

fig-FORTH FOR ALPHA MICRO

ASSEMBLY SOURCE LISTING

RELEASE 1

WITH COMPILER SECURITY

AND

VARIABLE LENGTH NAMES

SEPT 1980

The author holds the copyright to this publication.

MACRO ASSEMBLY LISTING

uA/FORTH V 3.2 INTRODUCTION SEPTEMBER 1980

THIS MODEL OF FORTH WAS ORIGINALLY DEVELOPED BY THE
FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
P.O. BOX 1105
SAN CARLOS, CA 94070

IMPLEMENTED FOR THE ALPHA-MICRO BY:

ROBERT BERKEY
PROFESSIONAL MANAGEMENT SERVICES
724 ARASTRADERO RD., SUITE 109
PALO ALTO, CA 94306

COPYRIGHT 1980 ROBERT BERKEY

WARRANTY: VENDOR WARRANTS uA/FORTH TO BE FREE FROM DEFECTS IN PRODUCTION FOR NINETY (90) DAYS FROM THE DATE OF DELIVERY. VENDOR'S LIABILITY IS LIMITED TO THE MODIFICATION OR REPLACEMENT OF DEFECTIVE ROUTINES, OR TO REFUND OF PURCHASE PRICE AT DISCRETION OF VENDOR. VENDOR SHALL SPECIFICALLY NOT BE LIABLE FOR USER'S DETERMINATION OF SUITABILITY OF uA/FORTH FOR ANY PARTICULAR TASK, OR FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES SUFFERED BY USER IN CONNECTION WITH THE USE OF uA/FORTH.

THE FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM HAVE ALSO DEVELOPED NEARLY IDENTICAL VERSIONS OF THIS SYSTEM FOR THE:

PDP-11	9900	8080	6800
6809	6502	PACE	ALPHA-MICRO

THIS VERSION FOR THE ALPHA-MICRO IS BASED ON THE SYSTEM IMPLEMENTED FOR THE PDP-11 BY JOHN S. JAMES.

FOR MORE INFORMATION, WRITE:

FORTH INTEREST GROUP
P.O. BOX 1105
SAN CARLOS, CA 94070

`ALPHA-MICRO' AND `AMOS' ARE TRADEMARKS OF ALPHA MICROSYSTEMS
`PDP' IS A TRADEMARK OF DIGITAL EQUIPMENT CORPORATION.

PAGE

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

; uA/FORTH HAS

- ; - FULL LENGTH NAMES (UP TO 31 CHARACTERS)
- ; - EXTENSIVE COMPILE-TIME CHECKS AND ERROR MESSAGES
- ; - 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 MULTI-USER/MULTI-TASKING (CURRENTLY SINGLE TASK)
- ; - AND CAN BE CONFIGURED TO RUN IN A 16K BYTE PARTITION
- ; (THIS INCLUDES BUFFERS AND ROOM FOR SUBSTANTIAL
- ; ADDITIONAL FORTH PROGRAMMING) ON AN ALPHA-MICRO RUNNING
- ; AMOS.

; 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, KIT
- ; 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
- ; PROFESSIONAL MANAGEMENT SERVICES OR
- ; FROM JOHN S. JAMES, BOX 348, BERKELEY, CA 94701
- ; - FORTH REFERENCE CARD FOR THE FORTH IMPLEMENTATION TEAM
- ; COMMON MODEL, AVAILABLE FROM FIG.
- ; - 'FIG-FORTH INSTALLATION MANUAL', ALSO FROM FIG.

; PAGE

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```
*****
```

```
BRINGING UP THE SYSTEM
```

```
*****
```

```
TO RUN:
```

```
- uA/FORTH IS DISTRIBUTED WITH THE SYSTEM IN ACCOUNT [170,10]. LOG
  INTO [170,10] AND TYPE `AFORTH'. uA/FORTH SHOULD COME UP AND TYPE
  `uA/FORTH V 3.2 '
```

```
- MAKE A COPY OF THE ACCOUNT. PROTECT AGAINST ACCIDENTALLY ERASING
  THE SYSTEM OR THE SOURCE PROGRAMS.
```

```
- AS PROVIDED, THE FILE `FORTH.SCR' CONTAINS FORTH SCREENS
  1-128. FORTH.SCR IS A CONTIGUOUS FILE.
```

```
- IF YOU WANT TO RE-ASSEMBLE THE SYSTEM (WHICH MOST USERS WOULD NEVER
  FIND NECESSARY), YOU MUST EDIT AND MACRO `AFORTH.MAC'.
```

```
- ADVANCED USERS MAY NOTE THAT THIS SYSTEM IS DESIGNED TO ALLOW THE
  MEMORY LAYOUT--NUMBER AND LOCATION OF DISK BUFFERS, LOCATION OF THE
  STACK, ETC.--TO BE CHANGED DYNAMICALLY, WITHOUT REASSEMBLY.
```

