COULD BE MOVED OUT OF
                                ;  LOW MEMORY IF NECESSARY FOR ROM SYSTEMS.
                                ;
                                ;
                                ;
000000                          GFORTH::                        ; GLOBAL LABEL - NORMALLY NOT USED
000000  000167  007724          ORIGIN: JMP     CENT            ; COLD START ENTRY POINT
000004  000167  010002                  JMP     WENT            ; WARM START ENTRY ADDRESS
                                ;  NOTE - COLD START WIPES OUT ANY NEW DICTIONARY DEFINITIONS, AND
                                ;  THEN DOES A WARM START.  WARM START CLEANS UP STACKS, TERMINAL
                                ;  BUFFER, ETC.
000010  000011                        .WORD   11              ; CPU
000012  000000                        .WORD   0               ; REVISION
000014  015412'                        .WORD   TASK-10         ; POINTER TO LATEST WORD DEFINED
000016  000010                        .WORD   10              ; BACKSPACE CHARACTER
000020  043436'                        .WORD   XUP             ; POINTER TO USER AREA
                                ;  NOTE - THE USER AREA IS A HOOK IN THIS SYSTEM TO ALLOW MULTITASKING
                                ;  TO BE ADDED LATER.
000022  035126'                        .WORD   XS0             ; POINTER TO BEGINNING OF THE STACK
000024  043436'                        .WORD   XR0             ; POINTER TO BEGINNING OF RETURN STACK
000026  043146'                        .WORD   XTIB            ; POINTER TO TERMINAL INPUT BUFFER
000030  000037                        .WORD   37              ; MAXIMUM NAME-FIELD WIDTH, NORMALLY 31
000032  000000                        .WORD   0               ; WARNING MODE; 0=ERROR #, 1=DISK MESSAGE
                                ;  NOTE - WARNING MODE INITIALIZED TO ZERO, IN CASE DISK ISN'T UP.
000034  015426'                        .WORD   XDP             ; FENCE TO PROTECT AGAINST ACCIDENTAL
                                                                ; 'FORGET' OF THE SYSTEM.
000036  015426'                        .WORD   XDP             ; POINTER TO NEXT AVAILABLE DICTIONARY
                                                                ; LOCATION (RETURNED BY 'HERE').
000040  015410'                        .WORD   XXVOC           ; POINTER TO INITIAL VOCABULARY LINK
000042  035132'                        .WORD   DSKBUF          ; INITIALIZE 'FIRST'
000044  043146'                        .WORD   ENDBUF          ; INITIALIZE 'LIMIT'
000046  000000                        .WORD   0               ; AVAILABLE
000050  000000                        .WORD   0               ; AVAILABLE
                                ;
```

```
                                ;*****************************************************************
                                ;
                                ;                       NUCLEUS
                                ;
                                ;*****************************************************************
                                ;
                                ;
                                ;
                                ;  THE NUCLEUS CONTAINS THE PRIMITIVES FROM WHICH THE SYSTEM IS BUILT.
                                ;
                                ;
                                ;
                                ;
000052                                  HEAD    203,LIT,324,LIT                         ; ***** LIT
                                ;  USED ONLY BY COMPILER.  PUSH FOLLOWING LITERAL ONTO STACK.
000062  012445                          MOV     (IP)+,-(S)
000064                                  NEXT
                                ;
000070                                  HEAD    207,EXECUTE,305,EXEC                    ; ***** EXECUTE
                                ;  EXECUTE FORTH WORD WHOSE CODE ADDRESS IS ON STACK
000104  012502                          MOV     (S)+,W
000106  000132                          JMP     @(W)+
                                ;
                                ;
000110                                  HEAD    206,BRANCH,240,BRAN                     ; ***** BRANCH
                                ;  USED ONLY BY COMPILER.  FORTH BRANCH TO ADDRESS WHICH FOLLOWS.
000124  061404                          ADD     (IP),IP
000126                                  NEXT
                                ;
000132                                  HEAD    207,0BRANCH,310,ZBRAN                   ; ***** 0BRANCH
                                ;  USED ONLY BY COMPILER.  FORTH BRANCH IF TOP OF STACK
                                ;  IS ZERO (FALSE).
000146  005725                          TST     (S)+
000150  001003                          BNE     3$
000152  061404                          ADD     (IP),IP
000154                                  NEXT
000160  062704  000002          3$:     ADD     #2,IP
000164                                  NEXT
                                ;
000170                                  HEAD    206,(LOOP),240,XLOOP                    ; ***** (LOOP)
                                ;  USED ONLY BY COMPILER.  INCREMENT LOOP INDEX BY 1, BRANCH
                                ;  IF BELOW LIMIT.
000204  005216                          INC     (RP)
000206  021666  000002                  CMP     (RP),2(RP)
000212  002003                          BGE     1$
000214  061404                          ADD     (IP),IP
000216                                  NEXT
000222  062706  000004          1$:     ADD     #4,RP
000226  062704  000002                  ADD     #2,IP
000232                                  NEXT
                                ;
000236                                  HEAD    207,(+LOOP),251,XPLOO                   ; ***** (+LOOP)
                                ;  USED ONLY BY COMPILER.  INCREMENT LOOP INDEX BY TOP OF STACK,
                                ;  MAYBE BRANCH.
000252  061516                          ADD     (S),(RP)
000254  005725                          TST     (S)+
000256  002414                          BLT     2$
```

```
544 000260  026616  000002              CMP     2(RP),(RP)
545 000264  003403                      BLE     1$
546 000266  061404                      ADD     (IP),IP
547 000270                              NEXT
548 000274  062706  000004      1$:     ADD     #4,RP
549 000300  062704  000002              ADD     #2,IP
550 000304                              NEXT
551 000310  021666  000002      2$:     CMP     (RP),2(RP)      ; HANDLE NEGATIVE INCREMENT
552 000314  003767                      BLE     1$
553 000316  061404                      ADD     (IP),IP
554 000320                              NEXT
555                             ;
556 000324                              HEAD    204,(DO),240,XDO                        ; ***** (DO)
557                             ; USED ONLY BY COMPILER.  SET UP 'DO' LIMIT AND INDEX.
558 000336  016546  000002              MOV     2(S),-(RP)
559 000342  011546                      MOV     (S),-(RP)
560 000344  062705  000004              ADD     #4,S
561 000350                              NEXT
562                             ;
563 000354                              HEAD    201,I,311,I                             ; ***** I
564                             ;  RETURN CURRENT LOOP INDEX TO STACK.
565 000362  011645                      MOV     (RP),-(S)
566 000364                              NEXT
567                             ;
568 000370                              HEAD    205,DIGIT,324,DIGIT                     ; ***** DIGIT
569                             ;  USED BY COMPILER.
570                             ;  ( ASCII-DIGIT BASE ==> DIGIT-VALUE TRUE (OR FALSE))
571 000402  162765  000060  000002          SUB     #60,2(S)        ; VALID DIGIT IS ASCII 60 -
572 000410  026527  000002  000011          CMP     2(S),#11        ; IF GREATER THAN 9,
573 000416  003407                          BLE     1$
574 000420  162765  000007  000002          SUB     #7,2(S) ; SUBTRACT 7.
575 000426  026527  000002  000012          CMP     2(S),#12        ; AND THEN IF <10 (A)
576 000434  002412                          BLT     2$              ; ERROR
577 000436  005765  000002      1$:     TST     2(S)            ; IF LESS THAN ZERO, ERROR
578 000442  002407                          BLT     2$
579 000444  026515  000002                  CMP     2(S),(S)        ; OR IF NOT LESS THAN BASE, ERR
580 000450  002004                          BGE     2$
581 000452  012715  000001                  MOV     #1,(S)          ; VALID RETURN
582 000456                                  NEXT
583 000462  062705  000002      2$:     ADD     #2,S
584 000466  005015                          CLR     (S)             ; ERROR - RETURN '0' FLAG
585 000470                                  NEXT
586                             ;
587                             ;
588 000474                              HEAD    206,(FIND),240,PFIND                    ; ***** (FIND)
589                             ;  USED BY COMPILER.  FIND A WORD IN THE DICTIONARY.
590                             ;  ( STRING-ADDRESS NFA ==> PFA LENGTH TRUE (OR FALSE)).
591                             ;  STRING-ADDRESS IS ADDRESS OF THE LENGTH BYTE OF THE
592                             ;  STRING BEING SOUGHT.  NFA IS NAME-FIELD ADDRESS OF
593                             ;  WORD IN DICTIONARY WHERE SEARCH BEGINS.  PFA IS
594                             ;  PARAMETER-FIELD ADDRESS OF THE DICTIONARY ENTRY
595                             ;  WHICH IS FOUND.  IF WORD NOT FOUND, ONLY ONE RESULT
596                             ;  (0, FALSE) IS RETURNED.
597                             ; SETUP - GET ARGS, PRESERVE NEEDED REGISTERS
598 000510  012500                      MOV     (S)+,R0 ; DICTIONARY ADDRESS
599 000512  012501                      MOV     (S)+,R1 ; STRING ADDRESS
600 000514  010546                      MOV     R5,-(RP)        ; PRESERVE REGISTERS
```

```
                                        MOV     R4,-(RP)
                                        MOV     R3,-(RP)
                                        CLR     -(RP)           ; SPACE TO STORE LENGTH BYTE
                                ; PREPARE R2 FOR FAST COMPARE
                                        MOV     (R1),R2
                                        BIC     #100200,R2
                                ;
                                FCOMP:
                                ;  FAST TEST TO ELIMINATE MOST WORDS
                                ;  COMPARE FIRST WORD TO SPECIALLY PREPARED R2
                                ;  THEN INCREMENT TO FIND END OF NAME.
                                FAST:   MOV     (R0),R3
                                        BIC     #100300,R3
                                        CMP     R2,R3
                                        BEQ     NOFAST          ; NO FAST ELIMINATION POSSIBLE
                                XMATCH: TST     (R0)+           ; BRANCH HERE IF NO MATCH THIS TIME
                                        BPL     XMATCH
                                ;  R0 NOW POINTS TO LINK
                                        TST     (R0)
                                        BEQ     FAILED
                                        MOV     (R0),R0
                                        BR      FCOMP
                                ;  END OF FAST ELIMINATION TEST
                                ;
                                NOFAST: MOV     (R0),(RP)       ; SAVE LENGTH BYTE
                                        MOV     R1,R5           ; SET R5
                                        BR      NOFST1
                                ; NOW DO THE MAIN LOOP TO CHECK FOR MATCH
                                MLOOP:  TST     (R5)+
                                        MOV     (R5),R4
                                        MOV     (R0),R3
                                        BIC     #100000,R3
                                        CMP     R3,R4
                                        BNE     XMATCH
                                NOFST1: BIT     #100000,(R0)+
                                        BEQ     MLOOP
                                ;  IF GET HERE, FOUND IT.
                                        MOV     (RP)+,R2        ; POP AND SAVE LENGTH BYTE
                                        MOV     (RP)+,R3        ; RESTORE REGISTERS
                                        MOV     (RP)+,R4
                                        MOV     (RP)+,R5
                                        ADD     #4,R0           ; GET PARAMETER FIELD ADDRESS
                                        MOV     R0,-(S)
                                        BIC     #177400,R2      ; R2 CONTAINS LENGTH BYTE
                                        MOV     R2,-(S)
                                        MOV     #1,-(S)
                                        NEXT
                                FAILED: TST     (RP)+           ; POP LENGTH BYTE
                                        MOV     (RP)+,R3        ; RESTORE REGISTERS
                                        MOV     (RP)+,R4
                                        MOV     (RP)+,R5
                                        CLR     -(S)            ; REPLACE LENGTH BYTE WITH
                                                                ; FAILURE FLAG.
                                        NEXT                    ; WE ARE DONE - FAILURE TO FIND
                                ;
                                        HEAD    207,ENCLOSE,305,ENCL                    ; ***** ENCLOSE
                                ;  USED BY COMPILER.  BREAK NEXT WORD OUT OF INPUT BUFFER.
```

```
                                ;  ( START-ADDRESS DELIMITER ==> ADDRESS OFFSET END NEXT-CHARACTER)
000700  011500                        MOV     (S),R0          ; DELIMITER
000702  016501  000002                MOV     2(S),R1         ; STARTING ADDRESS
000706  162705  000004                SUB     #4,S            ; MAKE SPACE FOR RESULTS
000712  122100                ENC1:   CMPB    (R1)+,R0
000714  001776                        BEQ     ENC1            ; SKIP OVER LEADING DELIMITERS
000716  162701  000001                SUB     #1,R1
000722  010165  000004                MOV     R1,4(S)
000726  105711                ENC2:   TSTB    (R1)            ; TEST FOR NULL
000730  001420                        BEQ     ENC4
000732  122100                        CMPB    (R1)+,R0        ; NOT NULL, SO FIND END OF TOKEN
000734  001374                        BNE     ENC2
000736  010115                        MOV     R1,(S)
000740  162701  000001                SUB     #1,R1
000744  010165  000002        ENC3:   MOV     R1,2(S)         ; FINISH UP AND RETURN
000750  016501  000006                MOV     6(S),R1
000754  160115                        SUB     R1,(S)
000756  160165  000002                SUB     R1,2(S)
000762  160165  000004                SUB     R1,4(S)
000766                                NEXT
000772  010115                ENC4:   MOV     R1,(S)          ; HANDLE NULL CASE
000774  020165  000004                CMP     R1,4(S)
001000  001361                        BNE     ENC3
001002  062701  000001                ADD     #1,R1
001006  000756                        BR      ENC3
                                ;
                                ;
                                ;  THE NEXT 4 HEADERS POINT TO INSTALLATION-DEPENDENT TERMINAL I/O
                                ;  ROUTINES.
                                ;
                                ;
001010                                  HEAD    204,EMIT,240,EMIT,PEMIT                 ; ***** EMIT
                                ;
001022                                  HEAD    203,KEY,331,KEY,PKEY                    ; ***** KEY
                                ;
001032                                  HEAD    211,?TERMINAL,314,QTERM,PQTER           ; ***** ?TERMINAL
                                ;
001050                                  HEAD    202,CR,240,CR,PCR                       ; ***** CR
                                ;
                                ;
                                ;
                                ;
                                ;
001060                                  HEAD    205,CMOVE,305,CMOVE                     ; ***** CMOVE
                                ;  MOVE BYTES IN MEMORY.   ( FROM TO N ==>)
005715                                  TST     (S)
001407                                  BEQ     2$              ; NO MOVE
016500  000002                          MOV     2(S),R0
016501  000004                          MOV     4(S),R1
112120                          1$:     MOVB    (R1)+,(R0)+
005315                                  DEC     (S)
001375                                  BNE     1$
062705  000006                  2$:     ADD     #6,S
001120                                  NEXT
                                ;
                                ;
```

```
715 001124                              HEAD    202,U*,240,USTAR                        ; ***** U*
716                             ;  ( N1 N2 ==> PRODUCT).  PRODUCT IS 32-BIT DOUBLE INTEGER,
717                             ;  HIGH WORD TOP.
718                             ;  THIS MUST BE UNSIGNED MULTIPLICATION.
719 001134  004767  000004              JSR     PC,UMULT
720 001140                              NEXT
721 001144                      UMULT:
722                             ; THE VALUES TO MULTIPLY ARE ON THE STACK.
723 001144  012502                      MOV     (S)+,R2
724 001146  012746  000020              MOV     #20,-(RP)       ; SET LOOP COUNT
725 001152  005000                      CLR     R0
726 001154  005001                      CLR     R1
727 001156  006101              2$:     ROL     R1
728 001160  006100                      ROL     R0
729 001162  006102                      ROL     R2
730 001164  103002                      BCC     1$
731 001166  061501                      ADD     (S),R1
732 001170  005500                      ADC     R0
733 001172  005316              1$:     DEC     (RP)
734 001174  001370                      BNE     2$
735 001176  010115                      MOV     R1,(S)
736 001200  010045                      MOV     R0,-(S)
737 001202  005726                      TST     (RP)+           ; POP TEMPORARY
738 001204  000207                      RTS     PC
739                             ;
740 001206                              HEAD    202,U/,240,USLAS                        ; ***** U/
741                             ;  THIS DIVISION MUST BE UNSIGNED.
742 001216  004767  000004              JSR     PC,UDIV
743 001222                              NEXT
744 001226                      UDIV:
745                             ;  THE VALUES TO DIVIDE ARE ON THE STACK
746 001226  012502                      MOV     (S)+,R2 ; DIVISOR
747 001230  012500                      MOV     (S)+,R0
748 001232  012501                      MOV     (S)+,R1
749 001234  012745  000020              MOV     #20,-(S)        ; LOOP COUNT
750 001240  006301              1$:     ASL     R1
751 001242  006100                      ROL     R0
752 001244  001405                      BEQ     2$              ; NO NEED TO SUBTRACT
753 001246  160200                      SUB     R2,R0
754 001250  005201                      INC     R1
755 001252  103002                      BCC     2$
756 001254  060200                      ADD     R2,R0           ; MUST RESTORE
757 001256  005301                      DEC     R1
758 001260  005315              2$:     DEC     (S)             ; LOOP SIXTEEN TIMES
759 001262  001366                      BNE     1$
760 001264  005725                      TST     (S)+            ; POP TO DISCARD COUNT
761 001266  010045                      MOV     R0,-(S) ; REMAINDER
762 001270  010145                      MOV     R1,-(S) ; QUOTIENT
763 001272  000207                      RTS     PC
764                             ;
765 001274                              HEAD    203,AND,304,AND                         ; ***** AND
766                             ;  BITWISE AND.  ( N1 N2 ==> N3).
767 001304  005115                      COM     (S)
768 001306  042515                      BIC     (S)+,(S)
769 001310                              NEXT
770                             ;
771 001314                              HEAD    202,OR,240,OR                           ; ***** OR
```

```
 772 001324  052515                        BIS     (S)+,(S)
 773 001326                                 NEXT
 774                                ;
 775 001332                                 HEAD    203,XOR,322,XOR                 ; ***** XOR
 776                                        .IFDF   EIS
 777                                        MOV     (S)+,R0
 778                                        XOR     R0,(S)
 779                                        .IFF
 780 001342  011546                         MOV     (S),-(RP)
 781 001344  046516  000002                 BIC     2(S),(RP)
 782 001350  042515                         BIC     (S)+,(S)
 783 001352  052615                         BIS     (RP)+,(S)
 784                                        .ENDC
 785 001354                                 NEXT
 786                                ;
 787 001360                                 HEAD    203,SP@,300,SPAT                ; ***** SP@
 788 001370  010501                         MOV     S,R1
 789 001372  010145                         MOV     R1,-(S)
 790 001374                                 NEXT
 791                                ;
 792 001400                                 HEAD    203,SP!,241,SPSTO               ; ***** SP!
 793 001410  016305  000006                 MOV     6(U),S          ; OFFSET 6 IN USER AREA
 794
 795 001414                                 NEXT
 796                                ;
 797 001420                                 HEAD    203,RP!,241,RPSTO               ; ***** RP!
 798 001430  016706  176370                 MOV     ORIGIN+24,RP
 799 001434                                 NEXT
 800                                ;
 801 001440                                 HEAD    202,<;S>,240,SEMIS              ; ***** ;S
 802 001450  012604                         MOV     (RP)+,IP
 803 001452                                 NEXT
 804                                ;
 805 001456                                 HEAD    205,LEAVE,305,LEAVE             ; ***** LEAVE
 806 001470  011666  000002                 MOV     (RP),2(RP)
 807 001474                                 NEXT
 808                                ;
 809 001500                                 HEAD    202,^/>R/,240,TOR               ; ***** >R
 810 001510  012546                         MOV     (S)+,-(RP)
 811 001512                                 NEXT
 812                                ;
 813 001516                                 HEAD    202,R>,240,FROMR                ; ***** R>
 814 001526  012645                         MOV     (RP)+,-(S)
 815 001530                                 NEXT
 816                                ;
 817 001534                                 HEAD    201,R,322,R                     ; ***** R
 818 001542  011645                         MOV     (RP),-(S)
 819 001544                                 NEXT
 820                                ;
 821 001550                                 HEAD    202,0=,240,ZEQU                 ; ***** 0=
 822 001560  005715                         TST     (S)
 823 001562  001402                         BEQ     1$
 824 001564  005015                         CLR     (S)
 825 001566  000402                         BR      2$
 826 001570  012715  000001         1$:     MOV     #1,(S)
 827 001574                         2$:
 828 001574                                 NEXT
```