```
- TEST THAT IT IS UP BY TRYING SOME ARITHMETIC OR DEFINITIONS, E.G.,
  88 88 * . (NOTE THAT THE `.' MEANS PRINT)
```

```
: SQUARE DUP * ;
```

```
25 SQUARE .
```

```
OR TYPE `VLIST' FOR A LIST OF ALL THE FORTH OPERATIONS IN THE
  DICTIONARY.
```

```
- TEST THE DISK BY TYPING
```

```
1 LIST
```

```
WHICH SHOULD LIST THE SCREEN WHICH LOADS THE EDITOR,
  ASSEMBLER, AND STRING ROUTINES.
```

```
INCIDENTALLY, `FORTH.SCR' IS THE SYSTEM'S `VIRTUAL MEMORY' FILE, USED
  FOR DISK I/O. THE REST OF THE SYSTEM (THIS PROGRAM ALONE) CAN RUN
  INDEPENDENTLY, EVEN IF `FORTH.SCR' IS NOT AVAILABLE.
```

```
THE SYSTEM AS SUPPLIED RESERVES 10000. BYTES FOR YOUR FORTH PROGRAMMING AND
  STACK. THIS IS ENOUGH FOR SUBSTANTIAL PROJECTS. (NOTE THAT THE EDITOR,
  ASSEMBLER, AND STRING PACKAGE, IF LOADED, USE MORE THAN 6K OF THIS.) TO
  CHANGE THIS MEMORY SIZE, CHANGE THE `10000.' WHICH IS IN THE LINES
  FOLLOWING THE LABEL `XDP:', NEAR THE END OF THIS PROGRAM. INCIDENTALLY,
  VERY FEW JOBS (E.G., RECURSION) WILL EVER USE MORE THAN 100 WORDS OF THIS
  SPACE FOR THE STACK; THE REST OF THE SPACE IS AVAILABLE FOR A STRING STACK
  (IF USED) OR FOR YOUR PROGRAMS--AND FORTH OBJECT CODE IS CONSIDERABLY MORE
  COMPACT THAN ASSEMBLY.
```

```
THE FORTH CONTIGUOUS FILE `FORTH.SCR' IS USED FOR STORING SOURCE PROGRAMS
  (OR DATA). THIS FILE HAS 128 1-K SCREENS (1-128), I.E., 256 DISK BLOCKS.
  SCREENS 4 AND 5 ARE USED BY THE SYSTEM FOR STORING ERROR AND WARNING
  MESSAGES. SCREENS 6-32 CONTAIN A TEXT EDITOR, ASSEMBLER, STRING PACKAGE,
  AND MISCELLANEOUS EXAMPLES. THE TEXT EDITOR, STRING PACKAGE, AND THE
  MISCELLANEOUS EXAMPLES ARE IN THE PUBLIC DOMAIN, WHILE THE ASSEMBLER IS
  COPYRIGHT. USERS MAY WANT TO SAVE THEIR SOURCE PROGRAMS AND DATA IN THE
  BLANK SCREENS.
```

```
PAGE
```

```
FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070
```

```

;
; *****
;

```

```

;
; 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.
;

```

```

; `?TERMINAL' IS TRIGGERED BY CNTL-C.
;

```

```

; `EXPECT' OPERATES UNDER CONTROL OF AMOS. THIS PROVIDES FAMILIAR TYPING
; CONVENTIONS AND AVOIDS SWAPPING DELAYS. NOTE THAT `KEY' WILL ACCEPT ALL
; CONTROL CHARACTERS. APPLICATIONS REQUIRING THAT `EXPECT' ACCEPT CONTROL
; CHARACTERS SHOULD FIRST LOAD THE `EXPECT' PROVIDED ON SCREEN 33 OF
; FORTH.SCR.
;

```

```

; *****
;

```

```

; IMPLEMENTATION NOTES
;
; *****
;

```

```

; CODE ROUTINES CAN USE REGISTERS 0, 1, AND 2 WITHOUT RESTORING THEM.
;

```

```

; IMPLEMENTATION FOR THE ALPHA-MICRO RUNNING UNDER AMOS REQUIRES THAT
; CODE BE RELOCATABLE. uA/FORTH ACHIEVES THIS REQUIREMENT BY
; A BOOT-UP CODE OVERLAYED IN THE DICTIONARY SPACE. THIS
; CODE RELOCATES THE CFA, LFA, DOCOLON PARAMETERS, VOCABULARY STRUCTURE,
; AND START-UP TABLE ADDRESSES. AS ALL ADDRESSES ARE RELOCATED, THE
; RELOCATION BECOMES TRANSPARENT TO FIG-FORTH APPLICATION PROGRAMS
; FROM OTHER SOURCES AND COMPUTERS.
;

```

```

; uA/FORTH CAN BE LOADED INTO USER MEMORY. WHEN FIRST RUN ALL
; APPROPRIATE ADDRESSES ARE RELOCATED. THE USER CAN EXIT TO
; MONITOR AND RETURN TO uA/FORTH WITHOUT LOSING THE DICTIONARY ENTRIES.
; WHEN uA/FORTH IS RERUN, THE RELOCATION CODE IS BYPASSED AND A WARM
; START (`ABORT') FOLLOWS. SAVING OF RELOCATED CODE ONTO DISK SHOULD BE DONE
; ONLY WITH UNDERSTANDING AND IS NOT RECOMMENDED AS SUCH CODE CAN BE RERUN
; AGAIN ONLY AT THE MEMORY BASE AT WHICH IT WAS FIRST RELOCATED.
; (USUALLY ONLY UNDER THE SAME SYSTEM.INI AND WITHIN THE SAME JOB.)
; IF SUCH A RELOCATED CODE IS RELOADED AND RUN AT A DIFFERENT MEMORY BASE
; uA/FORTH WILL RESPOND `INVALID MEMORY BASE' AND RETURN TO MONITOR.
;
;
;
;

```

PAGE

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

MACRO ASSEMBLY LISTING

PAGE 005

```

; *****
;
;           SET UP REGISTERS AND MACROS.
;
; *****
;
;           COPY      DSK0:SYS.MAC[7,7]
;
; W=R2          ; TEMPORARY USED BY `NEXT' MACRO (THE INNER INTERPRETER)
; U=R3          ; POINTER TO THE USER AREA
; IP=R4         ; FORTH INSTRUCTION COUNTER
; S=R5          ; FORTH STACK POINTER
; RP=SP        ; FORTH RETURN-STACK POINTER
;
;
;           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, LESS THE LAST CHARACTER
; IF THE NAME IS AN ODD LENGTH.
; (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 AT 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 FOUR-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 BEGIN--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
; FORTH INTEREST GROUP  POB 1105  SAN CARLOS, CA 94070

```

```

; SYSTEM, USERS WILL BE ABLE TO DEFINE THEIR OWN PRIMITIVES DIRECTLY
; IN FORTH, IMMEDIATELY READY FOR EXECUTION.)
;
; MACRO PARAMETERS DO NOT PASS CHARACTERS `;' AND `<'. THE NEXT FOUR
; ARGUMENTS PASS THIS INFORMATION.
; (6) IF DEFINED, THE LAST CHARACTER IS `<'.
; (7) IF DEFINED, THE FIRST CHARACTER IS `<'.
; (8) IF DEFINED, THE FIRST CHARACTER IS `;'.
; (9) IF DEFINED, THE FIRST TWO CHARACTERS ARE `(;'.
;
; 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.
;
LINK=0 ; LAST LINK FIELD IS 0, INDICATING END OF THE DICTIONARY.
;
DEFINE HEAD LENGTH,NAME,LCHAR,LABEL,CODE,LT1,LT2,LT3,LT4
LINK2=.
EVEN
BYTE LENGTH
IF NB,LT4
ASCII /(/
ENDC
IF NB,LT3
ASCII /;/
ENDC
IF NB,LT2
ASCII /</
ENDC
IF NB,NAME
ASCII ^NAME^
ENDC
IF NB,LT1
ASCII /</
ENDC
BYTE LCHAR
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 ASSEMBLY LISTING

PAGE 007

;

DEFINE NEXT
MOV (IP)+,W
JMP @(W)+
ENDM

;

;

PAGE
FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070


```

; *****
;
;           START-UP TABLE
;
; *****
;
; AT STARTUP, MOST OF THESE VALUES ARE MOVED INTO THE USER AREA
; (LOCATED AFTER `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.
;
;
;
;
; ORIGIN: JMP      RELOC          ; GOTO RELOCATION CODE, REPLACE THIS JUMP
;                               ; WITH NULLS FOR WARM START FROM MONITOR.
; IN OTHER SYSTEMS THIS IS JUMP TO COLD (JMP CENT)
;
;           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.
;
;           WORD     1600          ; CPU
0000      73F7      191E          ; REVISION
000A      0011          ; POINTER TO LATEST WORD DEFINED
000C      1916          ; DEL CHARACTER
000E      007F          ; POINTER TO USER AREA
0010      4CFA          ; NOTE--THE USER AREA IS A HOOK IN THIS SYSTEM TO ALLOW MULTITASKING
; TO BE ADDED LATER.
;
;           WORD     XS0          ; POINTER TO BEGINNING OF THE STACK
0012      4032          ; POINTER TO BEGINNING OF RETURN STACK
0014      4CFA          ; POINTER TO TERMINAL INPUT BUFFER
0016      4C42          ; MAXIMUM NAME-FIELD WIDTH, NORMALLY 31
0018      001F          ; WARNING MODE; 0=ERROR #, 1=DISK MESSAGE
001A      0000          ; NOTE--WARNING MODE INITIALIZED TO ZERO, IN CASE DISK ISN'T UP.
;
;           WORD     XDP          ; FENCE TO PROTECT AGAINST ACCIDENTAL
;                               ; `FORGET' OF THE SYSTEM.
001C      1922          ;
;
;           WORD     XDP          ; POINTER TO NEXT AVAILABLE DICTIONARY
;                               ; LOCATION (RETURNED BY `HERE').
001E      1922          ;
;
;           WORD     XXVOC        ; POINTER TO INITIAL VOCABULARY LINK
0020      0FC0          ; INITIALIZE `FIRST'
0022      4036          ; INITIALIZE `LIMIT'
0024      4C42          ;
;
;           WORD     0            ; AVAILABLE
0026      0000          ;
;
;           WORD     0            ; AVAILABLE
0028      0000          ;
;