```
                                        ;
001600                                          HEAD    202,0<,240,ZLESS                        ; ***** 0<
001610  005715                                 TST     (S)
001612  100402                                 BMI     1$
001614  005015                                 CLR     (S)
001616  000402                                 BR      2$
001620  012715  000001                1$:     MOV     #1,(S)
001624                                  2$:
001624                                          NEXT
                                        ;
001630                                          HEAD    201,+,253,PLUS                          ; ***** +
001636  062515                                 ADD     (S)+,(S)
001640                                          NEXT
                                        ;
001644                                          HEAD    202,D+,240,DPLUS                        ; ***** D+
001654  066565  000002  000006                 ADD     2(S),6(S)       ; ADD LOW
001662  005565  000004                         ADC     4(S)
001666  061565  000004                         ADD     (S),4(S)        ; ADD HIGH
001672  062705  000004                         ADD     #4,S
001676                                          NEXT
                                        ;
001702                                          HEAD    205,MINUS,323,MINUS                     ; ***** MINUS
                                        ;  CHANGE SIGN.
001714  005415                                  NEG     (S)
001716                                          NEXT
                                        ;
001722                                          HEAD    206,DMINUS,240,DMINU                    ; ***** DMINUS
                                        ;  CHANGE SIGN OF DOUBLE INTEGER WORD ON STACK.
001736  005415                                  NEG     (S)
001740  005465  000002                         NEG     2(S)
001744  005615                                 SBC     (S)
001746                                          NEXT
                                        ;
001752                                          HEAD    204,OVER,240,OVER                       ; ***** OVER
                                        ;  ( N1 N2 ==> N1 N2 N1)
001764  016545  000002                         MOV     2(S),-(S)
001770                                          NEXT
                                        ;
001774                                          HEAD    204,DROP,240,DROP                       ; ***** DROP
002006  062705  000002                         ADD     #2,S
002012                                          NEXT
                                        ;
002016                                          HEAD    204,SWAP,240,SWAP                       ; ***** SWAP
002030  016501  000002                         MOV     2(S),R1
002034  011565  000002                         MOV     (S),2(S)
002040  010115                                 MOV     R1,(S)
002042                                          NEXT
                                        ;
002046                                          HEAD    203,DUP,320,DUP                         ; ***** DUP
002056  011545                                 MOV     (S),-(S)
002060                                          NEXT
                                        ;
002064                                          HEAD    202,+!,240,PSTOR                        ; ***** +!
                                        ;  ADD NUMBER SECOND ON STACK TO ADDRESS ON TOP.
002074  066575  000002  000000                 ADD     2(S),@(S)
002102  062705  000004                         ADD     #4,S
002106                                          NEXT
```

```
                                        ;
                                            HEAD    206,TOGGLE,240,TOGGL                    ; ***** TOGGLE
                                        ;   ( BYTE-ADDRESS BIT-PATTERN ==> )  EXCLUSIVE-OR INTO MEMORY BYTE.
                                            MOV     2(S),-(S)       ; PUSH THE BYTE
                                            MOVB    @(S),(S)        ;   TO BE TOGGLED
                                        ;   AVOID USING 'XOR' INSTRUCTION - NOT AVAILABLE ON ALL PDP-11
                                            MOV     (S),-(RP)
                                            BIC     2(S),(RP)
                                            BIC     (S)+,(S)
                                            BIS     (RP)+,(S)
                                            MOV     2(S),-(S)       ; SET UP RETURN ADDRESS
                                            MOVB    2(S),@(S)       ; PUT THE TOGGLED BYTE BACK TO MEM.
                                            ADD     #6,S            ; ADJUST STACK POINTER
                                            NEXT
                                        ;
                                            HEAD    201,@,300,AT                            ; ***** @
                                            MOV     @(S),(S)
                                            NEXT
                                        ;
                                            HEAD    202,C@,240,CAT                          ; ***** C@
                                            MOVB    @(S),R1
                                            BIC     #177400,R1
                                            MOV     R1,(S)
                                            NEXT
                                        ;
                                            HEAD    201,!,241,STORE                         ; ***** !
                                            MOV     2(S),@(S)
                                            ADD     #4,S
                                            NEXT
                                        ;
                                            HEAD    202,C!,240,CSTOR                        ; ***** C!
                                            MOVB    2(S),@(S)
                                            ADD     #4,S
                                            NEXT
                                        ;
                                        ;
```

```
                                        ; ************************************************************
                                        ;
                                        ;                PRE-COMPILED FORTH SECTION
                                        ;
                                        ; ************************************************************
                                        ;
                                        ;
                                        ;
                                        ;  NOTE - A FEW OF THE FOLLOWING OPERATIONS HAVE BEEN
                                        ;  CONVERTED TO CODE FOR SPEED.  HOWEVER, THE WORD ORDER
                                        ;  IN THE DICTIONARY HAS NOT BEEN CHANGED.
                                        ;
 002310                                         HEAD    301,:,272,COLON,DOCOL                   ; ***** :
 002316  004202' 004074' 003150'                .WORD   QEXEC,SCSP,CURR,AT,CONT,STORE,CREAT,RBRAC,PSCOD
 002324  002176' 003132' 002242'
 002332  006700' 004412' 004562'
 002340  010446                         DOCOL:  MOV     IP,-(RP)
 002342  010204                                 MOV     W,IP
 002344                                         NEXT
                                        ;
 002350                                         HEAD    301,<;>,273,SEMI,DOCOL                  ; ***** ;
 002356  004256' 004346' 001446'                .WORD   QCSP,COMP,SEMIS,SMUDG,LBRAC,SEMIS
 002364  004440' 004374' 001446'
                                        ;
 002372                                         HEAD    210,CONSTANT,240,CON,DOCOL              ; ***** CONSTANT
 002410  006700' 004440' 003416'                .WORD   CREAT,SMUDG,COMMA,PSCOD
 002416  004562'
 002420  011245                         DOCON:  MOV     (W),-(S)
 002422                                         NEXT
                                        ;
 002426                                         HEAD    210,VARIABLE,240,VAR,DOCOL              ; ***** VARIABLE
 002444  002406' 004562'                        .WORD   CON,PSCOD
 002450  010245                         DOVAR:  MOV     W,-(S)
 002452                                         NEXT
                                        ;
 002456                                         HEAD    204,USER,240,USER,DOCOL                 ; ***** USER
                                        ;  CREATE A NEW USER VARIABLE.  ( N ==> ).
 002470  002406' 004562'                        .WORD   CON,PSCOD
 002474  011245                         DOUSE:  MOV     (W),-(S)
 002476  060315                                 ADD     U,(S)
 002500                                         NEXT
                                        ;
                                        ;
                                        ;
                                        ;  CONSTANTS
                                        ;
 002504                                         HEAD    201,0,260,ZERO,DOCON                    ; ***** 0
 002512  000000                                 .WORD   0
                                        ;
 002514                                         HEAD    201,1,261,ONE,DOCON                     ; ***** 1
 002522  000001                                 .WORD   1
                                        ;
 002524                                         HEAD    201,2,262,TWO,DOCON                     ; ***** 2
 002532  000002                                 .WORD   2
                                        ;
 002534                                         HEAD    201,3,263,THREE,DOCON                   ; ***** 3
 002542  000003                                 .WORD   3
```

```
                                ;
002544                                  HEAD    202,BL,240,BL,DOCON             ; ***** BL
                                ;  BLANK.
002554  000040                          .WORD   40
                                ;
002556                                  HEAD    203,C/L,314,CL,DOCON            ; ***** C/L
                                ;  # OF CHARACTERS PER LINE
002566  000100                          .WORD   100
                                ;
                                ;  'FIRST' AND 'LIMIT' MOVED TO USER AREA
                                ;
002570                                  HEAD    205,B/BUF,306,BBUF,DOCON        ; ***** B/BUF
                                ;  BYTES PER DISK-BLOCK BUFFER.
002602  002000                          .WORD   1024.
                                ;
002604                                  HEAD    205,B/SCR,322,BSCR,DOCON        ; ***** B/SCR
                                ;  DISK BLOCKS PER FORTH SCREEN.
002616  000001                          .WORD   1
                                ;
002620                                  HEAD    207,+ORIGIN,316,PORIG,DOCOL     ; ***** +ORIGIN
                                ;  RETURNS ADDRESS, GIVEN OFFSET FROM ORIGIN.
002634  000060' 000000' 001634'         .WORD   LIT,ORIGIN,PLUS,SEMIS
002642  001446'
                                ;
                                ;  USER VARIABLES
                                ;
002644                                  HEAD    202,S0,240,SZERO,DOUSE          ; ***** S0
                                ;  STACK ORIGIN.
002654  000006                          .WORD   6
                                ;
002656                                  HEAD    202,R0,240,RZERO,DOUSE          ; ***** R0
                                ;  RETURN STACK ORIGIN.
002666  000010                          .WORD   10
                                ;
002670                                  HEAD    203,TIB,302,TIB,DOUSE           ; ***** TIB
                                ;  TERMINAL INPUT BUFFER.
002700  000012                          .WORD   12
                                ;
002702                                  HEAD    205,WIDTH,310,WIDTH,DOUSE       ; ***** WIDTH
                                ;  MAXIMUM NAME LENGTH (DEFAULT, 31 CHARACTERS).
002714  000014                          .WORD   14
                                ;
002716                                  HEAD    207,WARNING,307,WARN,DOUSE      ; ***** WARNING
                                ;  WARNING MODE (DEFAULT, GIVE MESSAGE NUMBER AT ERROR
                                ;  OR WARNING CONDITION, DON'T GO TO DISK FOR MESSAGE).
002732  000016                          .WORD   16
                                ;
002734                                  HEAD    205,FENCE,305,FENCE,DOUSE       ; ***** FENCE
                                ;  PREVENTS 'FORGET' BELOW THIS 'FENCE' SETTING.
002746  000020                          .WORD   20
                                ;
002750                                  HEAD    202,DP,240,DP,DOUSE             ; ***** DP
                                ;  DICTIONARY POINTER TO NEXT AVAILABLE SPACE.
002760  000022                          .WORD   22
                                ;
002762                                  HEAD    210,VOC-LINK,240,VOCL,DOUSE     ; ***** VOC-LINK
                                ;  VOCABULARY LINK (MAINLY FOR FUTURE USE).
```

```
1032 003000  000024                     .WORD   24
1033                                ;
1034 003002                         ;       HEAD    205,FIRST,324,FIRST,DOUSE               ; ***** FIRST
1035                                ;  ADDRESS OF BEGINNING OF DISK BUFFER.
1036 003014  000026                     .WORD   26
1037                                ;
1038 003016                         ;       HEAD    205,LIMIT,324,LIMIT,DOUSE               ; ***** LIMIT
1039                                ;  ADDRESS JUST BEYOND END OF DISK BUFFERS.
1040 003030  000030                     .WORD   30
1041                                ;
1042                                ; POSITIONS 32 AND 34 ARE AVAILABLE FOR EXPANSION.
1043                                ; THEY ARE INITIALIZED FROM BOOT-UP TABLE, AT COLD START.
1044                                ;
1045 003032                         ;       HEAD    203,BLK,313,BLK,DOUSE                   ; ***** BLK
1046                                ;  CURRENT DISK BLOCK BEING LOADED (0=TERMINAL)
1047 003042  000036                     .WORD   36
1048                                ;
1049 003044                         ;       HEAD    202,IN,240,IN,DOUSE                     ; ***** IN
1050                                ;  OFFSET IN TERMINAL INPUT BUFFER.
1051 003054  000040                     .WORD   40
1052                                ;
1053 003056                         ;       HEAD    203,OUT,324,OUT,DOUSE                   ; ***** OUT
1054                                ;  OFFSET IN OUTPUT LINE.
1055 003066  000042                     .WORD   42
1056                                ;
1057 003070                         ;       HEAD    203,SCR,322,SCR,DOUSE                   ; ***** SCR
1058                                ;  CURRENT FORTH DISK SCREEN.
1059 003100  000044                     .WORD   44
1060                                ;
1061 003102                         ;       HEAD    206,OFFSET,240,OFSET,DOUSE              ; ***** OFFSET
1062                                ;  OFFSET TO GET TO ANOTHER DISK DRIVE.
1063 003116  000046                     .WORD   46
1064                                ;
1065 003120                         ;       HEAD    207,CONTEXT,324,CONT,DOUSE              ; ***** CONTEXT
1066 003134  000050                     .WORD   50
1067                                ;
1068 003136                         ;       HEAD    207,CURRENT,324,CURR,DOUSE              ; ***** CURRENT
1069 003152  000052                     .WORD   52
1070                                ;
1071 003154                         ;       HEAD    205,STATE,305,STATE,DOUSE               ; ***** STATE
1072 003166  000054                     .WORD   54
1073                                ;
1074 003170                         ;       HEAD    204,BASE,240,BASE,DOUSE                 ; ***** BASE
1075 003202  000056                     .WORD   56
1076                                ;
1077 003204                         ;       HEAD    203,DPL,314,DPL,DOUSE                   ; ***** DPL
1078                                ;  OFFSET OF DECIMAL POINT AFTER DOUBLE-INTEGER INPUT.
1079 003214  000060                     .WORD   60
1080                                ;
1081 003216                         ;       HEAD    203,FLD,304,FLD,DOUSE                   ; ***** FLD
1082                                ;  OUTPUT FIELD WIDTH.
1083 003226  000062                     .WORD   62
1084                                ;
1085 003230                         ;       HEAD    203,CSP,320,CSP,DOUSE                   ; ***** CSP
1086                                ;  USED BY COMPILER TO HOLD CURRENT STACK POSITION,
1087                                ;  FOR ERROR CHECKING.
1088 003240  000064                     .WORD   64
```

```
1089                                    ;
1090 003242                                     HEAD    202,R#,240,RNUM,DOUSE                   ; ***** R#
1091                                    ;  CURSOR POSITION (FOR SOME EDITORS).
1092 003252  000066                             .WORD   66
1093                                    ;
1094 003254                                     HEAD    203,HLD,304,HLD,DOUSE                   ; ***** HLD
1095                                    ;  POINTS TO LAST CHARACTER HELD IN 'PAD'
1096 003264  000070                             .WORD   70
1097                                    ;
1098 003266                                     HEAD    203,USE,305,USE,DOUSE                   ; ***** USE
1099 003276  000072                             .WORD   72
1100                                    ;
1101 003300                                     HEAD    204,PREV,240,PREV,DOUSE                 ; ***** PREV
1102 003312  000074                             .WORD   74
1103                                    ;
1104                                    ;
1105                                    ; END OF USER AREA
1106                                    ;
1107                                    ;
1108 003314                                     HEAD    202,1+,240,ONEP                         ; ***** 1+
1109 003324  005215                             INC     (S)
1110 003326                                     NEXT
1111                                    ;
1112 003332                                     HEAD    202,2+,240,TWOP                         ; ***** 2+
1113 003342  062715  000002                     ADD     #2,(S)
1114 003346                                     NEXT
1115                                    ;
1116 003352                                     HEAD    204,HERE,240,HERE,DOCOL                 ; ***** HERE
1117 003364  002756' 002176' 001446'            .WORD   DP,AT,SEMIS
1118                                    ;
1119 003372                                     HEAD    205,ALLOT,324,ALLOT,DOCOL               ; ***** ALLOT
1120 003404  002756' 002072' 001446'            .WORD   DP,PSTOR,SEMIS
1121                                    ;
1122 003412                                     HEAD    201,<,>,254,COMMA,DOCOL                 ; ***** ,
1123 003420  003362' 002242' 002530'            .WORD   HERE,STORE,TWO,ALLOT,SEMIS
     003426  003402' 001446'
1124                                    ;
1125                                    ;  THIS SYSTEM DOES NOT USE 'C,'
1126                                    ;
1127 003432                                     HEAD    201,-,255,SUB                           ; ***** -
1128 003440  162515                             SUB     (S)+,(S)
1129 003442                                     NEXT
1130                                    ;
1131 003446                                     HEAD    201,=,275,EQUAL                         ; ***** =
1132 003454  026525  000002                     CMP     2(S),(S)+
1133 003460  001402                             BEQ     1$
1134 003462  005015                             CLR     (S)
1135 003464  000402                             BR      2$
1136 003466  012715  000001            1$:      MOV     #1,(S)
1137 003472                             2$:     NEXT
1138                                    ;
1139 003476                                     HEAD    201,^/</,274,LESS                       ; ***** <
1140 003504  026525  000002                     CMP     2(S),(S)+
1141 003510  002402                             BLT     1$
1142 003512  005015                             CLR     (S)
1143 003514  000402                             BR      2$
1144 003516  012715  000001            1$:      MOV     #1,(S)
```

```
1145 003522                                2$:     NEXT
1146                                       ;
1147 003526                                        HEAD    201,^/>/,276,GREAT                      ; ***** >
1148 003534  026525  000002                   CMP     2(S),(S)+
1149 003540  003002                           BGT     1$
1150 003542  005015                           CLR     (S)
1151 003544  000402                           BR      2$
1152 003546  012715  000001             1$:   MOV     #1,(S)
1153 003552                                2$:     NEXT
1154                                       ;
1155 003556                                        HEAD    203,ROT,324,ROT                         ; ***** ROT
1156 003566  011500                           MOV     (S),R0
1157 003570  016515  000004                   MOV     4(S),(S)
1158 003574  016565  000002  000004           MOV     2(S),4(S)
1159 003602  010065  000002                   MOV     R0,2(S)
1160 003606                                        NEXT
1161                                       ;
1162 003612                                        HEAD    205,SPACE,305,SPACE,DOCOL               ; ***** SPACE
1163 003624  000060' 000040  001020'              .WORD   LIT,40,EMIT,SEMIS
     003632  001446'
1164                                       ;
1165 003634                                        HEAD    204,-DUP,240,DDUP                       ; ***** -DUP
1166 003646  005715                           TST     (S)
1167 003650  001401                           BEQ     1$
1168 003652  011546                           MOV     (S),-(S)
1169 003654                                1$:     NEXT
1170                                       ;
1171 003660                                        HEAD    210,TRAVERSE,240,TRAV,DOCOL             ; ***** TRAVERSE
1172                                       ;  MOVE (FORWARDS OR BACKWARDS) ACROSS A (VARIABLE LENGTH)
1173                                       ;  DICTIONARY NAME FIELD.
1174 003676  002026'                            .WORD   SWAP
1175 003700  001762' 001634' 000060' XXN1:     .WORD   OVER,PLUS,LIT,177,OVER,CAT,LESS,ZBRAN,XXN1-.
     003706  000177  001762' 002216'
     003714  003502' 000144' 177760
1176 003722  002026' 002004' 001446'              .WORD   SWAP,DROP,SEMIS
1177                                       ;
1178 003730                                        HEAD    206,LATEST,240,LATES,DOCOL              ; ***** LATEST
1179 003744  003150' 002176' 002176'              .WORD   CURR,AT,AT,SEMIS
     003752  001446'
1180                                       ;
1181                                       ;  THE NEXT 4 OPERATORS CAN DEPEND ON COMPUTER WORD SIZE.
1182                                       ;  THEY CONVERT ADDRESSES WITHIN THE NAME FIELDS OF FORTH
1183                                       ;  DICTIONARY ENTRIES.
1184                                       ;
1185 003754                                        HEAD    203,LFA,301,LFA,DOCOL                   ; ***** LFA
1186 003764  000060' 000004  003436'              .WORD   LIT,4,SUB,SEMIS
     003772  001446'
1187                                       ;
1188 003774                                        HEAD    203,CFA,301,CFA,DOCOL                   ; ***** CFA
1189 004004  002530' 003436' 001446'              .WORD   TWO,SUB,SEMIS
1190                                       ;
1191 004012                                        HEAD    203,NFA,301,NFA,DOCOL                   ; ***** NFA
1192 004022  000060' 000005  003436'              .WORD   LIT,5,SUB,LIT,-1,TRAV,SEMIS
     004030  000060' 177777  003674'
     004036  001446'
1193                                       ;
1194 004040                                        HEAD    203,PFA,301,PFA,DOCOL                   ; ***** PFA
```

```
1195 004050  002520' 003674' 000060'          .WORD   ONE,TRAV,LIT,5,PLUS,SEMIS
     004056  000005  001634' 001446'
1196                                    ;
1197                                    ;  THE NEXT 7 OPERATIONS ARE USED BY THE COMPILER,  FOR
1198                                    ;  COMPILE-TIME SYNTAX-ERROR CHECKS.
1199                                    ;
1200 004064                                    HEAD    204,!CSP,240,SCSP,DOCOL                 ; ***** !CSP
1201 004076  001366' 003236' 002242'           .WORD   SPAT,CSP,STORE,SEMIS
     004104  001446'
1202                                    ;
1203 004106                                    HEAD    206,?ERROR,240,QERR,DOCOL               ; ***** ?ERROR
1204 004122  002026' 000144' 000010            .WORD   SWAP,ZBRAN,XXN2-.,ERROR,BRAN,XXN3-.
     004130  006530' 000122' 000004
1205 004136  002004'                    XXN2:   .WORD   DROP
1206 004140  001446'                    XXN3:   .WORD   SEMIS
1207                                    ;
1208 004142                                    HEAD    205,?COMP,320,QCOMP,DOCOL               ; ***** ?COMP
1209 004154  003164' 002176' 001556'           .WORD   STATE,AT,ZEQU,LIT,21,QERR,SEMIS
     004162  000060' 000021  004120'
     004170  001446'
1210                                    ;
1211 004172                                    HEAD    205,?EXEC,303,QEXEC,DOCOL               ; ***** ?EXEC
1212 004204  003164' 002176' 000060'           .WORD   STATE,AT,LIT,22,QERR,SEMIS
     004212  000022  004120' 001446'
1213                                    ;
1214 004220                                    HEAD    206,?PAIRS,240,QPAIR,DOCOL              ; ***** ?PAIRS
1215 004234  003436' 000060' 000023            .WORD   SUB,LIT,23,QERR,SEMIS
     004242  004120' 001446'
1216                                    ;
1217 004246                                    HEAD    204,?CSP,240,QCSP,DOCOL                 ; ***** ?CSP
1218 004260  001366' 003236' 002176'           .WORD   SPAT,CSP,AT,SUB,LIT,24,QERR,SEMIS
     004266  003436' 000060' 000024
     004274  004120' 001446'
1219                                    ;
1220 004300                                    HEAD    210,?LOADINQ,240,QLOAD,DOCOL            ; ***** ?LOADINQ
1221 004316  003040' 002176' 001556'           .WORD   BLK,AT,ZEQU,LIT,26,QERR,SEMIS
     004324  000060' 000026  004120'
     004332  001446'
1222                                    ;
1223 004334                                    HEAD    207,COMPILE,305,COMP,DOCOL              ; ***** COMPILE
1224                                    ;  COMPILE THE EXECUTION ADDRESS FOLLOWINQ.
1225 004350  004152' 001524' 002054'           .WORD   QCOMP,FROMR,DUP,TWOP,TOR,AT,COMMA,SEMIS
     004356  003340' 001506' 002176'
     004364  003416' 001446'
1226                                    ;
1227 004370                                    HEAD    301,[,333,LBRAC,DOCOL                   ; ***** [
1228                                    ;  STOP COMPILATION,  ENTER EXECUTION STATE.
1229 004376  002510' 003164' 002242'           .WORD   ZERO,STATE,STORE,SEMIS
     004404  001446'
1230                                    ;
1231 004406                                    HEAD    201,],335,RBRAC,DOCOL                   ; ***** ]
1232                                    ;  ENTER COMPILATION STATE.
1233 004414  000060' 000300  003164'           .WORD   LIT,300,STATE,STORE,SEMIS
     004422  002242' 001446'
1234                                    ;
1235 004426                                    HEAD    206,SMUDGE,240,SMUDQ,DOCOL              ; ***** SMUDGE
1236                                    ;  ALTER LATEST WORD NAME (SO THAT DICTIONARY SEARCH
```

```
1237                                    ;  WON'T FIND A PARTIALLY-COMPLETE ENTRY.
1238 004442  003742' 000060' 000040            .WORD   LATES,LIT,40,TOGGL,SEMIS
     004450  002124' 001446'
1239                                    ;
1240 004454                                     HEAD    203,HEX,330,HEX,DOCOL                   ;  ***** HEX
1241 004464  000060' 000020  003200'            .WORD   LIT,20,BASE,STORE,SEMIS
     004472  002242' 001446'
1242                                    ;
1243 004476                                     HEAD    207,DECIMAL,314,DEC,DOCOL               ;  ***** DECIMAL
1244 004512  000060' 000012  003200'            .WORD   LIT,12,BASE,STORE,SEMIS
     004520  002242' 001446'
1245                                    ;
1246 004524                                     HEAD    205,OCTAL,314,OCTAL,DOCOL               ;  ***** OCTAL
1247 004536  000060' 000010  003200'            .WORD   LIT,10,BASE,STORE,SEMIS
     004544  002242' 001446'
1248                                    ;
1249 004550                                     HEAD    207,<(;CODE)>,251,PSCOD,DOCOL           ;  ***** (;CODE)
1250                                    ;  USED ONLY BY COMPILER;  COMPILED BY ';CODE'.
1251 004564  001524' 003742' 004046'            .WORD   FROMR,LATES,PFA,CFA,STORE,SEMIS
     004572  004002' 002242' 001446'
1252                                    ;
1253                                    ;
1254                                    ;  ***** THE DEFINITION OF ';CODE' WAS MOVED TO THE END OF
1255                                    ;  THE DICTIONARY,  BECAUSE IT IS NOT PURE CODE (IT IS PATCHED
1256                                    ;  WHEN A FORTH ASSEMBLER IS LOADED).
1257
1258                                    ;
1259 004600                                     HEAD    207,^/<BUILDS/,323,BUILD,DOCOL          ;  ***** <BUILDS
1260                                    ;  CREATE NEW DATA TYPE WITH CODE ROUTINE IN HIGHER-LEVEL FORTH.
1261 004614  002510' 002406' 001446'            .WORD   ZERO,CON,SEMIS
1262                                    ;
1263 004622                                     HEAD    205,DOES>,276,DOES,DOCOL                ;  ***** DOES>
1264 004634  001524' 003742' 004046'            .WORD   FROMR,LATES,PFA,STORE,PSCOD
     004642  002242' 004562'
1265 004646  010446                    DODOE:  MOV     IP,-(RP)
1266 004650  012204                            MOV     (W)+,IP
1267 004652  010245                            MOV     W,-(S)
1268 004654                                     NEXT
1269                                    ;
1270 004660                                     HEAD    205,COUNT,324,COUNT,DOCOL               ;  ***** COUNT
1271                                    ;  CONVERT STRING TO THE FORMAT USED BY 'TYPE'.
1272 004672  002054' 003322' 002026'            .WORD   DUP,ONEP,SWAP,CAT,SEMIS
     004700  002216' 001446'
1273                                    ;
1274 004704                                     HEAD    204,TYPE,240,TYPE,DOCOL                 ;  ***** TYPE
1275 004716  003644' 000144' 000030             .WORD   DDUP,ZBRAN,XXL2-.,OVER,PLUS,SWAP,XDO
     004724  001762' 001634' 002026'
     004732  000334'
1276 004734  000360' 002216' 001020' XXL1:     .WORD   I,CAT,EMIT,XLOOP,XXL1-.,BRAN,XXL3-.
     004742  000202' 177770  000122'
     004750  000004
1277 004752  002004'                  XXL2:     .WORD   DROP
1278 004754  001446'                  XXL3:     .WORD   SEMIS
1279                                    ;
1280 004756                                     HEAD    206,=CELLS,240,ECELL,DOCOL              ;  ***** =CELLS
1281                                    ;  NOTE - I NEED THIS,  TO FORCE EVEN ADDRESS.
1282 004772  002054' 002520' 001302'            .WORD   DUP,ONE,AND,PLUS,SEMIS
```

```
     005000  001634' 001446'
1283                                 ;
1284 005004                                  HEAD    211,-TRAILING,307,DTRAI,DOCOL          ; ***** -TRAILING
1285 005022  002054' 002510' 000334'         .WORD   DUP,ZERO,XDO
1286 005030  001762' 001762' 001634' XXW6:   .WORD   OVER,OVER,PLUS,ONE,SUB,CAT
     005036  002520' 003436' 002216'
1287 005044  002552' 003436' 000144'         .WORD   BL,SUB,ZBRAN,XXW7-.,LEAVE,BRAN,XXWA-.
     005052  000010  001466' 000122'
     005060  000006
1288 005062  002520' 003436'        XXW7:   .WORD   ONE,SUB
1289 005066  000202' 177740  001446' XXWA:   .WORD   XLOOP,XXW6-.,SEMIS
1290                                 ;
1291 005074                                  HEAD    204,(."),240,PDOTQ,DOCOL               ; ***** (.")
1292                                 ;  USED ONLY BY COMPILER.  COMPILED BY '."'
1293 005106  001540' 004670' 002054'         .WORD   R,COUNT,DUP,ONEP,ECELL
     005114  003322' 004770'
1294 005120  001524' 001634' 001506'         .WORD   FROMR,PLUS,TOR,TYPE,SEMIS
     005126  004714' 001446'
1295                                 ;
1296 005132                                  HEAD    302,.",240,DOTQ,DOCOL                  ; ***** ."
1297                                 ;  TYPE ASCII MESSAGE.
1298 005142  000060' 000042  003164'         .WORD   LIT,34.,STATE,AT,ZBRAN,XXL6-.
     005150  002176' 000144' 000026
1299 005156  004346' 005104' 005750'         .WORD   COMP,PDOTQ,WORD,HERE,CAT,ONEP,ECELL
     005164  003362' 002216' 003322'
     005172  004770'
1300 005174  003402' 000122' 000012         .WORD   ALLOT,BRAN,XXL7-.
1301 005202  005750' 003362' 004670' XXL6:   .WORD   WORD,HERE,COUNT,TYPE
     005210  004714'
1302 005212  001446'               XXL7:   .WORD   SEMIS
1303                                 ;
1304 005214                                  HEAD    206,?ALIGN,240,QALIG,DOCOL             ; ***** ?ALIGN
1305 005230  003362' 002520' 001302'         .WORD   HERE,ONE,AND,ALLOT,SEMIS
     005236  003402' 001446'
1306                                 ;
1307 005242                                  HEAD    206,EXPECT,240,EXPEC,DOCOL             ; ***** EXPECT
1308                                 ;  READ N CHARACTERS TO MEMORY (AND TERMINATE WITH NULLS).
1309                                 ;  ( ADDRESS N ==>).
1310 005256  001762' 001634' 001762'         .WORD   OVER,PLUS,OVER,XDO
     005264  000334'
1311 005266  001030' 002054' 000060' XXK1:   .WORD   KEY,DUP,LIT,16,PORIG,AT,EQUAL,ZBRAN,XXK2-.
     005274  000016  002632' 002176'
     005302  003452' 000144' 000040
1312 005310  002004' 000060' 000010         .WORD   DROP,LIT,10,OVER,I,EQUAL,DUP,FROMR
     005316  001762' 000360' 003452'
     005324  002054' 001524'
1313 005330  002530' 003436' 001634'         .WORD   TWO,SUB,PLUS,TOR,SUB,BRAN,XXK3-.
     005336  001506' 003436' 000122'
     005344  000060
1314 005346  002054' 000060' 000015  XXK2:   .WORD   DUP,LIT,15,EQUAL,ZBRAN,XXK4-.
     005354  003452' 000144' 000016
1315 005362  001466' 002004' 002552'         .WORD   LEAVE,DROP,BL,ZERO,BRAN,XXK5-.
     005370  002510' 000122' 000004
1316 005376  002054'               XXK4:   .WORD   DUP
1317 005400  000360' 002270' 002510' XXK5:   .WORD   I,CSTOR,ZERO,I,ONEP,CSTOR,ZERO,I,TWOP,CSTOR
     005406  000360' 003322' 002270'
     005414  002510' 000360' 003340'
```

```
     005422  002270'
1318                                 ;  NOTE DIFFERENCE FOR STAND-ALONE,  BELOW
1319                                        .IFDF    ALONE
1320 005424  001020' 000202' 177636  XXK3:    .WORD    EMIT,XLOOP,XXK1-.,DROP,SEMIS
     005432  002004' 001446'
1321                                        .IFF
1322                                 XXK3:    .WORD    DROP,XLOOP,XXK1-.,DROP,SEMIS
1323                                        .ENDC
1324                                 ;
1325 005436                                 HEAD     205,QUERY,331,QUERY,DOCOL                ;  ***** QUERY
1326 005450  002676' 002176' 000060'        .WORD    TIB,AT,LIT,120,EXPEC,ZERO,IN,STORE,SEMIS
     005456  000120  005254' 002510'
     005464  003052' 002242' 001446'
1327                                 ;
1328 005472                                 HEAD     301,X,200,NULL,DOCOL                     ;  ***** THE NULL
1329                                 ;  THE NULL OPERATION (ASCII 0) STOPS INTERPRETATION/COMPILATION
1330                                 ;  AT END OF A TERMINAL INPUT LINE,  OR A DISK SCREEN.   ALL DISK
1331                                 ;  BUFFERS MUST TERMINATE WITH NULLS,  AND 'EXPECT' PLACES NULLS
1332                                 ;  AFTER EACH TERMINAL INPUT LINE.
1333                                 ;  NOTE THAT THE 'X' IN THE HEADER ABOVE WILL BE CHANGED TO A NULL.
1334 005500  003040' 002176'                .WORD    BLK,AT
1335 005504  000144' 000046  002520'        .WORD    ZBRAN,XXJ2-.,ONE,BLK,PSTOR,ZERO,IN,STORE
     005512  003040' 002072' 002510'
     005520  003052' 002242'
1336 005524  003040' 002176' 002614'        .WORD    BLK,AT,BSCR,MOD,ZEQU,ZBRAN,XXJ1-.,QEXEC,FROMR,DROP
     005532  010534' 001556' 000144'
     005540  000010  004202' 001524'
     005546  002004'
1337 005550  000122' 000006           XXJ1:    .WORD    BRAN,XXJ4-.
1338 005554  001524' 002004'          XXJ2:    .WORD    FROMR,DROP
1339 005560  001446'                  XXJ4:    .WORD    SEMIS
1340                                 ;
1341 005562                                 HEAD     204,FILL,240,FILL,DOCOL                  ;  ***** FILL
1342 005574  002026' 001506' 001762'        .WORD    SWAP,TOR,OVER,CSTOR,DUP,ONEP,FROMR
     005602  002270' 002054' 003322'
     005610  001524'
1343 005612  002520' 003436' 001070'        .WORD    ONE,SUB,CMOVE,SEMIS
     005620  001446'
1344                                 ;
1345 005622                                 HEAD     205,ERASE,305,ERASE,DOCOL                ;  ***** ERASE
1346 005634  002510' 005572' 001446'        .WORD    ZERO,FILL,SEMIS
1347                                 ;
1348 005642                                 HEAD     206,BLANKS,240,BLANK,DOCOL               ;  ***** BLANKS
1349 005656  002552' 005572' 001446'        .WORD    BL,FILL,SEMIS
1350                                 ;
1351 005664                                 HEAD     204,HOLD,240,HOLD,DOCOL                  ;  ***** HOLD
1352 005676  000060' 177777  003262'        .WORD    LIT,-1,HLD,PSTOR,HLD,AT,CSTOR,SEMIS
     005704  002072' 003262' 002176'
     005712  002270' 001446'
1353                                 ;
1354 005716                                 HEAD     203,PAD,304,PAD,DOCOL                    ;  ***** PAD
1355 005726  003362' 000060' 000104         .WORD    HERE,LIT,104,PLUS,SEMIS
     005734  001634' 001446'
1356                                 ;
1357 005740                                 HEAD     204,WORD,240,WORD,DOCOL                  ;  ***** WORD
1358 005752  003040' 002176' 000144'        .WORD    BLK,AT,ZBRAN,XXI1-.,BLK,AT,BLOCK,BRAN,XXI2-.
     005760  000014  003040' 002176'
```

```
      005766  011314' 000122' 000006
1359  005774  002676' 002176'          XXI1:   .WORD   TIB,AT
1360  006000  003052' 002176' 001634' XXI2:   .WORD   IN,AT,PLUS,SWAP,ENCL,HERE,LIT,42,BLANK,IN
      006006  002026' 000676' 003362'
      006014  000060' 000042  005654'
      006022  003052'
1361  006024  002072' 001762' 003436'         .WORD   PSTOR,OVER,SUB,TOR,R,HERE,CSTOR,PLUS
      006032  001506' 001540' 003362'
      006040  002270' 001634'
1362  006044  003362' 003322' 001524'         .WORD   HERE,ONEP,FROMR,CMOVE,SEMIS
      006052  001070' 001446'
1363                                    ;
1364                                    ;
1365  006056                            HEAD    210,(NUMBER),240,PNUMB,DOCOL            ; ***** (NUMBER)
1366  006074  003322' 002054' 001506' XXF3:   .WORD   ONEP,DUP,TOR,CAT,BASE,AT,DIGIT
      006102  002216' 003200' 002176'
      006110  000400'
1367  006112  000144' 000054  002026'         .WORD   ZBRAN,XXQ4-.,SWAP,BASE,AT,USTAR,DROP
      006120  003200' 002176' 001132'
      006126  002004'
1368  006130  003564' 003200' 002176'         .WORD   ROT,BASE,AT,USTAR,DPLUS
      006136  001132' 001652'
1369  006142  003212' 002176' 003322'         .WORD   DPL,AT,ONEP,ZBRAN,XXQ5-.,ONE,DPL,PSTOR
      006150  000144' 000010  002520'
      006156  003212' 002072'
1370  006162  001524' 000122' 177706  XXQ5:   .WORD   FROMR,BRAN,XXF3-.
1371  006170  001524' 001446'          XXQ4:   .WORD   FROMR,SEMIS
1372                                    ;
1373  006174                            HEAD    206,NUMBER,240,NUMB,DOCOL               ; ***** NUMBER
1374  006210  002510' 002510' 003564'         .WORD   ZERO,ZERO,ROT,DUP,ONEP,CAT,LIT,55,EQUAL
      006216  002054' 003322' 002216'
      006224  000060' 000055  003452'
1375  006232  002054' 001506' 001634'         .WORD   DUP,TOR,PLUS,LIT,-1
      006240  000060' 177777
1376  006244  003212' 002242' 006072' XXF6:   .WORD   DPL,STORE,PNUMB,DUP,CAT,BL,SUB
      006252  002054' 002216' 002552'
      006260  003436'
1377  006262  000144' 000026  002054'         .WORD   ZBRAN,XXF7-.,DUP,CAT,LIT,56,SUB
      006270  002216' 000060' 000056
      006276  003436'
1378  006300  002510' 004120' 002510'         .WORD   ZERO,QERR,ZERO,BRAN,XXF6-.
      006306  000122' 177734
1379  006312  002004' 001524' 000144' XXF7:   .WORD   DROP,FROMR,ZBRAN,XXFA-.,DMINU
      006320  000004  001734'
1380  006324  001446'                  XXFA:   .WORD   SEMIS
1381                                    ;
1382  006326                            HEAD    205,-FIND,304,DFIND,DOCOL               ; ***** -FIND
1383  006340  002552' 005750' 003362'         .WORD   BL,WORD,HERE,COUNT,UPPER,HERE,CONT,AT,AT,PFIND
      006346  004670' 006416' 003362'
      006354  003132' 002176' 002176'
      006362  000506'
1384  006364  002054' 001556' 000144'         .WORD   DUP,ZEQU,ZBRAN,XXE3-.,DROP,HERE,LATES,PFIND
      006372  000012  002004' 003362'
      006400  003742' 000506'
1385  006404  001446'                  XXE3:   .WORD   SEMIS
1386                                    ;
1387  006406                            HEAD    205,UPPER,322,UPPER,DOCOL               ; ***** UPPER
```

```
1388                                    ;  SETS STRINGS TO UPPER CASE - TO ALLOW
1389                                    ;  LOWER AS WELL AS UPPER CASE FROM TERMINAL.
1390 006420  001762' 001634' 002026'          .WORD   OVER,PLUS,SWAP,XDO
     006426  000334'
1391 006430  000360' 002216' 000060' XXE2:    .WORD   I,CAT,LIT,137,GREAT,I,CAT,LIT,173,LESS
     006436  000137  003532' 000360'
     006444  002216' 000060' 000173
     006452  003502'
1392 006454  001302' 000144' 000012          .WORD   AND,ZBRAN,XXE1-.,I,LIT,40,TOGGL
     006462  000360' 000060' 000040
     006470  002124'
1393 006472  000202' 177734  001446' XXE1:    .WORD   XLOOP,XXE2-.,SEMIS
1394                                    ;
1395 006500                                    HEAD    207,(ABORT),251,PABOR,DOCOL             ; ***** (ABORT)
1396 006514  007652' 001446'                   .WORD   ABORT,SEMIS
1397                                    ;
1398 006520                                    HEAD    205,ERROR,322,ERROR,DOCOL               ; ***** ERROR
1399 006532  002730' 002176' 001606'          .WORD   WARN,AT,ZLESS,ZBRAN,XXN4-.,PABOR
     006540  000144' 000004  006512'
1400 006546  003362' 004670' 004714' XXN4:    .WORD   HERE,COUNT,TYPE,PDOTQ
     006554  005104'
1401 006556     003                           .BYTE   3
1402 006557     040     077     040           .ASCII  / ? /
1403                                           .EVEN
1404 006562  011550' 001406' 003052'          .WORD   MESS,SPSTO,IN,AT,BLK,AT,QUIT,SEMIS
     006570  002176' 003040' 002176'
     006576  007574' 001446'
1405                                    ;
1406 006602                                    HEAD    203,ID.,256,IDDOT,DOCOL                 ; ***** ID.
1407 006612  005724' 000060' 000040          .WORD   PAD,LIT,40,LIT,137,FILL,DUP
     006620  000060' 000137  005572'
     006626  002054'
1408 006630  004046' 003762' 001762'          .WORD   PFA,LFA,OVER,SUB,PAD,SWAP,CMOVE
     006636  003436' 005724' 002026'
     006644  001070'
1409 006646  005724' 004670' 000060'          .WORD   PAD,COUNT,LIT,37,AND,TYPE,SPACE,SEMIS
     006654  000037  001302' 004714'
     006662  003622' 001446'
1410                                    ;
1411 006666                                    HEAD    206,CREATE,240,CREAT,DOCOL              ; ***** CREATE
1412 006702  006336' 000144' 000020          .WORD   DFIND,ZBRAN,XXD2-.,DROP,NFA,IDDOT
     006710  002004' 004020' 006610'
1413 006716  000060' 000004  011550'          .WORD   LIT,4,MESS,SPACE
     006724  003622'
1414 006726  003362' 002054' 002216' XXD2:    .WORD   HERE,DUP,CAT,WIDTH,AT,MIN,ONEP,ALLOT
     006734  002712' 002176' 010206'
     006742  003322' 003402'
1415 006746  005226' 002054' 000060'          .WORD   GALIQ,DUP,LIT,240,TOGGL,HERE,ONE,SUB
     006754  000240  002124' 003362'
     006762  002520' 003436'
1416 006766  000060' 000200  002124'          .WORD   LIT,200,TOGGL,LATES,COMMA,CURR,AT,STORE
     006774  003742' 003416' 003150'
     007002  002176' 002242'
1417 007006  003362' 003340' 003416'          .WORD   HERE,TWOP,COMMA,SEMIS
     007014  001446'
1418                                    ;
1419 007016                                    HEAD    311,[COMPILE],335,BCOMP,DOCOL           ; ***** [COMPILE]
```