```

PAGE

NUCLEUS

002A	4389	504F	5259
0038	0046	000D	
003C	0046	4309	504F
0062	0046	0909	4C41
007C	0046	4854	2045
00BA	0046	4854	2045
00EE	0046	494C	4241
0116	B502	73DA	

```

; HEAD 211,COPYRIGH,324,COPYR ; ***** COPYRIGHT
; TYPECR
; TYPECR < COPYRIGHT 1980 ROBERT BERKEY>
; TYPECR < ALL RIGHTS RESERVED>
; TYPECR <THE SOFTWARE, PROGRAMS, AND DOCUMENTS CONTAINED HEREIN ARE>
; TYPECR <THE EXCLUSIVE PROPERTY OF THE COPYRIGHT HOLDER.>
; TYPECR <LIABILITY LIMITED TO PURCHASE PRICE.>
; NEXT

```

THE NUCLEUS CONTAINS THE PRIMITIVES FROM WHICH THE SYSTEM IS BUILT.

011A	4C83	D449	002A
0122	B525		
0124	B502	73DA	
0128	4587	4558	5543
0134	B542		
0136	73DA		

```

; HEAD 203,LI,324,LIT ; ***** LIT
; USED ONLY BY COMPILER. PUSH FOLLOWING LITERAL ONTO STACK.
; MOV (IP)+,-(S)
; NEXT
; HEAD 207,EXECUT,305,EXEC ; ***** EXECUTE
; EXECUTE FORTH WORD WHOSE CODE ADDRESS IS ON STACK
; MOV (S)+,W
; JMP @(W)+

```

0138	4286	4152	434E
0144	1304		
0146	B502	73DA	

```

; HEAD 206,BRANCH,240,BRAN ; ***** BRANCH
; USED ONLY BY COMPILER. FORTH BRANCH TO ADDRESS WHICH FOLLOWS.
; ADD (IP),IP
; NEXT

```

014A	3087	5242	4E41
0156	0A95		
0158	0203		
015A	1304		
015C	B502	73DA	
0160	0901		
0162	B502	73DA	

```

; HEAD 207,OBRANC,310,ZBRAN ; ***** OBRANCH
; USED ONLY BY COMPILER. FORTH BRANCH IF TOP OF STACK
; IS ZERO (FALSE).
; TST (S)+
; BNE A3$
; ADD (IP),IP
; NEXT
A3$: ADDI 2,IP
; NEXT

```

0166	2886	4F4C	504F
0172	0C8E		
0174	93B6	0002	
0178	8003		

```

; HEAD 206,(LOOP),240,XLOOP ; ***** (LOOP)
; USED ONLY BY COMPILER. INCREMENT LOOP INDEX BY 1, BRANCH
; IF INDEX BELOW LIMIT.
; INC (RP)
; CMP (RP),2(RP)
; BPL B2$

```

```

017A 1304
017C B502 73DA
0180 0983
0182 0901
0184 B502 73DA

;
;
0188 2887 4C2B 4F4F
; HEAD 207,(+LOOP,251,XPLOO ; ***** (+LOOP)
; USED ONLY BY COMPILER. INCREMENT LOOP INDEX BY TOP OF STACK,
; MAYBE BRANCH.
0194 134E
0196 0A95
0198 0506
019A 93B6 0002
019E 8009
01A0 1304
01A2 B502 73DA
01A6 9D8E 0002
01AA 8003
01AC 1304
01AE B502 73DA
01B2 0983
01B4 0901
01B6 B502 73DA

;
;
01BA 2884 4F44 A029
; HEAD 204,(DO),240,XDO ; ***** (DO)
; USED ONLY BY COMPILER. SET UP `DO' LIMIT AND INDEX.
01C4 BD66 0002
01C8 B366
01CA 0943
01CC B502 73DA
;
;
01D0 C981 01BA 01D6
; HEAD 201,,311,I ; ***** I
; RETURN CURRENT LOOP INDEX TO STACK.
01D6 B3A5
01D8 B502 73DA
;
;
01DC 4485 4749 D449
; HEAD 205,DIGI,324,DIGIT ; ***** DIGIT
; USED BY COMPILER.
; ( ASCII-DIGIT BASE ==> DIGIT-VALUE TRUE (OR FALSE))
01E6 25F5 0030 0002
01EC 9D57 0002 0009
01F2 0707
01F4 25F5 0007 0002
01FA 9D57 0002 000A
0200 050A
0202 0AB5 0002
0206 0507
0208 9D4D 0002
020C 0404
020E B5CD 0001
0212 B502 73DA
0216 0941

;
;
D1$: TST 2(S) ; IF LESS THAN ZERO, ERROR
BLT D2$
CMP 2(S),(S) ; OR IF NOT LESS THAN BASE, ERR
MOV #1,(S) ; VALID RETURN
NEXT
D2$: ADDI 2,S