```
1420 007034  006336' 001556' 002510'          .WORD   DFIND,ZEQU,ZERO,QERR,DROP,CFA,COMMA,SEMIS
     007042  004120' 002004' 004002'
     007050  003416' 001446'
1421                                          ;
1422 007054                                   HEAD    307,LITERAL,314,LITER,DOCOL             ; ***** LITERAL
1423 007070  003164' 002176' 000144'          .WORD   STATE,AT,ZBRAN,XXD6-.,COMP,LIT,COMMA
     007076  000010  004346' 000060'
     007104  003416'
1424 007106  001446'                    XXD6: .WORD   SEMIS
1425                                          ;
1426 007110                                   HEAD    310,DLITERAL,240,DLITE,DOCOL            ; ***** DLITERAL
1427 007126  003164' 002176' 000144'          .WORD   STATE,AT,ZBRAN,XXN5-.,SWAP,LITER,LITER
     007134  000010  002026' 007066'
     007142  007066'
1428 007144  001446'                    XXN5: .WORD   SEMIS
1429                                          ;
1430 007146                                   HEAD    202,U<,240,ULESS,DOCOL                  ; ***** U<
1431                                          ;  UNSIGNED LESS-THAN,  NEEDED FOR '?STACK'
1432                                          ;  : U< >R 0 R> 0 DMINUS D+ SWAP DROP 0< ;
1433 007156  001506' 002510' 001524'          .WORD   TOR,ZERO,FROMR,ZERO,DMINU,DPLUS
     007164  002510' 001734' 001652'
1434 007172  002026' 002004' 001606'          .WORD   SWAP,DROP,ZLESS,SEMIS
     007200  001446'
1435                                          ;
1436 007202                                   HEAD    206,?STACK,240,QSTAC,DOCOL              ; ***** ?STACK
1437                                          ;  ERROR CHECK.
1438 007216  002652' 002176' 002530'          .WORD   SZERO,AT,TWO,SUB,SPAT,ULESS,ONE,QERR
     007224  003436' 001366' 007154'
     007232  002520' 004120'
1439 007236  001366' 003362' 000060'          .WORD   SPAT,HERE,LIT,200,PLUS,ULESS,TWO,QERR
     007244  000200  001634' 007154'
     007252  002530' 004120'
1440 007256  001446'                          .WORD   SEMIS
1441                                          ;
1442 007260                                   HEAD    211,INTERPRET,324,INTER,DOCOL           ; ***** INTERPRET
1443 007276  006336'                    XXE4: .WORD   DFIND
1444 007300  000144' 000036  003164'          .WORD   ZBRAN,XXEA-.,STATE,AT,LESS
     007306  002176' 003502'
1445 007312  000144' 000012  004002'          .WORD   ZBRAN,XXE5-.,CFA,COMMA,BRAN,XXE6-.
     007320  003416' 000122' 000006
1446 007326  004002' 000102'            XXE5: .WORD   CFA,EXEC
1447 007332  007214' 000122' 000034     XXE6: .WORD   QSTAC,BRAN,XXE7-.
1448 007340  003362' 006206' 003212'    XXEA: .WORD   HERE,NUMB,DPL,AT,ONEP,ZBRAN,XXF4-.,DLITE,BRAN,XXF5-.
     007346  002176' 003322' 000144'
     007354  000010  007124' 000122'
     007362  000006
1449 007364  002004' 007066'           XXF4: .WORD   DROP,LITER
1450 007370  007214'                    XXF5: .WORD   QSTAC
1451 007372  000122' 177702             XXE7: .WORD   BRAN,XXE4-.
1452                                          ;
1453 007376                                   HEAD    211,IMMEDIATE,305,IMMED,DOCOL           ; ***** IMMEDIATE
1454 007414  003742' 000060' 000100           .WORD   LATES,LIT,100,TOGGL,SEMIS
     007422  002124' 001446'
1455                                          ;
1456 007426                                   HEAD    212,VOCABULARY,240,VOCAB,DOCOL          ; ***** VOCABULARY
1457 007446  004612' 000060' 120201           .WORD   BUILD,LIT,120201,COMMA,CURR,AT,CFA,COMMA
     007454  003416' 003150' 002176'
```

```
     007462  004002' 003416'
1458 007466  003362' 002776' 002176'          .WORD   HERE,VOCL,AT,COMMA,VOCL,STORE,DOES
     007474  003416' 002776' 002242'
     007502  004632'
1459 007504  003340' 003132' 002242' DOVOC:   .WORD   TWOP,CONT,STORE,SEMIS
     007512  001446'
1460                                 ;
1461                                 ;
1462                                 ;  ***** THE DEFINITION OF 'FORTH' WAS MOVED TO NEAR THE END OF THE
1463                                 ;  DICTIONARY, BECAUSE IT IS NOT PURE CODE.
1464                                 ;
1465                                 ;
1466 007514                                 HEAD    213,DEFINITIONS,323,DEFIN,DOCOL          ; ***** DEFINITIONS
1467 007534  003132' 002176' 003150'        .WORD   CONT,AT,CURR,STORE,SEMIS
     007542  002242' 001446'
1468                                 ;
1469 007546                                 HEAD    301,(,250,PAREN,DOCOL                    ; ***** (
1470 007554  000060' 000051  005750'        .WORD   LIT,51,WORD,SEMIS
     007562  001446'
1471                                 ;
1472 007564                                 HEAD    204,QUIT,240,QUIT,DOCOL                  ; ***** QUIT
1473 007576  002510' 003040' 002242'        .WORD   ZERO,BLK,STORE,LBRAC
     007604  004374'
1474 007606  001426' 001056' 005446' XXB1:   .WORD   RPSTO,CR,QUERY,INTER,STATE,AT
     007614  007274' 003164' 002176'
1475 007622  001556' 000144' 000010         .WORD   ZEQU,ZBRAN,XXB2-.,PDOTQ
     007630  005104'
1476 007632     003                         .BYTE   3
1477 007633     040     117     113         .ASCII  / OK/
1478                                        .EVEN
1479 007636  000122' 177746         XXB2:   .WORD   BRAN,XXB1-.
1480                                 ;
1481 007642                                 HEAD    205,ABORT,324,ABORT,DOCOL                ; ***** ABORT
1482 007654  001406' 004510' 003622'        .WORD   SPSTO,DEC,SPACE
1483 007662  001056' 005104'                .WORD   CR,PDOTQ
1484 007666     021                         .BYTE   21
1485 007667     106     111     107         .ASCII  /FIG-FORTH  V 1.3 /
     007672     055     106     117
     007675     122     124     110
     007700     040     040     126
     007703     040     061     056
     007706     063     040
1486                                        .EVEN
1487 007710  015400' 007532' 007574'        .WORD   FORTH,DEFIN,QUIT
1488                                 ;
1489                                 ;  COLD AND WARM STARTS
1490                                 ;
1491 007716                                 HEAD    204,COLD,240,COLD                        ; ***** COLD
1492 007730                         CENT:                               ; COLD START ENTRY POINT
1493 007730  016767  170060  005450         MOV     ORIGIN+14,FORTH+6   ; SET 'FORTH' VOCABULARY FROM STARTUP TABLE
1494 007736  016703  170056                 MOV     ORIGIN+20,U         ; INITIALIZE USER POINTER
1495                                 ;  NOTE - FOR SMALLER STAND-ALONE BOOT, INITIALIZE AREAS IN
1496                                 ;  HIGH MEMORY WHICH MUST BE INITIALIZED.
1497                                 ;  CLEAR DISK BUFFERS ON FIRST TIME THROUGH
1498 007742  016700  170074                 MOV     ORIGIN+42,R0        ; 'FIRST' - BEGINNING OF DISK BUFFERS
1499 007746  016701  170072                 MOV     ORIGIN+44,R1        ; 'LIMIT' - JUST BEYOND DISK BUFFERS
1500 007752  005020                 1$:     CLR     (R0)+
```

```
1501 007754  020001                          CMP     R0,R1
1502 007756  002775                          BLT     1$
1503                                 ;  NOW INITIALIZE 'OUT', 'OFFSET', 'USE' AND 'PREV'
1504 007760  005063  000042                  CLR     42(U)           ; CLEAR 'OUT'
1505 007764  005063  000046                  CLR     46(U)           ; CLEAR 'OFFSET'
1506 007770  016763  170046  000072          MOV     ORIGIN+42,72(U) ; TO 'USE'
1507 007776  016763  170040  000074          MOV     ORIGIN+42,74(U) ; TO 'PREV'
1508                                 ;  END OF SPECIAL HIGH-MEMORY INITIALIZE
1509 010004  012701  000030                  MOV     #30,R1          ; ON COLD START, MOVE 24. BYTES
1510 010010  000402                          BR      W2
1511 010012                          WENT:                           ; WARM START ENTRY POINT
1512 010012  012701  000012                  MOV     #12,R1          ; ON WARM START, MOVE TEN BYTES
1513 010016  012705  000022'        W2:      MOV     #ORIGIN+22,R5   ; START MOVING FROM HERE
1514 010022  016700  167772                  MOV     ORIGIN+20,R0    ; MOVE TO THE USER AREA
1515 010026  062700  000006                  ADD     #6,R0           ;   PLUS 6
1516 010032  060501                          ADD     R5,R1           ; COMPUTE LOOP STOP ADDRESS
1517 010034  012520                  1$:     MOV     (R5)+,(R0)+
1518 010036  020501                          CMP     R5,R1
1519 010040  002775                          BLT     1$
1520 010042  016706  167756                  MOV     ORIGIN+24,RP    ; INITIALIZE THE RETURN-STACK POINTER
1521                                 ;  NOW SET FORTH'S INSTRUCTION COUNTER, AND GO
1522 010046  012704  010056'                 MOV     #GO,IP  ; START EXECUTION WITH 'ABORT'
1523 010052                                  NEXT
1524                                 ;  NOTE - NORMALLY THE ABOVE INSTRUCTION WOULD BE 'MOV #ABORT+2,IP'.
1525                                 ;  IT HAS BEEN CHANGED HERE TO ALLOW USER TO PATCH A DIFFERENT
1526                                 ;  START-UP.  BUT THE SYSTEM WON'T WORK UNTIL SOME OF THE WORK
1527                                 ;  OF 'ABORT' HAS BEEN DONE, SO THAT WORK IS REPEATED.  THE USER
1528                                 ;  CAN PATCH OVER THE 'ABORT' AND THE ZEROS.
1529                                 ;
1530 010056  001406' 004510' 015400' GO:      .WORD   SPSTO,DEC,FORTH,DEFIN,ABORT,0,0,0
     010064  007532' 007652' 000000
     010072  000000  000000
1531                                 ;
1532                                 ;
1533                                 ;
1534                                 ;
1535                                 ;
1536 010076                                  HEAD    204,S->D,240,STOD                       ; ***** S->D
1537 010110  005045                          CLR     -(S)            ; SIGN EXTEND WITH ZEROS
1538 010112  005765  000002                  TST     2(S)            ; BUT IF NEGATIVE,
1539 010116  100001                          BPL     1$
1540 010120  005315                          DEC     (S)             ;  CHANGE THE ZEROS TO ONES
1541 010122                          1$:     NEXT
1542                                 ;
1543                                 ;  NOTE - THIS SYSTEM DOESN'T NEED THE OPERATIONS '+-' AND 'D+-',
1544                                 ;    BECAUSE 'M*' AND 'M/' ARE DEFINED IN CODE.
1545                                 ;
1546 010126                                  HEAD    203,ABS,323,ABS,DOCOL                   ; ***** ABS
1547 010136  002054' 001606' 000144'         .WORD   DUP,ZLESS,ZBRAN,XXR5-.,MINUS
     010144  000004  001712'
1548 010150  001446'                 XXR5:   .WORD   SEMIS
1549                                 ;
1550 010152                                  HEAD    204,DABS,240,DABS,DOCOL                 ; ***** DABS
1551 010164  002054' 001606' 000144'         .WORD   DUP,ZLESS,ZBRAN,XXRB-.,DMINU
     010172  000004  001734'
1552 010176  001446'                 XXRB:   .WORD   SEMIS
1553                                 ;
```

```
1554 010200                                        HEAD    203,MIN,316,MIN,DOCOL                   ; ***** MIN
1555 010210  001762' 001762' 003532'               .WORD   OVER,OVER,GREAT,ZBRAN,XXR7-.,SWAP
     010216  000144' 000004  002026'
1556 010224  002004' 001446'          XXR7:        .WORD   DROP,SEMIS
1557                                               ;
1558 010230                                        HEAD    203,MAX,330,MAX,DOCOL                   ; ***** MAX
1559 010240  001762' 001762' 003502'               .WORD   OVER,OVER,LESS,ZBRAN,XXR6-.,SWAP
     010246  000144' 000004  002026'
1560 010254  002004' 001446'          XXR6:        .WORD   DROP,SEMIS
1561                                               ;
1562 010260                                        HEAD    202,M*,240,MSTAR                        ; ***** M*
1563                                               .IFDF   EIS             ; HARDWARE MULTIPLY/DIVIDE?
1564                                               MOV     (S)+,R0
1565                                               MUL     (S),R0
1566                                               MOV     R1,(S)
1567                                               MOV     R0,-(S)
1568                                               NEXT
1569                                               .IFF
1570 010270  016546  000002                       MOV     2(S),-(RP)      ; USE RETURN STACK FOR SAVING SIGN
1571 010274  100002                               BPL     1$
1572 010276  005465  000002                       NEG     2(S)            ; GET ABSOLUTE VALUE
1573 010302  005715                    1$:        TST     (S)
1574 010304  100002                               BPL     2$
1575 010306  005416                               NEG     (RP)            ; ADJUST SIGN WHICH WAS SAVED
1576 010310  005415                               NEG     (S)             ; GET ABSOLUTE VALUE
1577 010312  004767  170626              2$:      JSR     PC,UMULT
1578 010316  005726                               TST     (RP)+           ; NEGATIVE RESULT?
1579 010320  100007                               BPL     3$              ; NO
1580                                    ; IF GET HERE, NEGATE THE DOUBLE-INTEGER NUMBER ON THE STACK
1581 010322  005115                               COM     (S)
1582 010324  005165  000002                       COM     2(S)
1583 010330  062765  000001  000002               ADD     #1,2(S)
1584 010336  005515                               ADC     (S)
1585 010340                            3$:        NEXT
1586                                               .ENDC
1587                                               ;
1588 010344                                        HEAD    202,M/,240,MSLAS                        ; ***** M/
1589                                               .IFDF   EIS             ; HARDWARD MULTIPLY/DIVIDE?
1590                                               MOV     2(S),R0
1591                                               MOV     4(S),R1
1592                                               DIV     (S)+,R0
1593                                               MOV     R1,2(S)
1594                                               MOV     R0,(S)
1595                                               NEXT
1596                                               .IFF
1597 010354  016546  000002                       MOV     2(S),-(RP)      ; SAVE DIVIDEND SIGN
1598 010360  001001                               BNE     5$              ; ZERO WOULDN'T INDICATE
1599 010362  005216                               INC     (RP)            ;   A  SIGN CHANGE.
1600 010364  011646                    5$:        MOV     (RP),-(RP)      ; DUPLICATE IT
1601 010366  100011                               BPL     1$
1602                                    ;  IF GET HERE, TAKE ABSOLUTE VALUE OF DOUBLE-INTEGER DIVIDEND.
1603 010370  005165  000002                       COM     2(S)
1604 010374  005165  000004                       COM     4(S)
1605 010400  062765  000001  000004               ADD     #1,4(S)
1606 010406  005565  000002                       ADC     2(S)
1607 010412  005715                    1$:        TST     (S)             ; IS DIVISOR NEGATIVE?
1608 010414  100002                               BPL     2$
```

```
1609 010416  005416                          NEG     (RP)            ; IF YES, NEGATE QUOTIENT SIGN
1610 010420  005415                          NEG     (S)             ; AND TAKE ABS. VALUE OF DIVISOR
1611 010422  004767  170600          2$:     JSR     PC,UDIV
1612 010426  005726                          TST     (RP)+           ; NEGATIVE QUOTIENT?
1613 010430  100001                          BPL     3$              ; NO
1614 010432  005415                          NEG     (S)             ; NEGATE THE QUOTIENT
1615 010434  005726                  3$:     TST     (RP)+           ; NEGATIVE DIVIDEND?
1616 010436  100002                          BPL     4$              ; NEGATE THE REMAINDER
1617 010440  005465  000002                  NEG     2(S)
1618 010444                          4$:     NEXT
1619                                         .ENDC
1620                                 ;
1621 010450                                  HEAD    201,*,252,STAR,DOCOL                    ; ***** *
1622 010456  010266' 002004' 001446'         .WORD   MSTAR,DROP,SEMIS
1623                                 ;
1624 010464                                  HEAD    204,/MOD,240,SLMOD,DOCOL                ; ***** /MOD
1625 010476  001506' 010106' 001524'         .WORD   TOR,STOD,FROMR,MSLAS,SEMIS
     010504  010352' 001446'
1626                                 ;
1627 010510                                  HEAD    201,/,257,SLASH,DOCOL                   ; ***** /
1628 010516  010474' 002026' 002004'         .WORD   SLMOD,SWAP,DROP,SEMIS
     010524  001446'
1629                                 ;
1630 010526                                  HEAD    203,MOD,304,MOD,DOCOL                   ; ***** MOD
1631 010536  010474' 002004' 001446'         .WORD   SLMOD,DROP,SEMIS
1632                                 ;
1633 010544                                  HEAD    205,*/MOD,304,SSMOD,DOCOL               ; ***** */MOD
1634 010556  001506' 010266' 001524'         .WORD   TOR,MSTAR,FROMR,MSLAS,SEMIS
     010564  010352' 001446'
1635                                 ;
1636 010570                                  HEAD    202,*/,240,SSLA,DOCOL                   ; ***** */
1637 010600  010554' 002026' 002004'         .WORD   SSMOD,SWAP,DROP,SEMIS
     010606  001446'
1638                                 ;
1639 010610                                  HEAD    205,M/MOD,304,MSMOD,DOCOL               ; ***** M/MOD
1640 010622  001506' 002510' 001540'         .WORD   TOR,ZERO,R,USLAS,FROMR
     010630  001214' 001524'
1641 010634  002026' 001506' 001214'         .WORD   SWAP,TOR,USLAS,FROMR,SEMIS
     010642  001524' 001446'
1642                                 ;
1643                                 ;
1644                                 ;
1645                                 ;
1646                                 ;
```