```

```

0218 0B4D CLR (S) ; ERROR--RETURN '0' FLAG
021A B502 73DA NEXT
;
;
021E 2886 4946 444E HEAD 206,(FIND),240,PFIND ; ***** (FIND)
; USED BY COMPILER. FIND A WORD IN THE DICTIONARY.
; ( STRING-ADDRESS NFA ==> PFA LENGTH TRUE (OR FALSE)).
; STRING-ADDRESS IS ADDRESS OF THE LENGTH BYTE OF THE
; STRING BEING SOUGHT. NFA IS NAME-FIELD ADDRESS OF
; WORD IN DICTIONARY WHERE SEARCH BEGINS. PFA IS
; PARAMETER-FIELD ADDRESS OF THE DICTIONARY ENTRY
; WHICH IS FOUND. IF WORD NOT FOUND, ONLY ONE RESULT
; (0, FALSE) IS RETURNED.
; SETUP--GET ARGS, PRESERVE NEEDED REGISTERS
022A B540 MOV (S)+,R0 ; DICTIONARY ADDRESS
022C B541 MOV (S)+,R1 ; STRING ADDRESS
022E B166 MOV R5,-(RP) ; PRESERVE REGISTERS
0230 B126 MOV R4,-(RP)
0232 B0E6 MOV R3,-(RP)
0234 0B66 CLR -(RP) ; SPACE TO STORE LENGTH BYTE
; PREPARE R2 FOR FAST COMPARE
0236 B242 MOV (R1),R2
0238 45C2 8080 BIC #100200,R2
;
;FCOMP:
; FAST TEST TO ELIMINATE MOST WORDS
; COMPARE FIRST WORD TO SPECIALLY PREPARED R2
; THEN INCREMENT TO FIND END OF NAME.
023C B203 FAST: MOV (R0),R3
023E 45C3 80C0 BIC #100300,R3
0242 9083 CMP R2,R3
0244 0306 BEQ NOFAST ; NO FAST ELIMINATION POSSIBLE
0246 0A90 XMATCH: TST (R0)+ ; BRANCH HERE IF NO MATCH THIS TIME
0248 80FE BPL XMATCH
; R0 NOW POINTS TO LINK
024A 0A88 TST (R0)
024C 031C BEQ FAILED
024E B200 MOV (R0),R0
0250 01F5 BR FCOMP
; END OF FAST ELIMINATION TEST
;
0252 B20E NOFAST: MOV (R0),(RP) ; SAVE LENGTH BYTE
0254 B045 MOV R1,R5 ; SET R5
0256 0107 BR NOFAST1
; NOW DO THE MAIN LOOP TO CHECK FOR MATCH
0258 0A95 MLOOP: TST (R5)+
025A B344 MOV (R5),R4
025C B203 MOV (R0),R3
025E 45C3 8000 BIC #100000,R3
0262 90C4 CMP R3,R4
0264 02F0 BNE XMATCH
0266 A5D0 8000 NOFAST1: BIT #100000,(R0)+
026A 03F6 BEQ MLOOP
; IF GET HERE, FOUND IT.
uA/FORTH [170,10] FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

```

026C B582          MOV      (RP)+,R2      ; POP AND SAVE LENGTH BYTE
026E B583          MOV      (RP)+,R3      ; RESTORE REGISTERS
0270 B584          MOV      (RP)+,R4
0272 B585          MOV      (RP)+,R5
0274 0803         ADDI     4,R0          ; GET PARAMETER FIELD ADDRESS
0276 B025          MOV      R0,-(S)
0278 45C2 FF00     BIC      #177400,R2      ; R2 CONTAINS LENGTH BYTE
027C B0A5          MOV      R2,-(S)
027E B5E5 0001     MOV      #1,-(S)
0282 B502 73DA     NEXT
0286 0A96         FAILED: TST     (RP)+      ; POP LENGTH BYTE
0288 B583          MOV      (RP)+,R3      ; RESTORE REGISTERS
028A B584          MOV      (RP)+,R4
028C B585          MOV      (RP)+,R5
028E 0B65         CLR      -(S)          ; REPLACE LENGTH BYTE WITH
                                ; FAILURE FLAG.
0290 B502 73DA     NEXT          ; WE ARE DONE--FAILURE TO FIND
;
0294 4587 434E 4F4C HEAD 207,ENCLOS,305,ENCL ; ***** ENCLOSE
; USED BY COMPILER. BREAK NEXT WORD OUT OF INPUT BUFFER.
; ( START-ADDRESS DELIMITER ==> ADDRESS OFFSET END NEXT-CHARACTER)
02A0 B340          MOV      (S),R0          ; DELIMITER
02A2 BD41 0002     MOV      2(S),R1        ; STARTING ADDRESS
02A6 0953         SUBI     4,S            ; MAKE SPACE FOR RESULTS
02A8 C440         ENC1:  CMPB    (R1)+,R0
02AA 03FE         BEQ      ENC1          ; SKIP OVER LEADING DELIMITERS
02AC 0850         SUBI     1,R1
02AE B075 0004     MOV      R1,4(S)
02B2 8A89         ENC2:  TSTB   (R1)          ; TEST FOR NULL
02B4 030F         BEQ      ENC4
02B6 C440         CMPB    (R1)+,R0      ; NOT NULL, SO FIND END OF TOKEN
02B8 02FC         BNE     ENC2
02BA B04D         MOV      R1,(S)
02BC 0850         SUBI     1,R1
02BE B075 0002     ENC3:  MOV      R1,2(S)      ; FINISH UP AND RETURN
02C2 BD41 0006     MOV      6(S),R1
02C6 204D         SUB      R1,(S)
02C8 2075 0002     SUB      R1,2(S)
02CC 2075 0004     SUB      R1,4(S)
02D0 B502 73DA     NEXT
02D4 B04D         ENC4:  MOV      R1,(S)      ; HANDLE NULL CASE
02D6 9075 0004     CMP      R1,4(S)
02DA 02F1         BNE     ENC3
02DC 0840         ADDI    1,R1
02DE 01EF         BR      ENC3
;
;
; THE NEXT 4 HEADERS POINT TO INSTALLATION-DEPENDENT TERMINAL I/O
; ROUTINES.
;
02E0 4584 494D A054 HEAD 204,EMIT,240,EMIT,PEMIT ; ***** EMIT

```

```

02EA 4B83 D945 02E0 HEAD 203,KE,331,KEY,PKEY ; ***** KEY
;
02F2 3F89 4554 4D52 HEAD 211,?TERMINA,314,QTERM,PQTER ; ***** ?TERMINAL
;
0300 4382 A052 02F2 HEAD 202,CR,240,CR,PCR ; ***** CR
;
;
;
;
0308 4385 4F4D C556 HEAD 205,CMOV,305,CMOVE ; ***** CMOVE
; MOVE BYTES IN MEMORY. ( FROM TO N ==>)
0312 B540 MOV (S)+,R0
0314 0305 BEQ G1$, ; NO MOVE
0316 B541 MOV (S)+,R1
0318 B542 MOV (S)+,R2
031A 0E91 MBBU R2,R1
031C B502 73DA NEXT
0320 0943 G1$: ADDI 4,S ; REMOVE ADDRESSES
0322 B502 73DA NEXT
;
;
0326 5582 A02A 0308 HEAD 202,U*,240,USTAR ; ***** U*
; ( N1 N2 ==> PRODUCT). PRODUCT IS 32-BIT DOUBLE INTEGER,
; HIGH WORD TOP.
; THIS MUST BE UNSIGNED MULTIPLICATION.
032E 71F7 0004 JSR PC,UMULT
0332 B502 73DA NEXT
0336
UMULT:
; THE VALUES TO MULTIPLY ARE ON THE STACK.
0336 B540 MOV (S)+,R0
0338 7C0D MUL R0,(S)
033A B00D MOV R0,(S)
033C B065 MOV R1,-(S)
033E 001F RTN PC
;
0340 5582 A02F 0326 HEAD 202,U/,240,USLAS ; ***** U/
; THIS DIVISION MUST BE UNSIGNED.
0348 71F7 0004 JSR PC,UDIV
034C B502 73DA NEXT
0350
UDIV:
; THE VALUES TO DIVIDE ARE ON THE STACK
0350 BD40 0004 MOV 4(S),R0
0354 BD41 0002 MOV 2(S),R1
0358 7E15 DIV R0,(S)+
035A B075 0002 MOV R1,2(S)
035E B00D MOV R0,(S)
0360 001F RTN PC
;
0362 4183 C44E 0340 HEAD 203,AN,304,AND ; ***** AND
; BITWISE AND. ( N1 N2 ==> N3).
036A 354D AND (S)+,(S)
036C B502 73DA NEXT

```

uA/FORTH	[170,10]				MACRO ASSEMBLY LISTING	
0370	4F82	A052	0362		HEAD 202,OR,240,OR	; ***** OR
0378	554D				BIS (S)+,(S)	
037A	B502	73DA			NEXT	
037E	5883	D24F	0370		HEAD 203,XO,322,XOR	; ***** XOR
0386	654D				XOR (S)+,(S)	
0388	B502	73DA			NEXT	
038C	5383	C050	037E		HEAD 203,SP,300,SPAT	; ***** SP@
0394	B165				MOV S,-(S)	
0396	B502	73DA			NEXT	
039A	5383	A150	038C		HEAD 203,SP,241,SPSTO	; ***** SP!
03A2	BCC5	0006			MOV 6(U),S	; OFFSET 6 IN USER AREA
03A6	B502	73DA			NEXT	
03AA	5283	A150	039A		HEAD 203,RP,241,RPSTO	; ***** RP!
03B2	BDC6	FC5E			MOV ORIGIN+24,RP	
03B6	B502	73DA			NEXT	
03BA	3B82	A053	03AA		HEAD 202,S,240,SEMIS,,,,LT3	; ***** ;S
03C2	B584				MOV (RP)+,IP	
03C4	B502	73DA			NEXT	
03C8	4C85	4145	C556		HEAD 205,LEAV,305,LEAVE	; ***** LEAVE
03D2	B3B6	0002			MOV (RP),2(RP)	
03D6	B502	73DA			NEXT	
03DA	3E82	A052	03C8		HEAD 202,>R,240,TOR	; ***** >R
03E2	B566				MOV (S)+,-(RP)	
03E4	B502	73DA			NEXT	
03E8	5282	A03E	03DA		HEAD 202,R>,240,FROMR	; ***** R>
03F0	B5A5				MOV (RP)+,-(S)	
03F2	B502	73DA			NEXT	
03F6	D281	03E8	03FC		HEAD 201,,322,R	; ***** R
03FC	B3A5				MOV (RP),-(S)	
03FE	B502	73DA			NEXT	