```
1648                                    ; ***************************************************************
1649                                    ;
1650                                    ;       DISK I/O  (SECTION COMMON TO ALL OPERATING SYSTEMS)
1651                                    ;       NOTE THAT EACH OPERATING SYSTEM DEFINED 'R/W' - READ
1652                                    ;       OR WRITE A 1024-BYTE RANDOM-ACCESS BLOCK.
1653                                    ;
1654                                    ; ***************************************************************
1655                                    ;
1656                                    ;
1657                                    ; 'USE' AND 'PREV' MOVED TO USER AREA
1658                                    ;
1659 010646                                    HEAD    204,+BUF,240,PBUF,DOCOL                 ; ***** +BUF
1660 010660  002600' 000060' 000004            .WORD   BBUF,LIT,4,PLUS,PLUS,DUP,LIMIT,AT,EQUAL
     010666  001634' 001634' 002054'
     010674  003026' 002176' 003452'
1661 010702  000144' 000010  002004'          .WORD   ZBRAN,XXT1-.,DROP,FIRST,AT
     010710  003012' 002176'
1662 010714  002054' 003310' 002176' XXT1:    .WORD   DUP,PREV,AT,SUB,SEMIS
     010722  003436' 001446'
1663                                    ;
1664 010726                                    HEAD    206,UPDATE,240,UPDAT,DOCOL              ; ***** UPDATE
1665 010742  003310' 002176' 002176'          .WORD   PREV,AT,AT,LIT,100000,OR,PREV
     010750  000060' 100000  001322'
     010756  003310'
1666 010760  002176' 002242' 001446'          .WORD   AT,STORE,SEMIS
1667                                    ;
1668 010766                                    HEAD    215,EMPTY-BUFFERS,323,MTBUF,DOCOL       ; ***** EMPTY-BUFFERS
1669 011010  003012' 002176' 003026'          .WORD   FIRST,AT,LIMIT,AT,OVER,SUB,ERASE,SEMIS
     011016  002176' 001762' 003436'
     011024  005632' 001446'
1670                                    ;
1671 011030                                    HEAD    205,FLUSH,310,FLUSH,DOCOL               ; ***** FLUSH
1672                                    ;  SOME SYSTEMS DEFINE THIS IN THE EDITOR,  NOT HERE.
1673 011042  003026' 002176' 003012'          .WORD   LIMIT,AT,FIRST,AT,XDO
     011050  002176' 000334'
1674 011054  000360' 002176' 001606' XXTA:    .WORD   I,AT,ZLESS,ZBRAN,XXT7-.,I,TWOP,I,AT
     011062  000144' 000024  000360'
     011070  003340' 000360' 002176'
1675 011076  000060' 077777  001302'          .WORD   LIT,77777,AND,ZERO,RW
     011104  002510' 015174'
1676 011110  002600' 000060' 000004  XXT7:    .WORD   BBUF,LIT,4,PLUS,XPLOO,XXTA-.,MTBUF,SEMIS
     011116  001634' 000250' 177732
     011124  011006' 001446'
1677                                    ;
1678 011130                                    HEAD    203,DR0,260,DRZER,DOCOL                 ; ***** DR0
1679                                    ;  SELECT DRIVE #0 - NOT USED WITH RT11 OR RSX11
1680 011140  002510' 003114' 002242'          .WORD   ZERO,OFSET,STORE,SEMIS
     011146  001446'
1681                                    ;
1682 011150                                    HEAD    203,DR1,261,DRONE,DOCOL                 ; ***** DR1
1683                                    ; SELECT DRIVE #1 - NOT USED IN RSX11 OR RT11
1684 011160  000060' 000360  003114'          .WORD   LIT,240.,OFSET,STORE,SEMIS
     011166  002242' 001446'
1685                                    ;
1686 011172                                    HEAD    206,BUFFER,240,BUFFE,DOCOL              ; ***** BUFFER
1687 011206  003274' 002176' 002054'          .WORD   USE,AT,DUP,TOR
     011214  001506'
```

```
1688 011216  010656' 000144' 177774  XXT2:    .WORD   PBUF, ZBRAN, XXT2-., USE, STORE
     011224  003274' 002242'
1689 011230  001540' 002176' 001606'          .WORD   R, AT, ZLESS, ZBRAN, XXT3-.
     011236  000144' 000024
1690 011242  001540' 003340' 001540'          .WORD   R, TWOP, R, AT, LIT, 77777, AND
     011250  002176' 000060' 077777
     011256  001302'
1691 011260  002510' 015174'                  .WORD   ZERO, RW
1692 011264  001540' 002242' 001540' XXT3:    .WORD   R, STORE, R, PREV, STORE, FROMR, TWOP, SEMIS
     011272  003310' 002242' 001524'
     011300  003340' 001446'
1693                                  ;
1694 011304                                   HEAD    205, BLOCK, 313, BLOCK, DOCOL                   ; ***** BLOCK
1695                                  ; CHANGED TO MASK OFF THE UPDATE BIT WHEN COMPARING
1696 011316  003114' 002176' 001634'          .WORD   OFSET, AT, PLUS, TOR
     011324  001506'
1697 011326  003310' 002176' 002054'          .WORD   PREV, AT, DUP, AT, LIT, 077777, AND, R, SUB, ZBRAN, XXT4-.
     011334  002176' 000060' 077777
     011342  001302' 001540' 003436'
     011350  000144' 000066
1698 011354  010656' 001556' 000144' XXT5:    .WORD   PBUF, ZEQU, ZBRAN, XXT6-.
     011362  000024
1699 011364  002004' 001540' 011204'          .WORD   DROP, R, BUFFE
1700 011372  002054' 001540' 002520'          .WORD   DUP, R, ONE, RW, TWO, SUB
     011400  015174' 002530' 003436'
1701 011406  002054' 002176' 000060' XXT6:    .WORD   DUP, AT, LIT, 077777, AND, R, SUB, ZEQU
     011414  077777  001302' 001540'
     011422  003436' 001556'
1702 011426  000144' 177724                  .WORD   ZBRAN, XXT5-.
1703 011432  002054' 003310' 002242'          .WORD   DUP, PREV, STORE
1704 011440  001524' 002004' 003340' XXT4:    .WORD   FROMR, DROP, TWOP, SEMIS
     011446  001446'
1705                                  ;
1706                                  ;
1707                                  ;
1708                                  ;
1709                                  ;
1710                                  ;
1711 011450                                   HEAD    206, (LINE), 240, PLINE, DOCOL                  ; ***** (LINE)
1712 011464  001506' 002564' 002600'          .WORD   TOR, CL, BEUF, SSMOD, FROMR, BSCR
     011472  010554' 001524' 002614'
1713 011500  010454' 001634' 011314'          .WORD   STAR, PLUS, BLOCK, PLUS, CL, SEMIS
     011506  001634' 002564' 001446'
1714                                  ;
1715 011514                                   HEAD    205, .LINE, 305, DLINE, DOCOL                   ; ***** .LINE
1716 011526  011462' 005020' 004714'          .WORD   PLINE, DTRAI, TYPE, SEMIS
     011534  001446'
1717                                  ;
1718 011536                                   HEAD    207, MESSAGE, 305, MESS, DOCOL                  ; ***** MESSAGE
1719 011552  002730' 002176' 000144'          .WORD   WARN, AT, ZBRAN, XXW5-., DDUP, ZBRAN, XXW3-., LIT, 4
     011560  000034  003644' 000144'
     011566  000022  000060' 000004
1720 011574  003114' 002176' 002614'          .WORD   OFSET, AT, BSCR, SLASH, SUB, DLINE
     011602  010514' 003436' 011524'
1721 011610  000122' 000016          XXW3:    .WORD   BRAN, XXW4-.
1722 011614  005104'                 XXW5:    .WORD   PDOTQ
1723 011616     006                           .BYTE   6
```

```
1724 011617     115     123     107          .ASCII  /MSG # /
     011622     040     043     040
1725                                          .EVEN
1726 011626  013144'                          .WORD   DOT
1727 011630  001446'                  XXW4:   .WORD   SEMIS
1728                                          ;
1729 011632                                   HEAD    204,LOAD,240,LOAD,DOCOL                 ; ***** LOAD
1730 011644  003040' 002176' 001506'          .WORD   BLK,AT,TOR,IN,AT,TOR,ZERO,IN,STORE
     011652  003052' 002176' 001506'
     011660  002510' 003052' 002242'
1731 011666  002614' 010454' 003040'          .WORD   BSCR,STAR,BLK,STORE,INTER,FROMR,IN,STORE
     011674  002242' 007274' 001524'
     011702  003052' 002242'
1732 011706  001524' 003040' 002242'          .WORD   FROMR,BLK,STORE,SEMIS
     011714  001446'
1733                                          ;
1734 011716                                   HEAD    303,-->,276,ARROW,DOCOL                 ; ***** -->
1735 011726  004314' 002510' 003052'          .WORD   GLOAD,ZERO,IN,STORE,BSCR,BLK,AT,OVER
     011734  002242' 002614' 003040'
     011742  002176' 001762'
1736 011746  010534' 003436' 003040'          .WORD   MOD,SUB,BLK,PSTOR,SEMIS
     011754  002072' 001446'
1737                                          ;
1738                                          ;
1739                                          ;
1740                                          ;
1741                                          ;  NOTE - THE INSTALLATION-DEPENDENT I/O IS AT THE END
1742                                          ;  OF THE DICTIONARY - JUST BELOW 'TASK'.   'XI/O' IS THE
1743                                          ;  PRIMITIVE READ OR WRITE OF A 512-BYTE BLOCK.
1744                                          ;
1745                                          ;
1746                                          ;
1747                                          ;
1748                                          ;
1749                                          ;
```

```
1751                                    ; ***************************************************************
1752                                    ;
1753                                    ;       MISCELLANEOUS HIGHER LEVEL
1754                                    ;
1755                                    ; ***************************************************************
1756                                    ;
1757                                    ;
1758 011760                                    HEAD    301,',247,TICK,DOCOL                    ; ***** '
1759 011766  006336' 001556' 002510'           .WORD   DFIND,ZEQU,ZERO,QERR,DROP,LITER,SEMIS
     011774  004120' 002004' 007066'
     012002  001446'
1760                                    ;
1761 012004                                    HEAD    206,FORGET,240,FORGE,DOCOL              ; ***** FORGET
1762 012020  003150' 002176' 003132'           .WORD   CURR,AT,CONT,AT,SUB,LIT,30.,QERR,TICK,DUP
     012026  002176' 003436' 000060'
     012034  000030  004120' 011764'
     012042  002054'
1763 012044  002744' 002176' 003502'           .WORD   FENCE,AT,LESS,LIT,25.,QERR
     012052  000060' 000025  004120'
1764 012060  002054' 004020' 002756'           .WORD   DUP,NFA,DP,STORE,LFA,AT,CONT,AT
     012066  002242' 003762' 002176'
     012074  003132' 002176'
1765 012100  002242' 001446'                   .WORD   STORE,SEMIS
1766                                    ;
1767                                    ;
1768                                    ;
1769                                    ;
1770 012104                                    HEAD    204,BACK,240,BACK,DOCOL                 ; ***** BACK
1771 012116  003362' 003436' 003416'           .WORD   HERE,SUB,COMMA,SEMIS
     012124  001446'
1772                                    ;
1773 012126                                    HEAD    305,BEGIN,316,BEGIN,DOCOL               ; ***** BEGIN
1774 012140  004152' 003362' 002520'           .WORD   QCOMP,HERE,ONE,SEMIS
     012146  001446'
1775                                    ;
1776 012150                                    HEAD    305,ENDIF,306,ENDIF,DOCOL               ; ***** ENDIF
1777 012162  004152' 002530' 004232'           .WORD   QCOMP,TWO,QPAIR,HERE,OVER,SUB,SWAP,STORE,SEMIS
     012170  003362' 001762' 003436'
     012176  002026' 002242' 001446'
1778                                    ;
1779 012204                                    HEAD    304,THEN,240,THEN,DOCOL                 ; ***** THEN
1780 012216  012160' 001446'                   .WORD   ENDIF,SEMIS
1781                                    ;
1782 012222                                    HEAD    302,DO,240,DO,DOCOL                     ; ***** DO
1783 012232  004346' 000334' 003362'           .WORD   COMP,XDO,HERE,LIT,3,SEMIS
     012240  000060' 000003  001446'
1784                                    ;
1785 012246                                    HEAD    304,LOOP,240,LOOP,DOCOL                 ; ***** LOOP
1786 012260  000060' 000003  004232'           .WORD   LIT,3,QPAIR,COMP,XLOOP,BACK,SEMIS
     012266  004346' 000202' 012114'
     012274  001446'
1787                                    ;
1788 012276                                    HEAD    305,+LOOP,320,PLOOP,DOCOL               ; ***** +LOOP
1789 012310  000060' 000003  004232'           .WORD   LIT,3,QPAIR,COMP,XPLOO,BACK,SEMIS
     012316  004346' 000250' 012114'
     012324  001446'
1790                                    ;
```

```
1791 012326                                        HEAD    305,UNTIL,314,UNTIL,DOCOL                    ; ***** UNTIL
1792 012340  002520' 004232' 004346'               .WORD   ONE,QPAIR,COMP,ZBRAN,BACK,SEMIS
     012346  000144' 012114' 001446'
1793                                       ;
1794 012354                                        HEAD    303,END,304,END,DOCOL                        ; ***** END
1795 012364  012336' 001446'                       .WORD   UNTIL,SEMIS
1796                                       ;
1797 012370                                        HEAD    305,AGAIN,316,AGAIN,DOCOL                    ; ***** AGAIN
1798 012402  002520' 004232' 004346'               .WORD   ONE,QPAIR,COMP,BRAN,BACK,SEMIS
     012410  000122' 012114' 001446'
1799                                       ;
1800 012416                                        HEAD    306,REPEAT,240,REPEAT,DOCOL                  ; ***** REPEAT
1801 012432  001506' 001506' 012400'               .WORD   TOR,TOR,AGAIN,FROMR,FROMR,TWO,SUB,ENDIF,SEMIS
     012440  001524' 001524' 002530'
     012446  003436' 012160' 001446'
1802                                       ;
1803 012454                                        HEAD    302,IF,240,IF,DOCOL                          ; ***** IF
1804 012464  004346' 000144' 003362'               .WORD   COMP,ZBRAN,HERE,ZERO,COMMA,TWO,SEMIS
     012472  002510' 003416' 002530'
     012500  001446'
1805                                       ;
1806 012502                                        HEAD    304,ELSE,240,ELSE,DOCOL                      ; ***** ELSE
1807 012514  002530' 004232' 004346'               .WORD   TWO,QPAIR,COMP,BRAN,HERE,ZERO,COMMA
     012522  000122' 003362' 002510'
     012530  003416'
1808 012532  002026' 002530' 012160'               .WORD   SWAP,TWO,ENDIF,TWO,SEMIS
     012540  002530' 001446'
1809                                       ;
1810 012544                                        HEAD    305,WHILE,305,WHILE,DOCOL                    ; ***** WHILE
1811 012556  012462' 003340' 001446'               .WORD   IF,TWOP,SEMIS
1812                                       ;
1813                                       ;
1814                                       ;
1815                                       ;
1816 012564                                        HEAD    206,SPACES,240,SPACS,DOCOL                   ; ***** SPACES
1817 012600  002510' 010236' 003644'               .WORD   ZERO,MAX,DDUP,ZBRAN,XXR4-.,ZERO,XDO
     012606  000144' 000014  002510'
     012614  000334'
1818 012616  003622' 000202' 177774  XXRA:         .WORD   SPACE,XLOOP,XXRA-.
1819 012624  001446'                 XXR4:         .WORD   SEMIS
1820                                       ;
1821 012626                                        HEAD    202,^/<#/,240,BDIGS,DOCOL                    ; ***** <#
1822 012636  005724' 003262' 002242'               .WORD   PAD,HLD,STORE,SEMIS
     012644  001446'
1823                                       ;
1824 012646                                        HEAD    202,#>,240,EDIGS,DOCOL                       ; ***** #>
1825 012656  002004' 002004' 003262'               .WORD   DROP,DROP,HLD,AT,PAD,OVER,SUB,SEMIS
     012664  002176' 005724' 001762'
     012672  003436' 001446'
1826                                       ;
1827 012676                                        HEAD    204,SIGN,240,SIGN,DOCOL                      ; ***** SIGN
1828 012710  003564' 001606' 000144'               .WORD   ROT,ZLESS,ZBRAN,XXR1-.,LIT,55,HOLD
     012716  000010  000060' 000055
     012724  005674'
1829 012726  001446'                 XXR1:         .WORD   SEMIS
1830                                       ;
1831 012730                                        HEAD    201,#,243,DIG,DOCOL                          ; ***** #
```

```
1832 012736  003200' 002176' 010620'          .WORD   BASE,AT,MSMOD,ROT,LIT,11,OVER,LESS
     012744  003564' 000060' 000011
     012752  001762' 003502'
1833 012756  000144' 000010  000060'          .WORD   ZBRAN,XXR2-.,LIT,7,PLUS
     012764  000007  001634'
1834 012770  000060' 000060  001634' XXR2:    .WORD   LIT,60,PLUS,HOLD,SEMIS
     012776  005674' 001446'
1835                                  ;
1836 013002                                   HEAD    202,#S,240,DIGS,DOCOL                   ; ***** #S
1837 013012  012734' 001762' 001762' XXR3:    .WORD   DIG,OVER,OVER,OR,ZEQU,ZBRAN,XXR3-.,SEMIS
     013020  001322' 001556' 000144'
     013026  177764  001446'
1838                                  ;
1839 013032                                   HEAD    203,D.R,322,DDOTR,DOCOL                 ; ***** D.R
1840 013042  001506' 002026' 001762'          .WORD   TOR,SWAP,OVER,DABS,BDIGS,DIGS,SIGN,EDIGS
     013050  010162' 012634' 013010'
     013056  012706' 012654'
1841 013062  001524' 001762' 003436'          .WORD   FROMR,OVER,SUB,SPACS,TYPE,SEMIS
     013070  012576' 004714' 001446'
1842                                  ;
1843 013076                                   HEAD    202,.R,240,DOTR,DOCOL                   ; ***** .R
1844 013106  001506' 010106' 001524'          .WORD   TOR,STOD,FROMR,DDOTR,SEMIS
     013114  013040' 001446'
1845                                  ;
1846 013120                                   HEAD    202,D.,240,DDOT,DOCOL                   ; ***** D.
1847 013130  002510' 013040' 003622'          .WORD   ZERO,DDOTR,SPACE,SEMIS
     013136  001446'
1848                                  ;
1849 013140                                   HEAD    201,.,256,DOT,DOCOL                     ; ***** .
1850 013146  010106' 013126' 001446'          .WORD   STOD,DDOT,SEMIS
1851                                  ;
1852 013154                                   HEAD    201,?,277,QUEST,DOCOL                   ; ***** ?
1853 013162  002176' 013144' 001446'          .WORD   AT,DOT,SEMIS
1854                                  ;
1855 013170                                   HEAD    202,U.,240,UDOT,DOCOL                   ; ***** U.
1856 013200  002510' 013126' 001446'          .WORD   ZERO,DDOT,SEMIS
1857                                  ;
1858                                  ;  UTILITY SECTION.
1859                                  ;
1860                                  ;
1861 013206                                   HEAD    204,LIST,240,LIST,DOCOL                 ; ***** LIST
1862                                  ; ( N---.  LIST GIVEN SCREEN. )
1863 013220  004510' 001056' 002054'          .WORD   DEC,CR,DUP,SCR,STORE,PDOTQ
     013226  003076' 002242' 005104'
1864 013234     006                          .BYTE   6
1865 013235     123      103      122         .ASCII  /SCR # /
     013240     040      043      040
1866                                          .EVEN
1867 013244  013144' 000060' 000020          .WORD   DOT,LIT,20,ZERO,XDO
     013252  002510' 000334'
1868 013256  001056' 000360' 002540' XXZ1:    .WORD   CR,I,THREE,DOTR,SPACE
     013264  013104' 003622'
1869 013270  000360' 003076' 002176'          .WORD   I,SCR,AT,DLINE,XLOOP,XXZ1-.,CR,SEMIS
     013276  011524' 000202' 177754
     013304  001056' 001446'
1870                                  ;
1871 013310                                   HEAD    205,INDEX,330,INDEX,DOCOL               ; ***** INDEX
```