0402	3082	A03D	03F6		HEAD 202,0=,240,ZEQU	; ***** 0=
040A	0A8D				TST (S)	
040C	0303				BEQ I1\$	
040E	0B4D				CLR (S)	
0410	B502	73DA			NEXT	
0414	B5CD	0001		I1\$:	MOV #1,(S)	
0418	B502	73DA			NEXT	
041C	3082	A03C	0402		HEAD 202,0,240,ZLESS,,LT1	; ***** 0<
0424	0A8D				TST (S)	
0426	8103				BMI J1\$	
0428	0B4D				CLR (S)	
042A	B502	73DA			NEXT	
042E	B5CD	0001		J1\$:	MOV #1,(S)	

uA/FORTH

[170,10]

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

0432 B502 73DA ; NEXT
0436 AB81 041C 043C ; HEAD 201,,253,PLUS ; ***** +
043C 154D ; ADD (S)+,(S)
043E B502 73DA ; NEXT
0442 4482 A02B 0436 ; HEAD 202,D+,240,DPLUS ; ***** D+
044A 1D75 0002 0006 ; ADD 2(S),6(S) ; ADD LOW
0450 8DB5 0004 ; ADC 4(S)
0454 1375 0004 ; ADD (S),4(S) ; ADD HIGH
0458 0943 ; ADDI 4,S
045A B502 73DA ; NEXT
045E 4D85 4E49 D355 ; HEAD 205,MINU,323,MINUS ; ***** MINUS
0468 0C4D ; CHANGE SIGN.
046A B502 73DA ; NEG (S)
; NEXT
046E 4486 494D 554E ; HEAD 206,DMINUS,240,DMINU ; ***** DMINUS
; CHANGE SIGN OF DOUBLE INTEGER WORD ON STACK.
047A 0C4D ; NEG (S)
047C 0C75 0002 ; NEG 2(S)
0480 8DCD ; SBC (S)
0482 B502 73DA ; NEXT
0486 4F84 4556 A052 ; HEAD 204,OVER,240,OVER ; ***** OVER
; ( N1 N2 ==> N1 N2 N1)
0490 BD65 0002 ; MOV 2(S),-(S)
0494 B502 73DA ; NEXT
0498 4484 4F52 A050 ; HEAD 204,DROP,240,DROP ; ***** DROP
04A2 0941 ; ADDI 2,S
04A4 B502 73DA ; NEXT
04A8 5384 4157 A050 ; HEAD 204,SWAP,240,SWAP ; ***** SWAP
04B2 BD41 0002 ; MOV 2(S),R1
04B6 B54D ; MOV (S)+,(S)
04B8 B065 ; MOV R1,-(S)
04BA B502 73DA ; NEXT
04BE 4483 D055 04A8 ; HEAD 203,DU,320,DUP ; ***** DUP
04C6 B365 ; MOV (S),-(S)
04C8 B502 73DA ; NEXT
04CC 2B82 A021 04BE ; HEAD 202,+1,240,PSTOR ; ***** +!
; ADD NUMBER SECOND ON STACK TO ADDRESS ON TOP.
04D4 1D5D 0002 ; ADD 2(S),@(S)+
04D8 0941 ; ADDI 2,S
04DA B502 73DA ; NEXT
04DE 5486 474F 4C47 ; HEAD 206,TOGGLE,240,TOGGL ; ***** TOGGLE
; ( BYTE-ADDRESS BIT-PATTERN ==> ) EXCLUSIVE-OR INTO MEMORY BYTE.
04EA DF41 0002 ; MOVB @2(S),R1
04EE 6541 ; XOR (S)+,R1

```


uA/FORTH

[170,10]

MACRO ASSEMBLY LISTING

04F0
04F2

D05D
B502 73DA

MOVB
NEXT

R1,@(S)+

04F6
04FC
04FE

C081 04DE
B765
B502 73DA

04FC

;

HEAD
MOV
NEXT

201,,300,AT
@(S)+,-(S)

; ***** @

0502
050A
050E
0512
0514

4382 A040
DF41 0000
45C1 FF00
B04D
B502 73DA

04F6

;

HEAD
MOVB
BIC
MOV
NEXT

202,C@,240,CAT
@(S),R1
#177400,R1
R1,(S)

; ***** C@

0518
051E
0522
0524

A181 0502
BD5D 0002
0941
B502 73DA

051E

;

HEAD
MOV
ADDI
NEXT

201,,241,STORE
2(S),@(S)+
2,S

; ***** !