```
1872                                    ;  LIST FIRST LINE OF A RANGE OF DISK SCREENS.
1873 013322  001056' 003322' 002026'            .WORD   CR,ONEP,SWAP,XDO
     013330  000334'
1874 013332  001056' 000360' 002540' XXZ2:       .WORD   CR,I,THREE,DOTR,SPACE,ZERO,I,DLINE
     013340  013104' 003622' 002510'
     013346  000360' 011524'
1875 013352  001046' 000144' 000004             .WORD   QTERM,ZBRAN,XXZ3-.,LEAVE
     013360  001466'
1876 013362  000202' 177746  001446' XXZ3:      .WORD   XLOOP,XXZ2-.,SEMIS
1877                                    ;
1878 013370                                     HEAD    205,TRIAD,304,TRIAD,DOCOL               ; ***** TRIAD
1879                                    ;  LIST DISK SCREENS THREE PER PAGE.
1880 013402  000060' 000014  001020'            .WORD   LIT,14,EMIT             ; FORM FEED
1881 013410  002540' 010514' 002540'            .WORD   THREE,SLASH,THREE,STAR,THREE
     013416  010454' 002540'
1882 013422  001762' 001634' 002026'            .WORD   OVER,PLUS,SWAP,XDO
     013430  000334'
1883 013432  001056' 000360' 013216' XXZ4:      .WORD   CR,I,LIST,XLOOP,XXZ4-.,CR,LIT,17,MESS,CR,SEMIS
     013440  000202' 177770  001056'
     013446  000060' 000017  011550'
     013454  001056' 001446'
1884                                    ;
1885 013460                                     HEAD    205,VLIST,324,VLIST,DOCOL               ; ***** VLIST
1886 013472  000060' 000200  003064'            .WORD   LIT,200,OUT,STORE,CONT,AT,AT
     013500  002242' 003132' 002176'
     013506  002176'
1887 013510  003064' 002176' 000060' XXZ5:      .WORD   OUT,AT,LIT,100,GREAT,ZBRAN,XXZ6-.
     013516  000100  003532' 000144'
     013524  000012
1888 013526  001056' 002510' 003064'            .WORD   CR,ZERO,OUT,STORE
     013534  002242'
1889 013536  002054' 006610' 003622' XXZ6:      .WORD   DUP,IDDOT,SPACE,SPACE,PFA,LFA,AT
     013544  003622' 004046' 003762'
     013552  002176'
1890 013554  002054' 001556' 001046'            .WORD   DUP,ZEQU,QTERM,OR,ZBRAN,XXZ5-.,DROP,SEMIS
     013562  001322' 000144' 177722
     013570  002004' 001446'
1891                                    ;
1892                                            .IFDF   LINKS
1893                                            HEAD    205,VLINK,313,XVLINK,DOCON              ; ***** VLINK
1894                                    ;  THIS IS ONLY USED FOR LINKAGE FROM FORTH TO SUBROUTINES
1895                                    ;  IN OTHER LANGUAGES.  SEE USER'S GUIDE FOR DOCUMENTATION.
1896                                            .WORD   VLINK
1897                                            .ENDC
1898                                    ;
1899                                    ;
1900                                    ;
1901                                    ;
1902                                    ;
1903                                    ;
```

```
1905                    ; ****************************************************************
1906                    ;
1907                    ; INSTALLATION-DEPENDENT SECTION (TERMINAL AND DISK I/O, AND TRAPS)
1908                    ;
1909                    ; ****************************************************************
1910                    ;
1911                    ;
1912                    ; ***************
1913                    ;
1914                    ; RSX-11M TERMINAL I/O
1915                    ;
1916                    ; ***************
1917                    ;
1918                            .IFDF   RSX11
1919                    ;
1920                            .EVEN
1921                            ; NOTE - FOR RSX-11 ON HEAVILY LOADED MACHINES, IT IS BETTER
1922                            ; FOR 'KEY' TO READ A WHOLE LINE AT A TIME, AND UNPACK IT.
1923                            ; ALSO, 'KEY' SHOULD EMIT A LINE FEED WHEN A CARRIAGE RETURN
1924                            ; HAS BEEN READ.
1925
1926                    PEMIT:  JSR     R1,ITERM        ; INITIALIZE RSX?
1927                    ;  INCREMENT 'OUT', UNLESS A CONTROL CHARACTER BEING OUTPUT.
1928                            CMP     (S),#40         ; TEST FOR CONTROL CHARACTER
1929                            BLT     1$
1930                            INC     42(U)           ; INCREMENT 'OUT'
1931                    1$:
1932                            JSR     R1,XCOUT        ; OUTPUT A CHARACTER
1933                            NEXT
1934                    ;
1935                    PKEY:   JSR     R1,ITERM        ; INITIALIZE RSX?
1936                            TST     INTERM          ; ZERO MEANS READ NEW LINE
1937                            BNE     XCHAR
1938                    XLINE:  MOV     #XBUFF,INTERM   ; READ NEW LINE
1939                            QIOW$C  IO.RVB,4,4,,IOSTAT,,<XBUFF,80.>
1940                            ADD     #XBUFF,IOSTAT+2 ; TERMINATE LINE WITH CR
1941                            MOVB    #15,@IOSTAT+2
1942                    XCHAR:  TST     -(S)            ; DECREMENT STACK POINTER
1943                            MOVB    @INTERM,(S)     ;   FOR THIS BYTE INSTRUCTION
1944                            INC     INTERM
1945                            BIC     #177600,(S)
1946                            CMP     (S),#15 ; IF CR IS BEING SENT,
1947                            BNE     XRETRN
1948                            CLR     INTERM          ; THEN READ NEW LINE NEXT TIME.
1949                            MOV     #12,-(S)        ; AND ALSO EMIT A LINE FEED
1950                            JSR     R1,XCOUT
1951                    XRETRN: NEXT
1952                    ;
1953                    PGTER:  JSR     R1,ITERM        ; INITIALIZE RSX?
1954                            MOV     @FLAG,-(S)
1955                            CLR     @FLAG
1956                            NEXT
1957                    ;  PUT THE AST ROUTINE HERE
1958                    AST1:   MOV     (RP)+,@FLAG     ; SET UP FOR NEXT '?TERMINAL';
1959                                                    ;   NOTE THAT 'RP' IS SYSTEM STACK.
1960                            CMP     @FLAG,#3        ; TEST FOR ^C
1961                            BNE     1$
```

```
1962                                    EXIT$S
1963                            1$:     ASTX$S
1964                            ;
1965                            ;
1966                            PCR:    JSR     R1,ITERM        ; INITIALIZE RSX?
1967                                    MOV     #15,-(S)
1968                                    JSR     R1,XCOUT
1969                                    MOV     #12,-(S)
1970                                    JSR     R1,XCOUT
1971                                    NEXT
1972                            ;
1973                            XCOUT:  MOV     (S)+,IOCHR
1974                                    QIOW$C  IO.WVB!TF.WAL,4,4,,IOSTAT,,<IOCHR,1>
1975                                    RTS     R1
1976                            ;
1977                            ITERM:                  ; INITIALIZE RSX IF FIRST TIME THROUGH
1978                                    CMP     INTERM,#-1      ; FIRST TIME TERMINAL I/O?
1979                                    BNE     RSXRTS
1980                                    CLR     INTERM          ; YES
1981                                    ALUN$C  4,TI,0          ; ASSIGN LUN
1982                                    QIOW$C  IO.ATA,4,,,,,<AST1>     ; ATTACH - UNSOLICITED I/O
1983                                    SVTK$S  #TRAPV,#6       ; SET UP FOR TRAPS
1984                            RSXRTS: RTS     R1
1985                            INTERM: .WORD   -1              ; FLAG FOR FIRST TIME TERMINAL I/O.
1986                            ; DO NOT REINITIALIZE 'INTERM' AT COLD START, LEST ASSIGNMENT DONE TWICE.
1987                            IOCHR:  .WORD   0               ; TEMPORARY AREA FOR 'KEY' AND 'EMIT'
1988                            QFLAG:  .WORD   0               ; FOR '?TERMINAL'
1989                            XBUFF:  .BLKB   82.             ; TERMINAL BUFFER FOR RSX LINE I/O
1990                            ;
1991                            ;
1992                            ;
1993                                    HEAD    203,BYE,305,BYE                         ; ***** BYE (LOG OFF)
1994                                    CLOSE$  #FDBIO  ; CLOSE DISK I/O
1995                                    EXIT$S
1996                            ;
1997                            ;
1998                            ; ***************
1999                            ;
2000                            ; RSX11-M DISK I/O
2001                            ;
2002                            ; ***************
2003                            ;
2004                                    HEAD    204,XI/O,240,XIO                        ; ***** XI/O (RSX)
2005                            ;  PHYSICAL READ-WRITE
2006                            ;  ADDRESS  BLOCK# FLAG ==> REPORT.  READS OR WRITES A 512-BYTE BLOCK.
2007                            ;  FLAG 1=READ, 0=WRITE.  REPORT '0'=GOOD I/O, '1'=I/O ERROR.
2008                            ;  IF REPORT IS '1', THEN NEXT ON STACK IS '1'=OPEN ERROR,
2009                            ;  '2'=READ ERROR, '3'=WRITE ERROR, '4'=WAIT ERROR, '5'=ARGUMENT
2010                            ;  ERROR (FLAG NOT '0' OR '1').
2011                                    CLR     DSKERR  ; FOR I/O ERROR REPORT
2012                                    TST     OPENF   ; DISK FILE ALREADY OPENED?
2013                                    BNE     2$
2014                                    OPEN$M  #FDBIO
2015                                    BCC     2$
2016                                    MOV     #1,DSKERR       ; ERROR IN OPEN
2017                                    BR      ERRR
2018                            2$:     MOV     #1,OPENF        ; INDICATE FILE IS OPEN
```

```
2019                                    CLR     VIRBLK
2020                                    MOV     2(S),VIRBLK+2   ; SET UP VIRTUAL BLOCK NUMBER
2021                                    MOV     4(S),IOADDR     ; SET UP I/O ADDRESS
2022                                    TST     (S)             ; WAS TOP OF STACK - READ OR WRITE?
2023                                    BEQ     WRITE
2024                                    CMP     (S),#1
2025                                    BEQ     READ
2026                                    MOV     #5,DSKERR       ; ERROR, FLAG NOT EITHER '0' OR '1'
2027                                    BR      ERRR
2028                            READ:   READ$   #FDBIO,IOADDR,,#VIRBLK,#2
2029                                    BCC     WAIT
2030                                    MOV     #2,DSKERR       ; ERROR IN READ
2031                                    BR      ERRR
2032                            WRITE:  WRITE$  #FDBIO,IOADDR,,#VIRBLK,#2
2033                                    BCC     WAIT
2034                                    MOV     #3,DSKERR       ; ERROR IN WRITE
2035                                    BR      ERRR
2036                            WAIT:   WAIT$
2037                                    BCC     DONE
2038                                    MOV     #4,DSKERR       ; ERROR IN WAIT
2039                                    BR      ERRR
2040                            DONE:   ADD     #6,S
2041                                    CLR     -(S)            ; INDICATE GOOD I/O
2042                                    BR      DONE2
2043                            ERRR:   ADD     #6,S
2044                                    MOV     DSKERR,-(S)     ; RETURN THE ERROR INDICATOR
2045                                    MOV     #1,-(S)         ; INDICATE ERROR IN I/O
2046                            DONE2:  NEXT
2047                                    FSRSZ$  0
2048                            FDBIO:  FDBDF$
2049                                    FDRC$A  FD.RWM
2050                                    FDBK$A  ,512.,,2,IOSTAT
2051                                    FDOP$A  3,DESCR,,FO.MFY
2052                            DESCR:  .WORD   0,0             ; USE DEFAULT DEVICE
2053                                    .WORD   0,0             ; AND DIRECTORY.
2054                                    .WORD   FILSZ,FIL
2055                            FIL:    .ASCII  /FORTH.DAT/
2056                            FILSZ=.-FIL
2057                                    .EVEN
2058                            ;
2059                            OPENF:  .WORD   0               ; FLAG FOR FIRST TIME DISK I/O
2060                            ;  DO NOT INITIALIZE 'OPENF' AT COLD START
2061                            DSKERR: .WORD   0               ; SPACE FOR DISK ERROR MESSAGE
2062                            IOADDR: .WORD   0               ; ADDRESS FOR DISK READ/WRITE
2063                            IOSTAT: .BLKW   2               ; I/O STATUS REPORT
2064                            VIRBLK: .BLKW   2               ; VIRTUAL BLOCK NUMBER
2065                            ;
2066                                    HEAD    212,BLOCK-READ,240,BREAD,DOCOL          ; ***** BLOCK-READ
2067                            ; ( ADDRESS BLOCK# ==> REPORT).  REPORT: 0=GOOD READ, ELSE ERROR
2068                                    .WORD   ONE,XIO,SEMIS
2069                            ;
2070                                    HEAD    213,BLOCK-WRITE,305,BWRIT,DOCOL         ; ***** BLOCK-WRITE
2071                            ; ( ADDRESS BLOCK# ==> REPORT).  REPORT: 0=GOOD WRITE, ELSE ERROR
2072                                    .WORD   ZERO,XIO,SEMIS
2073                            ;
2074                                    HEAD    203,I/O,317,IO,DOCOL                    ; ***** I/O
2075                            ; READ OR WRITE 512-BYTE BLOCK, HANDLE ERRORS.
```

```
2076                            ; ( ADDRESS BLOCK# FLAG(1=READ,0=WRITE) ==> )
2077                                    .WORD   DUP,ONE,EQUAL,ZBRAN,XXS1-.,DROP,BREAD
2078                                    .WORD   ZBRAN,XXS2-.,CR,PDOTQ
2079                                    .BYTE   22
2080                                    .ASCII  /DISK READ ERROR # /
2081                                    .EVEN
2082                                    .WORD   DOT,ABORT
2083                            XXS2:   .WORD   BRAN,XXS3-.
2084                            XXS1:   .WORD   ZEQU,ZBRAN,XXS4-.,BWRIT,ZBRAN,XXS5-.
2085                                    .WORD   CR,PDOTQ
2086                                    .BYTE   23
2087                                    .ASCII  /DISK WRITE ERROR # /
2088                                    .EVEN
2089                                    .WORD   DOT,ABORT
2090                            XXS5:
2091                            XXS4:
2092                            XXS3:   .WORD   SEMIS
2093                            ;
2094                                    HEAD    203,R/W,327,RW,DOCOL                    ; ***** R/W
2095                            ; READ OR WRITE 1024-BYTE SCREEN.   ( ADDRESS SCREEN# FLAG ==> )
2096                            ; NOTE THAT SCREEN N IS BLOCKS 2N-1 AND 2N.
2097                                    .WORD   TOR,TWO,STAR,OVER,OVER,ONE,SUB,R,IO
2098                                    .WORD   SWAP,LIT,512.,PLUS
2099                                    .WORD   SWAP,FROMR,IO,SEMIS
2100                            ;
2101                                    .ENDC
2102                            ;
2103                            ; ***************
2104                            ;
2105                            ; RT-11 TERMINAL I/O
2106                            ;
2107                            ; ***************
2108                            ;
2109                                    .IFDF   RT11
2110                            ITERM:  CMP     INTERM,#-1
2111                                    BNE     RTRTS
2112                                    CLR     INTERM
2113                                    .RCTRLO                 ; RESET CNTL-O
2114                                    .TRPSET #TRAPBL,#TRAPZ
2115                            RTRTS:  RTS     R1
2116                            ;
2117                            PEMIT:
2118                                    JSR     R1,ITERM
2119                            ;  INCREMENT 'OUT', UNLESS A CONTROL CHARACTER BEING OUTPUT.
2120                                    CMP     (S),#40         ; TEST FOR CONTROL CHARACTER
2121                                    BLT     1$
2122                                    INC     42(U)           ; INCREMENT 'OUT'
2123                            1$:
2124                                    JSR     R1,XCOUT
2125                                    NEXT
2126                            ;
2127                            PKEY:   JSR     R1,ITERM
2128                                    .TTYIN
2129                                    BIC     #177600,R0
2130                                    CMP     R0,#12          ; IGNORE LINEFEED
2131                                    BEQ     PKEY
2132                                    MOV     R0,-(S)
```

```
2133                            NEXT
2134                    ;
2135                    PQTER:  JSR     R1,ITERM
2136                            MOV     44,-(RP)        ; SAVE JSW
2137                            BIS     #10100,44       ; SET BITS 6  AND 12 OF JSW
2138                            .TTINR
2139                            BCC     1$
2140                    ;  IF CARRY SET, NO CHARACTER - SUPPLY ZERO
2141                            CLR     R0
2142                    1$:     BIC     #177600,R0
2143                            CMP     R0,#12          ; IGNORE LINEFEED
2144                            BNE     2$
2145                            CLR     R0
2146                    2$:     MOV     R0,-(S)
2147                            MOV     (RP)+,44        ; RESTORE JSW
2148                            NEXT
2149                    ;
2150                    PCR:    JSR     R1,ITERM
2151                            MOV     #15,-(S)
2152                            JSR     R1,XCOUT
2153                            MOV     #12,-(S)
2154                            JSR     R1,XCOUT
2155                            NEXT
2156                    ;
2157                    XCOUT:  MOV     (S)+,R0
2158                            .TTYOUT
2159                            RTS     R1
2160                    ;
2161                    INTERM: .WORD   -1                      ; FLAG FOR FIRST TIME TERMINAL I/O
2162                    ;  DO NOT INITIALIZE 'INTERM' AT COLD START
2163                    IOCHR:  .WORD   0                       ; TEMPORARY AREA FOR 'KEY', 'EMIT'
2164                    ;
2165                            HEAD    203,BYE,305,BYE                         ; ***** BYE (RT)
2166                            .EXIT
2167                    ;
2168                    ;
2169                    ; ***************
2170                    ;
2171                    ; RT-11 DISK I/O
2172                    ;
2173                    ; ***************
2174                    ;
2175                            HEAD    204,XI/O,240,XIO                        ; ***** XI/O (RT)
2176                            CLR     DSKERR
2177                            TST     OPENF
2178                            BNE     RTOPEN
2179                            MOV     #1,OPENF        ; INDICATE FILE IS OPEN
2180                    ; NOW OPEN THE FILE
2181                            .SETTOP #-2
2182                            MOV     #RTSTAT,R1
2183                            .DSTATUS R1,#RTFILE
2184                            BCC     1$
2185                            MOV     #1,DSKERR
2186                            BR      RTRET
2187                    1$:     TST     4(R1)           ; HANDLER IN?
2188                            BNE     2$
2189                            .FETCH  HANDLR,#RTFILE
```

```
2190                            BCC     2$
2191                            MOV     #2,DSKERR
2192                            BR      RTRET
2193                    2$:     .LOOKUP #LOOK1,#0,#RTFILE
2194                            BCC     RTOPEN
2195                            MOV     #3,DSKERR
2196                            BR      RTRET
2197                    RTOPEN:     ; FILE IS OPEN - NOW READ IT
2198                            MOV     2(S),R1 ; BLOCK #
2199                            DEC     R1              ; BEGINS AT 1 IN FORTH
2200                            MOV     4(S),IOADDR     ; BUFFER ADDRESS
2201                            TST     (S)
2202                            BEQ     WRITE
2203                            CMP     (S),#1
2204                            BEQ     READ
2205                            MOV     #5,DSKERR
2206                            BR      RTRET
2207                    READ:   .READW  #RTBLK,#0,IOADDR,,R1
2208                            BCC     1$
2209                            MOV     #6,DSKERR
2210                    1$:     BR      RTRET
2211                    WRITE:  .WRITW  #RTBLK,#0,IOADDR,,R1
2212                            BCC     2$
2213                            MOV     #7,DSKERR
2214                    2$:     BR      RTRET
2215                    RTRET:  ADD     #6,S
2216                            MOV     DSKERR,-(S)
2217                            TST     DSKERR
2218                            BEQ     1$
2219                            MOV     #1,-(S)         ; INDICATE ERROR OCCURRED
2220                    1$:     NEXT
2221                    RTFILE: .RAD50  /DK FORTH DAT/
2222                    RTBLK:  .BYTE   0,10
2223                            .WORD   0,0,256.,0
2224                    ;
2225                    OPENF:  .WORD   0               ; FLAG FOR FIRST TIME DISK I/O
2226                    DSKERR: .WORD   0               ; SPACE FOR DISK ERROR MESSAGE
2227                    IOADDR: .WORD   0               ; ADDRESS FOR DISK READ/WRITE
2228                    RTSTAT: .BLKW   4               ; DISK I/O STATUS
2229                    LOOK1:  .BLKW   3               ; EMT ARGUMENT BLOCK
2230                    TRAPBL: .BLKW   2               ; EMT ARGUMENT BLOCK
2231                    ;
2232                            HEAD    212,BLOCK-READ,240,BREAD,DOCOL          ; ***** BLOCK-READ
2233                    ; ( ADDRESS BLOCK# ==> REPORT).  REPORT: 0=GOOD READ, ELSE ERROR
2234                            .WORD   ONE,XIO,SEMIS
2235                    ;
2236                            HEAD    213,BLOCK-WRITE,305,BWRIT,DOCOL         ; ***** BLOCK-WRITE
2237                    ; ( ADDRESS BLOCK# ==> REPORT).  REPORT: 0=GOOD WRITE, ELSE ERROR
2238                            .WORD   ZERO,XIO,SEMIS
2239                    ;
2240                            HEAD    203,I/O,317,IO,DOCOL                    ; ***** I/O
2241                    ; READ OR WRITE 512-BYTE BLOCK, HANDLE ERRORS.
2242                    ; ( ADDRESS BLOCK# FLAG(1=READ,0=WRITE) ==> )
2243                            .WORD   DUP,ONE,EQUAL,ZBRAN,XXS1-.,DROP,BREAD
2244                            .WORD   ZBRAN,XXS2-.,CR,PDOTQ
2245                            .BYTE   22
2246                            .ASCII  /DISK READ ERROR # /
```

```
2247                                            .EVEN
2248                                            .WORD   DOT,ABORT
2249                                    XXS2:   .WORD   BRAN,XXS3-.
2250                                    XXS1:   .WORD   ZEQU,ZBRAN,XXS4-.,BWRIT,ZBRAN,XXS5-.
2251                                            .WORD   CR,PDOTQ
2252                                            .BYTE   23
2253                                            .ASCII  /DISK WRITE ERROR # /
2254                                            .EVEN
2255                                            .WORD   DOT,ABORT
2256                                    XXS5:
2257                                    XXS4:
2258                                    XXS3:   .WORD   SEMIS
2259                                    ;
2260                                            HEAD    203,R/W,327,RW,DOCOL                    ; ***** R/W
2261                                    ; READ OR WRITE 1024-BYTE SCREEN.   ( ADDRESS SCREEN# FLAG ==> )
2262                                    ; NOTE THAT SCREEN N IS BLOCKS 2N-1 AND 2N.
2263                                            .WORD   TOR,TWO,STAR,OVER,OVER,ONE,SUB,R,IO
2264                                            .WORD   SWAP,LIT,512.,PLUS
2265                                            .WORD   SWAP,FROMR,IO,SEMIS
2266                                    ;
2267                                    ;
2268                                            .ENDC
2269                                    ;
2270                                    ; ***************
2271                                    ;
2272                                    ; STAND-ALONE TERMINAL I/O
2273                                    ;
2274                                    ; ****************
2275                                    ;
2276                                            .IFDF   ALONE
2277 013574                            PEMIT:
2278                                    ;  INCREMENT 'OUT', UNLESS A CONTROL CHARACTER BEING OUTPUT.
2279 013574  021527  000040                       CMP     (S),#40         ; TEST FOR CONTROL CHARACTER
2280 013600  002402                               BLT     1$
2281 013602  005263  000042                       INC     42(U)           ; INCREMENT 'OUT'
2282 013606  005737  177564            1$:      TST     @#177564
2283 013612  001775                               BEQ     1$
2284 013614  012537  177566                       MOV     (S)+,@#177566
2285 013620                                       NEXT
2286                                    ;                       TSTB
2287 013624  005737  177560            PKEY:    TST     @#177560
2288 013630  001775                               BEQ     PKEY
2289 013632  005037  177560                       CLR     @#177560
2290 013636  113701  177562                       MOVB    @#177562,R1
2291 013642  042701  177600                       BIC     #177600,R1
2292 013646  022701  000177                       CMP     #177,R1
2293 013652  001002                               BNE     1$
2294 013654  012701  000010                       MOV     #10,R1
2295 013660  010145                      1$:      MOV     R1,-(S)
2296 013662                                       NEXT
2297                                    ;                       TSTB
2298 013666  005737  177560            PQTER:   TST     @#177560
2299 013672  001403                               BEQ     1$
2300 013674  013745  177562                       MOV     @#177562,-(S)
2301 013700  000401                               BR      2$
2302 013702  005045                      1$:      CLR     -(S)
2303 013704  005037  177560            2$:      CLR     @#177560
```

```
2304 013710                                    NEXT
2305                                ;
2306 013714  005737  177564        PCR:        TST     @#177564
2307 013720  001775                            BEQ     PCR
2308 013722  012737  000015  177566            MOV     #15,@#177566
2309 013730  005737  177564        1$:         TST     @#177564
2310 013734  001775                            BEQ     1$
2311 013736  012737  000012  177566            MOV     #12,@#177566
2312 013744                                    NEXT
2313                                ;
2314 013750                                    HEAD    203,BYE,305,BYE                         ; ***** BYE (ALONE)
2315
2316 013760  000000                            HALT
2317                                ;
2318                                ; ***************
2319                                ;
2320                                ;  STAND-ALONE DISK I/O
2321                                ;
2322                                ; ***************
2323                                ;
2324         177170                 RXCS=177170     ; CONTROL AND STATUS REGISTER
2325         177172                 RXDB=177172     ; DATA BUFFER REGISTER
2326                                ;
2327                                ;
2328 013762                                    HEAD    204,NRTS,240,NRTS                       ; ***** NRTS
2329                                ;  ADDRN TRN SECN...ADDR1 TR1 SEC1 N -> FLAG
2330                                ;  READ N SECTORS.   USES R0, R1, R2
2331                                ;  THIS OPERATION IS IN CODE TO KEEP UP WITH DISK TIMING FOR
2332                                ;  STANDARD PDP-11 SECTOR SKEWING.
2333 013774  012501                        MOV     (S)+,R1         ; # OF SECTORS TO READ
2334 013776  012702  000010        1$:     MOV     #10,R2          ; RETRY COUNT
2335 014002  012700  000007        21$:    MOV     #7,R0           ; 'READ' COMMAND
2336 014006  004767  000172                JSR     PC,DRIV2        ; ADJUST R0 COMMAND
2337                                                               ; IF SECOND DRIVE
2338 014012  010037  177170                MOV     R0,@#RXCS               ; READ COMMAND
2339 014016  032737  000200  177170  2$:    BIT     #200,@#RXCS     ; WAIT FOR TRANSFER FLAG
2340 014024  001774                        BEQ     2$
2341 014026  011537  177172                MOV     (S),@#RXDB      ; SECTOR #
2342 014032  032737  000200  177170  3$:    BIT     #200,@#RXCS     ; WAIT FOR TRANSFER FLAG
2343 014040  001774                        BEQ     3$
2344 014042  016537  000002  177172        MOV     2(S),@#RXDB     ; TRACK #
2345 014050  032737  000040  177170  4$:    BIT     #40,@#RXCS      ; WAIT FOR DONE FLAG
2346 014056  001774                        BEQ     4$
2347                                ;  CHECK FOR ERROR
2348 014060  005737  177170                TST     @#RXCS
2349 014064  002423                        BLT     20$             ; ERROR
2350                                ;
2351 014066  012737  000003  177170        MOV     #3,@#RXCS       ; 'EMPTY' COMMAND
2352                                ;  EMPTY THE CONTROLLER'S BUFFER
2353 014074  016500  000004                MOV     4(S),R0         ; ADDRESS TO RECEIVE DATA
2354 014100  012745  000200                MOV     #200,-(S)       ; COUNT OF TIMES TO LOOP
2355 014104  032737  000200  177170  6$:    BIT     #200,@#RXCS     ; WAIT FOR TRANSFER FLAG
2356 014112  001774                        BEQ     6$
2357 014114  113720  177172                MOVB    @#RXDB,(R0)+
2358 014120  005315                        DEC     (S)             ; DECREMENT THE COUNT
2359 014122  001370                        BNE     6$
2360 014124  005725                        TST     (S)+            ; POP THE COUNT
```

```
2361                                    ;  CHECK FOR ERROR
2362 014126  005737  177170                    TST     @#RXCS
2363 014132  002015                            BGE     7$
2364 014134                             20$:   ;  ERROR, SO RE-TRY
2365 014134  012737  040000  177170            MOV     #40000,@#RXCS   ;  CLEAR ERROR STATUS
2366 014142  032737  000040  177170  22$:      BIT     #40,@#RXCS
2367 014150  001774                            BEQ     22$
2368 014152  005302                            DEC     R2              ;  RE-TRY COUNT
2369 014154  003312                            BGT     21$
2370 014156  012745  177777                    MOV     #-1,-(S)
2371 014162                                    NEXT                    ;  ERROR EXIT
2372 014166  062705  000006            7$:     ADD     #6,S            ;  GOOD READ, SO POP THE 3 ARGS
2373 014172  005301                            DEC     R1
2374 014174  001300                            BNE     1$              ;  LOOP UNLESS ALL SECTORS READ
2375                                    ;
2376 014176  005045                            CLR     -(S)            ;  GOOD-READ INDICATOR
2377 014200                                    NEXT                    ;  EXIT
2378                                    ;
2379                                    ;  SUBROUTINE TO ADJUST COMMAND FOR SECOND DISK DRIVE
2380                                    ;  NOTE - 'NWTS' ALSO USES THIS SUBROUTINE.
2381                                    ;  NOTE USE OF R0, AND OF FORTH STACK.
2382 014204  026527  000002  000114    DRIV2:  CMP     2(S),#114       ;  TRACK > 76 ?
2383 014212  003405                            BLE     10$
2384 014214  162765  000115  000002            SUB     #115,2(S)       ;  SUBTRACT 77
2385 014222  052700  000020                    BIS     #20,R0          ;  SET UNIT-SELECT BIT
2386 014226  000207                    10$:    RTS     PC
2387                                    ;
2388 014230                                    HEAD    204,NWTS,240,NWTS                       ;  ***** NWTS
2389                                    ;  ADDRN TRN SECN...ADDR1 TR1 SEC1 N -> FLAG
2390                                    ;  WRITE N SECTORS.  USES R0, R1, R2.
2391 014242  012501                            MOV     (S)+,R1         ;  # OF SECTORS TO BE WRITTEN
2392 014244  012702  000010            1$:     MOV     #10,R2          ;  RE-TRY COUNT
2393 014250  012737  000001  177170    21$:    MOV     #1,@#RXCS       ;  'FILL' COMMAND
2394 014256  032737  000200  177170    2$:     BIT     #200,@#RXCS     ;  WAIT FOR TRANSFER FLAG
2395 014264  001774                            BEQ     2$
2396                                    ;  NOW FILL THE BUFFER
2397 014266  016500  000004                    MOV     4(S),R0
2398 014272  012745  000200                    MOV     #200,-(S)       ;  COUNT
2399 014276  032737  000200  177170    3$:     BIT     #200,@#RXCS     ;  WAIT FOR TRANSFER FLAG
2400 014304  001774                            BEQ     3$
2401 014306  112037  177172                    MOVB    (R0)+,@#RXDB    ;  MOVE ONE BYTE
2402 014312  005315                            DEC     (S)
2403 014314  001370                            BNE     3$
2404 014316  005725                            TST     (S)+            ;  POP STACK
2405                                    ;  CHECK FOR ERROR
2406 014320  005737  177170                    TST     @#RXCS
2407 014324  002432                            BLT     20$             ;  ERROR
2408                                    ;
2409 014326  012700  000005                    MOV     #5,R0           ;  'WRITE' COMMAND
2410 014332  004767  177646                    JSR     PC,DRIV2        ;  ADJUST IF SECOND DRIVE
2411 014336  010037  177170                    MOV     R0,@#RXCS               ;  'WRITE' COMMAND
2412 014342  032737  000200  177170    5$:     BIT     #200,@#RXCS     ;  WAIT FOR TRANSFER FLAG
2413 014350  001774                            BEQ     5$
2414 014352  011537  177172                    MOV     (S),@#RXDB      ;  MOVE SECTOR #
2415 014356  032737  000200  177170    6$:     BIT     #200,@#RXCS     ;  WAIT FOR TRANSFER FLAG
2416 014364  001774                            BEQ     6$
2417 014366  016537  000002  177172            MOV     2(S),@#RXDB     ;  MOVE TRACK #
```

```
2418 014374  032737  000040  177170  7$:     BIT     #40,@#RXCS      ; WAIT FOR DONE FLAG
2419 014402  001774                          BEQ     7$
2420                                 ;  CHECK FOR ERROR
2421 014404  005737  177170                  TST     @#RXCS
2422 014410  002015                          BGE     10$
2423 014412                          20$:    ; ERROR SO RE-TRY
2424 014412  012737  040000  177170          MOV     #40000,@#RXCS   ; CLEAR ERROR STATUS
2425 014420  032737  000040  177170  22$:    BIT     #40,@#RXCS
2426 014426  001774                          BEQ     22$
2427 014430  005302                          DEC     R2              ; RE-TRY COUNT
2428 014432  003306                          BGT     21$
2429 014434  012745  177777                  MOV     #-1,-(S)        ; ERROR INDICATOR
2430 014440                                  NEXT                    ; EXIT
2431 014444  062705  000006          10$:    ADD     #6,S            ; GOOD WRITE, SO POP THE 3 ARGS
2432 014450  005301                          DEC     R1
2433 014452  001274                          BNE     1$              ; LOOP UNLESS ALL SECTORS WRITTEN
2434                                 ;
2435 014454  005045                          CLR     -(S)            ; GOOD-WRITE INDICATOR
2436 014456                                  NEXT
2437                                 ;
2438 014462                                  HEAD    203,RTS,323,RTS,DOCOL                   ; ***** RTS
2439                                 ;  ADDR TR SEC ->
2440                                 ;  READ A SINGLE SECTOR.
2441 014472  002520' 013772' 000144'         .WORD   ONE,NRTS,ZBRAN,1$-.,PDOTQ
     014500  000036  005104'
2442 014504     026                          .BYTE   26
2443 014505     104     111     123          .ASCII  /DISK READ ERROR IN RTS/
     014510     113     040     122
     014513     105     101     104
     014516     040     105     122
     014521     122     117     122
     014524     040     111     116
     014527     040     122     124
     014532     123
2444                                         .EVEN
2445 014534  007574'                         .WORD   QUIT
2446 014536  001446'                 1$:     .WORD   SEMIS
2447                                 ;
2448 014540                                  HEAD    203,WTS,323,WTS,DOCOL                   ; ***** WTS
2449                                 ;  ADDR TR SEC ->
2450                                 ;  WRITE A SINGLE SECTOR.
2451 014550  002520' 014240' 000144'         .WORD   ONE,NWTS,ZBRAN,1$-.,PDOTQ
     014556  000040  005104'
2452 014562     030                          .BYTE   30
2453 014563     104     111     123          .ASCII  /DISK WRITE ERROR IN WTS /
     014566     113     040     127
     014571     122     111     124
     014574     105     040     105
     014577     122     122     117
     014602     122     040     111
     014605     116     040     127
     014610     124     123     040
2454                                         .EVEN
2455 014614  007574'                         .WORD   QUIT
2456 014616  001446'                 1$:     .WORD   SEMIS
2457                                 ;
2458                                 ;
```

```
2459 014620                                         HEAD    205,SKEW1,261,SKEW1,DOCOL               ; ***** SKEW1
2460                                        ;  SEQUENCE -> TRACK SECTOR
2461                                        ;  HANDLE THE SECTOR SKEWING.
2462                                        ;  NOTE - 'SEQUENCE #' IS ZERO-ORIGIN INDEX OF SECTOR (SKEWED).
2463                                        ;  NOTE - 'SKEW1' DOES SKEWING OF ONLY ONE DRIVE;  'SKEW' GENERALIZES
2464                                        ;  'SKEW1' TO BOTH DRIVES.
2465 014632  002054' 000060' 000032                .WORD   DUP,LIT,32,SLASH,SWAP
      014640  010514' 002026'
2466 014644  001762' 000060' 000006                .WORD   OVER,LIT,6,STAR,OVER,DUP,PLUS,PLUS,SWAP
      014652  010454' 001762' 002054'
      014660  001634' 001634' 002026'
2467 014666  000060' 000032  010534'                .WORD       LIT,32,MOD,LIT,15,SLASH,PLUS
      014674  000060' 000015  010514'
      014702  001634'
2468 014704  000060' 000032  010534'                .WORD       LIT,32,MOD,ONEP
      014712  003322'
2469 014714  002026' 003322' 002026'                .WORD   SWAP,ONEP,SWAP,SEMIS
      014722  001446'
2470                                        ;
2471 014724                                         HEAD    206,S-SKIP,240,SSKIP,DOVAR              ; ***** S-SKIP
2472                                        ;  VARIABLE - NUMBER OF SECTORS SKIPPED AT BEGINNING OF DISK.
2473                                        ;  DEFAULT IS 56 DECIMAL (SKIP AN RT-11 DIRECTORY).  ALSO, TRACK
2474                                        ;  ZERO IS SKIPPED, FOR COMPATIBILITY.
2475 014740  000070                                .WORD   56.
2476                                        ;
2477 014742                                         HEAD    206,S-USED,240,SUSED,DOVAR              ; ***** S-USED
2478                                        ;  VARIABLE - NUMBER OF SECTORS USED ON ON ONE DISK.
2479                                        ;  NORMALLY,  S-USED + S-SKIP = 1976.  (2002.  - 26.  OF TR 0).
2480 014756  003600                                .WORD   1920.
2481                                        ;
2482 014760                                         HEAD    204,SKEW,240,SKEW,DOCOL                 ; ***** SKEW
2483                                        ;  SEQUENCE# -> TRACK SECTOR
2484 014772  002054' 003322' 014754'                .WORD   DUP,ONEP,SUSED,AT,GREAT,ZBRAN,1$-.
      015000  002176' 003532' 000144'
      015006  000036
2485 015010  014754' 002176' 003436'                .WORD   SUSED,AT,SUB,SSKIP,AT,PLUS,SKEW1
      015016  014736' 002176' 001634'
      015024  014630'
2486 015026  002026' 000060' 000115                .WORD   SWAP,LIT,77.,PLUS,SWAP
      015034  001634' 002026'
2487 015040  000122' 000012                       .WORD   BRAN,2$-.
2488 015044  014736' 002176' 001634' 1$:          .WORD   SSKIP,AT,PLUS,SKEW1
      015052  014630'
2489 015054  001446'                        2$:          .WORD   SEMIS
2490                                        ;
2491 015056                                         HEAD    206,NSETUP,240,NSET,DOCOL               ; ***** NSETUP
2492                                        ;  ADDR SEQUENCE# N -> ADDRN TRN SECN...ADDR1 TR1 SEC1
2493                                        ;  THIS PREPARES A WHOLE SCREEN (IF N=8) FOR 'NRTS' OR 'NWTS'.
2494 015072  003564' 001762' 000060'                .WORD   ROT,OVER,LIT,128.,STAR,PLUS,ROT,ROT
      015100  000200  010454' 001634'
      015106  003564' 003564'
2495 015112  001762' 001634' 002520'                .WORD   OVER,PLUS,ONE,SUB,SWAP,ONE,SUB,SWAP
      015120  003436' 002026' 002520'
      015126  003436' 002026'
2496 015132  000334'                               .WORD   XDO
2497 015134  000060' 000200  003436' 2$:          .WORD   LIT,128.,SUB,DUP,I,SKEW,ROT
      015142  002054' 000360' 014770'
```

```
     015150  003564'
2498 015152  000060' 177777  000250'          .WORD   LIT,-1,XPLOO,2$-.
     015160  177754
2499 015162  002004' 001446'                  .WORD   DROP,SEMIS
2500                                    ;
2501 015166                                   HEAD    203,R/W,327,RW,DOCOL                    ; ***** R/W
2502                                    ; READ OR WRITE 1024-BYTE SCREEN.
2503                                    ;  ADDR BLOCK# FLAG(R=1,W=0) ->
2504 015176  001506' 002520' 003436'          .WORD   TOR,ONE,SUB,LIT,8.,STAR,FROMR
     015204  000060' 000010  010454'
     015212  001524'
2505                                    ;  CHANGE THE SCREEN # TO FIRST SEQUENCE #.
2506                                    ;  IF READ, SETUP AND READ 8 SECTORS
2507 015214  000144' 000054  000060'          .WORD   ZBRAN,1$-.,LIT,8.,NSET,LIT,8.,NRTS
     015222  000010  015070' 000060'
     015230  000010  013772'
2508 015234  000144' 000030  005104'          .WORD   ZBRAN,2$-.,PDOTQ
2509 015242     020                          .BYTE   20
2510 015243     104     111     123          .ASCII  /DISK READ ERROR /
     015246     113     040     122
     015251     105     101     104
     015254     040     105     122
     015257     122     117     122
     015262     040
2511                                          .EVEN
2512 015264  007574'                         .WORD   QUIT
2513 015266  000122' 000050           2$:    .WORD   BRAN,3$-.
2514                                    ;  SETUP AND WRITE 8 SECTORS
2515 015272  000060' 000010  015070' 1$:    .WORD   LIT,8.,NSET,LIT,8.,NWTS
     015300  000060' 000010  014240'
2516 015306  000144' 000030  005104'          .WORD   ZBRAN,4$-.,PDOTQ
2517 015314     020                          .BYTE   20
2518 015315     104     111     123          .ASCII  /DISK WRITE ERROR/
     015320     113     040     127
     015323     122     111     124
     015326     105     040     105
     015331     122     122     117
     015334     122
2519                                          .EVEN
2520 015336  007574'                         .WORD   QUIT
2521 015340                             4$:
2522 015340  001446'                  3$:    .WORD   SEMIS
2523                                    ;
2524                                          .ENDC
2525                                    ;
2526                                    ; ***************
2527                                    ;
2528                                    ;       TRAP RECOVERY SECTION, RSX-11M
2529                                    ;
2530                                    ; ***************
2531                                    ;
2532                                          .IFNDF  ALONE   ; STAND-ALONE MUST HANDLE OWN INTERRUPTS.
2533                                          HEAD    205,TRAPS,323,TRAPS,DOCOL               ; ***** TRAPS
2534                                          .WORD   CR,PDOTQ
2535                                          .BYTE   14
2536                                          .ASCII  /TRAP-ERROR,
2537                                          .EVEN
```

```
2538                            .WORD   DOT,SWAP,UDOT,UDOT,QUIT
2539                            .ENDC
2540                    ;
2541                            .IFDF   RSX11
2542                    TRAPV:  .WORD   TRAP0,TRAP1,TRAP2,TRAP3,TRAP4,TRAP5
2543                    TRAP0:  CLR     R1                      ; TRAP # 0
2544                            BR      TRAPZ
2545                    TRAP1:  MOV     #1,R1                   ; TRAP # 1
2546                            ADD     #6,SP                   ; DROP MMU INFO
2547                            BR      TRAPZ
2548                    TRAP2:  MOV     #2,R1
2549                            BR      TRAPZ
2550                    TRAP3:  MOV     #3,R1
2551                            BR      TRAPZ
2552                    TRAP4:  MOV     #4,R1
2553                            BR      TRAPZ
2554                    TRAP5:  MOV     #5,R1
2555                            BR      TRAPZ
2556                    ;  TO RETURN FROM TRAP HANDLER, SET UP STACK, ETC. FOR FORTH 'TRAPS'
2557                    ;  DON'T USE RTT OR RTI.
2558                    TRAPZ:  MOV     (SP)+,-(S)      ; PC
2559                            MOV     (SP)+,-(S)      ; PS
2560                            MOV     R1,-(S)         ; TRAP #
2561                            MOV     #TRAPS+2,IP     ; EXECUTE 'TRAPS'
2562                            NEXT
2563                    ;
2564                            .ENDC
2565                    ; ****************
2566                    ;
2567                    ;  TRAP RECOVERY SECTION, RT-11
2568                    ;
2569                    ; ****************
2570                            .IFDF   RT11
2571                    TRAPZ:  BCS     1$
2572                    ;  IF CARRY CLEAR, TRAP 4
2573                            MOV     #4,R1
2574                            BR      2$
2575                    1$:     MOV     #10,R1
2576                    2$:     MOV     (SP)+,-(S)              ; PC
2577                            MOV     (SP),-(S)               ; PS
2578                            MOV     R1,-(S)                 ; TRAP #
2579                            MOV     #3$,-(SP)               ; SO RTI WILL RESTORE PC TO '3$'
2580                            .TRPSET #TRAPBL,#TRAPZ          ; RE-SET TRAPS
2581                            RTI
2582                    3$:     MOV     #TRAPS+2,IP             ; EXECUTE 'TRAPS'
2583                            NEXT
2584                            .ENDC
2585                    ;
2586                    ;
```

```
                                ; NOTE - '.W' ('DW') IS USED ONLY FOR TESTING - TO GET OCTAL OUTPUT
                                ;        WHEN '.' IS NOT WORKING DURING SYSTEM DEVELOPMENT.
                                ;
                                ;
                                ;       HEAD    202,.W,240,DW                           ; ***** .W
                                ;       MOV     (S),XOUT
                                ;       ROL     XOUT
                                ;       ROL     XOUT
                                ;       MOV     XOUT,IOCHR
                                ;       ROR     XOUT
                                ;       BIC     #177776,IOCHR
                                ;       ADD     #60,IOCHR
                                ;       MOV     IOCHR,-(S)
                                ;       JSR     R1,XCOUT
                                ;       MOV     #5,XCOUNT
                                ;XLP:   ROL     XOUT
                                ;       ROL     XOUT
                                ;       ROL     XOUT
                                ;       ROL     XOUT
                                ;       MOV     XOUT,IOCHR
                                ;       ROR     XOUT
                                ;       BIC     #177770,IOCHR
                                ;       ADD     #60,IOCHR
                                ;       MOV     IOCHR,-(S)
                                ;       JSR     R1,XCOUT
                                ;       DEC     XCOUNT
                                ;       BNE     XLP
                                ;       MOV     #40,IOCHR
                                ;       MOV     IOCHR,-(S)
                                ;       JSR     R1,XCOUT
                                ;       NEXT
                                ;XOUT:  .WORD   0
                                ;XCOUT: .WORD   0
                                ;
                                ;
                                ;
                                ;
                                ;
                                ; ***************************************************************
                                ;
                                ;  THE FOLLOWING TWO DEFINITIONS ARE NOT PURE CODE, SO THEY WERE
                                ;  MOVED HERE, NEAR THE END OF THE DICTIONARY.
                                ;
                                ; ***************************************************************
                                ;
015342                                  HEAD    305,<;CODE>,305,SEMIC,DOCOL             ; ***** ;CODE
                                ;  CREATE NEW DATA TYPE WITH CODE ROUTINE WRITTEN IN ASSEMBLY.
015354  004256' 004346' 004562'       .WORD   QCSP,COMP,PSCOD,LBRAC,SMUDQ,SEMIS
015362  004374' 004440' 001446'
                                ;  NOTE: LATER, THE ASSEMBLER WILL PATCH THIS DEFINITION.
                                ;
015370                                  HEAD    305,FORTH,310,FORTH,DODOE               ; ***** FORTH
015402  007504'                         .WORD   DOVOC
015404  120201                         .WORD   120201          ; DUMMY HEADER AT INTERSECTION
015406  015412'                         .WORD   TASK-10
015410  000000                  XXVOC:  .WORD   0               ; THE VOCABULARY LINK (FOR FUTURE USE)
015412                                  HEAD    204,TASK,240,TASK,DOCOL                 ; ***** TASK
```

```
2644 015424  001446'                          .WORD   SEMIS
2645                                ;
2646                                ;
2647                                ;
2648                                ;
```

```
                                ;*****************************************************************
                                ;
                                ;  STACKS AND BUFFERS
                                ;
                                ;*****************************************************************
                                ;
                                ;  NOTE - 'UP', 'OPENF', 'INTERM', AND DISK BUFFERS ARE
                                ;   INITIALIZED AT COLD START, OR AT FIRST TIME THROUGH.
                                        .EVEN
015426                          XDP:                            ; DICTIONARY STARTS HERE
015426                                  .BLKB   8000.           ; FOR DICTIONARY AND COMP. STACK
                                ; INCREASE THIS NUMBER TO USE A LARGER MEMORY SIZE.
035126                          XS0:                            ; START OF COMPUTATION STACK
035126                                  .BLKW   2               ; IN CASE OF EMPTY STACK
                                ;
                                ;
                                ;
                                ;
                                ;
035132                          DSKBUF:         ; ROOM  FOR 3 1K DISK BUFFERS
                                ; INITIALIZE BUFFERS' UPDATE BITS, AND TERMINATING NULLS, TO ZERO.
                                ; NOTE - THESE BUFFERS ARE CLEARED AT COLD START, ANYWAY,
                                ; BECAUSE A STAND-ALONE BOOT MAY NOT INITIALIZE HIGH MEMORY;
                                ; AND ALSO SO THAT THE NUMBER OR LOCATION OF BUFFERS CAN BE
                                ; CHANGED AT RUN TIME.
035132  000000                          .WORD   0
035134                                  .BLKB   1024.
037134  000000                          .WORD   0
037136  000000                          .WORD   0
037140                                  .BLKB   1024.
041140  000000                          .WORD   0
041142  000000                          .WORD   0
041144                                  .BLKB   1024.
043144  000000                          .WORD   0
043146                          ENDBUF:         ; CAUTION - 'ENDBUF' - 'DSKBUF' MUST BE EXACT MULTIPLE
                                ; OF THE BUFFER LENGTH PLUS 4.
                                ;
                                ;
                                ;
                                ;
                                ;
                                ; 'XTIB', 'XR0', AND 'XUP' ARE ONLY USED IN BOOT-UP TABLE;
                                ;  THEREFORE THE AREAS DEFINED HERE CAN BE MOVEDAT RUN TIME.
043146                          XTIB:   .BLKW   42.             ; TERMINAL INPUT BUFFER
043272                                  .BLKW   50.             ; FOR RETURN STACK
        043436'                 XR0=.
043436                          XUP:    .BLKW   100             ; ROOM FOR 100 USER VARIABLES
                                ;
                                ;
                                        .IFDF   RT11
                                ;  DISK HANDLER GOES HERE
                                HANDLR: .WORD   .+2
                                        .ENDC
                                ;
                                ;
                                ;  NOTE - CHANGE THE FOLLOWING LINE TO '.END' IF LINKING TO OTHER LANGUAGES.
        000000'                         .END    ORIGIN
```

SYMBOL TABLE

```
ABORT   007652R     DOES    004632R     INDEX   013320R     PORIG   002632R     THEN    012214R
ABS     010134R     DOT     013144R     INTER   007274R     PGTER   013666R     THREE   002540R
AGAIN   012400R     DOTG    005140R     IP     =%000004     PREV    003310R     TIB     002676R
ALLOT   003402R     DOTR    013104R     KEY     001030R     PSCOD   004562R     TICK    011764R
ALONE = 000001      DOUSE   002474R     LATES   003742R     PSTOR   002072R     TOGGL   002124R
AND     001302R     DOVAR   002450R     LBRAC   004374R     QALIG   005226R     TOR     001506R
ARROW   011724R     DOVOC   007504R     LEAVE   001466R     QCOMP   004152R     TRAV    003674R
AT      002176R     DP      002756R     LESS    003502R     QCSP    004256R     TRIAD   013400R
BACK    012114R     DPL     003212R     LFA     003762R     QERR    004120R     TWO     002530R
BASE    003200R     DPLUS   001652R     LIMIT   003026R     QEXEC   004202R     TWOP    003340R
BBUF    002600R     DRIV2   014204R     LINK  = 015412R     QLOAD   004314R     TYPE    004714R
BCOMP   007032R     DRONE   011156R     LINK2 = 015412R     QPAIR   004232R     U      =%000003
BDIGS   012634R     DROP    002004R     LIST    013216R     QSTAC   007214R     UDIV    001226R
BEGIN   012136R     DRZER   011136R     LIT     000060R     QTERM   001046R     UDOT    013176R
BL      002552R     DSKBUF  035132R     LITER   007066R     QUERY   005446R     ULESS   007154R
BLANK   005654R     DTRAI   005020R     LOAD    011642R     QUEST   013160R     UMULT   001144R
BLK     003040R     DUP     002054R     LOOP    012256R     QUIT    007574R     UNTIL   012336R
BLOCK   011314R     ECELL   004770R     MAX     010236R     R       001540R     UPDAT   010740R
BRAN    000122R     EDIGS   012654R     MESS    011550R     RBRAC   004412R     UPPER   006416R
BSCR    002614R     ELSE    012512R     MIN     010206R     REPEAT  012430R     USE     003274R
BUFFE   011204R     EMIT    001020R     MINUS   001712R     RNUM    003250R     USER    002466R
BUILD   004612R     ENCL    000676R     MLOOP   000566R     ROT     003564R     USLAS   001214R
BYE     013756R     ENC1    000712R     MOD     010534R     RP     =%000006     USTAR   001132R
CAT     002216R     ENC2    000726R     MSLAS   010352R     RPSTO   001426R     VAR     002442R
CENT    007730R     ENC3    000744R     MSMOD   010620R     RTS     014470R     VLIST   013470R
CFA     004002R     ENC4    000772R     MSTAR   010266R     RW      015174R     VOCAB   007444R
CL      002564R     END     012362R     MTBUF   011006R     RXCS  = 177170      VOCL    002776R
CMOVE   001070R     ENDBUF  043146R     NFA     004020R     RXDB  = 177172      W      =%000002
COLD    007726R     ENDIF   012160R     NOFAST  000560R     RZERO   002664R     WARN    002730R
COLON   002314R     EQUAL   003452R     NOFST1  000604R     S      =%000005     WENT    010012R
COMMA   003416R     ERASE   005632R     NRTS    013772R     SCR     003076R     WHILE   012554R
COMP    004346R     ERRDR   006530R     NSET    015070R     SCSP    004074R     WIDTH   002712R
CON     002406R     EXEC    000102R     NULL    005476R     SEMI    002354R     WORD    005750R
CONT    003132R     EXPEC   005254R     NUMB    006206R     SEMIC   015352R     WTS     014546R
COUNT   004670R     FAILED  000646R     NWTS    014240R     SEMIS   001446R     W2      010016R
CR      001056R     FAST    000532R     OCTAL   004534R     SIGN    012706R     XDO     000334R
CREAT   006700R     FCOMP   000532R     OFSET   003114R     SKEW    014770R     XDP     015426R
CSP     003236R     FENCE   002744R     ONE     002520R     SKEW1   014630R     XLOOP   000202R
CSTOR   002270R     FILL    005572R     ONEP    003322R     SLASH   010514R     XMATCH  000544R
CURR    003150R     FIRST   003012R     OR      001322R     SLMOD   010474R     XOR     001340R
DABS    010162R     FLD     003224R     ORIGIN  000000R     SMUDG   004440R     XPLOO   000250R
DDOT    013126R     FLUSH   011040R     OUT     003064R     SPACE   003622R     XR0   = 043436R
DDOTR   013040R     FORGE   012016R     OVER    001762R     SPACS   012576R     XS0     035126R
DDUP    003644R     FORTH   015400R     PABOR   006512R     SPAT    001366R     XTIB    043146R
DEC     004510R     FROMR   001524R     PAD     005724R     SPSTO   001406R     XUP     043436R
DEFIN   007532R     GFORTH  000000RG    PAREN   007552R     SSKIP   014736R     XXB1    007606R
DFIND   006336R     GO      010056R     PBUF    010656R     SSLA    010576R     XXB2    007636R
DIG     012734R     GREAT   003532R     PCR     013714R     SSMOD   010554R     XXD2    006726R
DIGIT   000400R     HERE    003362R     PDOTG   005104R     STAR    010454R     XXD6    007106R
DIGS    013010R     HEX     004462R     PEMIT   013574R     STATE   003164R     XXEA    007340R
DLINE   011524R     HLD     003262R     PFA     004046R     STOD    010106R     XXE1    006472R
DLITE   007124R     HOLD    005674R     PFIND   000506R     STORE   002242R     XXE2    006430R
DMINU   001734R     I       000360R     PKEY    013624R     SUB     003436R     XXE3    006404R
DO      012230R     IDDOT   006610R     PLINE   011462R     SUSED   014754R     XXE4    007276R
DOCOL   002340R     IF      012462R     PLOOP   012306R     SWAP    002026R     XXE5    007326R
DOCON   002420R     IMMED   007412R     PLUS    001634R     SZERO   002652R     XXE6    007332R
DODOE   004646R     IN      003052R     PNUMB   006072R     TASK    015422R     XXE7    007372R
```

SYMBOL TABLE

XXFA	006324R	XXK1	005266R	XXN4	006546R	XXT1	010714R	XXW6	005030R
XXF3	006074R	XXK2	005346R	XXN5	007144R	XXT2	011216R	XXW7	005062R
XXF4	007364R	XXK3	005424R	XXRA	012616R	XXT3	011264R	XXZ1	013256R
XXF5	007370R	XXK4	005376R	XXRB	010176R	XXT4	011440R	XXZ2	013332R
XXF6	006244R	XXK5	005400R	XXR1	012726R	XXT5	011354R	XXZ3	013362R
XXF7	006312R	XXL1	004734R	XXR2	012770R	XXT6	011406R	XXZ4	013432R
XXG4	006170R	XXL2	004752R	XXR3	013012R	XXT7	011110R	XXZ5	013510R
XXG5	006162R	XXL3	004754R	XXR4	012624R	XXVOC	015410R	XXZ6	013536R
XXI1	005774R	XXL6	005202R	XXR5	010150R	XXWA	005066R	ZBRAN	000144R
XXI2	006000R	XXL7	005212R	XXR6	010254R	XXW3	011610R	ZEQU	001556R
XXJ1	005550R	XXN1	003700R	XXR7	010224R	XXW4	011630R	ZERO	002510R
XXJ2	005554R	XXN2	004136R	XXTA	011054R	XXW5	011614R	ZLESS	001606R
XXJ4	005560R	XXN3	004140R						

. ABS.	000000	000
	043636	001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 2454 WORDS (10 PAGES)
DYNAMIC MEMORY: 18694 WORDS (71 PAGES)
ELAPSED TIME: 00:01:51
FORTH,FORTH=FORTH

Boot Block

```
                                ;  BOOT BLOCK.
                                ;
                                ;  THIS CODE IS IN TRACK 1 SECTOR 1 OF THE DISK, FOR STAND-ALONE BOOT.
                                ;
                                ;  THIS BOOT BLOCK READS A 2-SECTOR LOADER (256 BYTES) INTO THE LAST
                                ;  256 BYTES OF THE FIRST 8K OF MEMORY.  THIS LOADER IS ON DISK IN THE
                                ;  LAST TWO SECTORS OF SCREENS 40-47 OF THE 'FORTH.DAT' FILE ( THESE
                                ;  ARE SECTORS 26 AND 30 - OCTAL - OF TRACK 21).  WHEN THE LOADER
                                ;  EXECUTES IT READS IN THE REST OF THE BINARY IMAGE (STARTING AT
                                ;  ZERO) AND BRANCHES TO ADDRESS 1000 TO START EXECUTION OF THE
                                ;  LOADED PROGRAM.  THE LOADER, WHICH IS TABLE-DRIVEN, IS CREATED
                                ;  AND WRITTEN TO DISK BY FORTH SCREENS 36 AND 37.
                                ;
000000  000240                ORIGIN: NOP
000002  000415                        BR      SETUP
        000020'                 .=ORIGIN+20
000020  000064                        .WORD   64              ; IOT TRAP - USED FOR WAIT SUBROUTINE
000022  000340                        .WORD   340             ; SET PRIORITY 7
        000034'                 .=ORIGIN+34
000034  000000                ERROR:  HALT
000036  012700  017400        SETUP:  MOV     #17400,R0       ; 2-SECTOR LOADER WILL START AT 17400
000042  012706  001000                MOV     #1000,SP
000046  012704  177170                MOV     #177170,R4
000052  000004                        IOT                     ; WAIT FOR DISK TO FINISH
000054  000413                        BR      CONTIN          ; SKIP OVER INTERRUPT VECTOR
        000060'                 .=ORIGIN+60
000060  000034'                       .WORD   ERROR           ; HALT IF TERMINAL INTERRUPT
000062  000341                        .WORD   341             ; FLAG (WHICH HALT)
000064  005714                WAIT:   TST     (R4)            ; THIS SUBROUTINE IS CALLED BY IOT
000066  001776                        BEQ     WAIT            ; WAIT FOR SOMETHING
000070  100743                        BMI     ORIGIN          ; IF ERROR, START OVER (BRANCH TO ZERO)
000072  000002                RETURN: RTI
        000100'                 .=ORIGIN+100
000100  000072'                       .WORD   RETURN          ; IF CLOCK INTERRUPT, JUST RTI
000102  000340                        .WORD   340
000104  010405                CONTIN: MOV     R4,R5           ; PUT #177172
000106  005725                        TST     (R5)+           ;    IN R5
000110  012701  000026                MOV     #26,R1          ; READ SECTOR 26 FIRST
000114  012714  000007        RDLOOP: MOV     #7,(R4)         ; START THE READ OPERATION
000120  000004                        IOT                     ; WAIT (FOR TRANSFER FLAG)
000122  010115                        MOV     R1,(R5)         ; MOVE SECTOR # TO CONTROL REGISTER
000124  000004                        IOT
000126  012715  000021                MOV     #21,(R5)        ; MOVE TRACK #
000132  000004                        IOT
000134  012714  000003                MOV     #3,(R4)         ; EMPTY BUFFER COMMAND
000140  000004                        IOT
000142  111520                CHLOOP: MOVB    (R5),(R0)+      ; THIS LOOP MOVES 200 CHARACTERS
000144  000004                        IOT
000146  105714                        TSTB    (R4)            ; TEST FOR TRANSFER FLAG
000150  100774                        BMI     CHLOOP
                                ; WHEN GET HERE, DONE TRANSMITTING ONE SECTOR'S CHARACTERS
000152  022701  000026                CMP     #26,R1          ; GO AROUND AGAIN IF TRACK WAS 26
000156  001402                        BEQ     1$
000160  000167  017214                JMP     ORIGIN+17400    ; ELSE JUMP TO THE LOADER AT 17400
000164  012701  000030        1$:     MOV     #30,R1          ; SECOND (AND LAST) TIME, READ SECTOR 30
000170  000751                        BR      RDLOOP
        000001                        .END
```

```
.MAIN.  MACRO M1113  20-JAN-80 18:10  PAGE 1-1
SYMBOL TABLE

CHLOOP  000142R          ERROR   000034R          RDLOOP  000114R          SETUP   000036R          WAIT    000064R
CONTIN  000104R          ORIGIN  000000R          RETURN  000072R

. ABS.  000000      000
        000172      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  89 WORDS  ( 1 PAGES)
DYNAMIC MEMORY:  18694 WORDS  ( 71 PAGES)
ELAPSED TIME:  00:00:04
BOOT,BOOT=BOOT
```

Boot Block