0528
0530
0536
0538

4382 A021
DD7D 0002
0943
B502 73DA

0518
0000

;

HEAD
MOVB
ADDI
NEXT

202,C!,240,CSTOR
2(S),@(S)
4,S

; ***** C!

;

;

PAGE

uA/FORTH

[170,10]

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

; *****

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.

053C BAC1 0528 0554
0542 0900 08BA 06DC
0554 B126
0556 B084
0558 B502 73DA

HEAD 301,,272, COLON, DOCOL ; ***** :
WORD QEXEC, SCSP, CURR, AT, CONT, STORE, CREAT, RBRAC, PSCOD
DOCOL: MOV IP, - (RP)
MOV W, IP
NEXT

055C BBC1 053C 0554
0562 092C 0964 03C0

HEAD 301,,273, SEMI, DOCOL ; ***** ;
WORD QCSP, COMP, SEMIS, SMUDG, LBRAC, SEMIS

056E 4388 4E4F 5453
057C 0E24 099E 078E
0584 B2A5
0586 B502 73DA

HEAD 210, CONSTANT, 240, CON, DOCOL ; ***** CONSTANT
WORD CREAT, SMUDG, COMMA, PSCOD
DOCON: MOV (W), - (S)
NEXT

058A 5688 5241 4149
0598 057A 09F0
059C B0A5
059E B502 73DA

HEAD 210, VARIABLE, 240, VAR, DOCOL ; ***** VARIABLE
WORD CON, PSCOD
DOVAR: MOV W, - (S)
NEXT

05A2 5584 4553 A052
05AC 057A 09F0
05B0 B2A5
05B2 10CD
05B4 B502 73DA

HEAD 204, USER, 240, USER, DOCOL ; ***** USER
; CREATE A NEW USER VARIABLE. (N ==>).
WORD CON, PSCOD
DOUSE: MOV (W), - (S)
ADD U, (S)
NEXT

CONSTANTS

05B8 B081 05A2 0584
05BE 0000

HEAD 201,,260, ZERO, DOCON ; ***** 0
WORD 0

05C0 B181 05B8 0584
05C6 0001

HEAD 201,,261, ONE, DOCON ; ***** 1
WORD 1

05C8 B281 05C0 0584
05CE 0002

HEAD 201,,262, TWO, DOCON ; ***** 2
WORD 2

05D0 B381 05C8 0584
05D6 0003

HEAD 201,,263, THREE, DOCON ; ***** 3
WORD 3


```

; VOCABULARY LINK (MAINLY FOR FUTURE USE).
0674 0014          WORD 24
;
0676 4685 5249 D453 HEAD 205,FIRS,324,FIRST,DOUSE ; ***** FIRST
; ADDRESS OF BEGINNING OF DISK BUFFER.
0680 0016          WORD 26
;
0682 4C85 4D49 D449 HEAD 205,LIMI,324,LIMIT,DOUSE ; ***** LIMIT
; ADDRESS JUST BEYOND END OF DISK BUFFERS.
068C 0018          WORD 30
;
; POSITIONS 32 AND 34 ARE AVAILABLE FOR EXPANSION.
; THEY ARE INITIALIZED FROM BOOT-UP TABLE, AT COLD START.
;
068E 4283 CB4C 0682 HEAD 203,BL,313,BLK,DOUSE ; ***** BLK
; CURRENT DISK BLOCK BEING LOADED (0=TERMINAL)
0696 001E          WORD 36
;
0698 4982 A04E 068E HEAD 202,IN,240,IN,DOUSE ; ***** IN
; OFFSET IN TERMINAL INPUT BUFFER.
06A0 0020          WORD 40
;
06A2 4F83 D455 0698 HEAD 203,OU,324,OUT,DOUSE ; ***** OUT
; OFFSET IN OUTPUT LINE.
06AA 0022          WORD 42
;
06AC 5383 D243 06A2 HEAD 203,SC,322,SCR,DOUSE ; ***** SCR
; CURRENT FORTH DISK SCREEN.
06B4 0024          WORD 44
;
06B6 4F86 4646 4553 HEAD 206,OFFSET,240,OFSET,DOUSE ; ***** OFFSET
; OFFSET TO GET TO ANOTHER DISK DRIVE.
06C2 0026          WORD 46
;
06C4 4387 4E4F 4554 HEAD 207,CONTEX,324,CONT,DOUSE ; ***** CONTEXT
06D0 0028          WORD 50
;
06D2 4387 5255 4552 HEAD 207,CURREN,324,CURR,DOUSE ; ***** CURRENT
06DE 002A          WORD 52
;
06E0 5385 4154 C554 HEAD 205,STAT,305,STATE,DOUSE ; ***** STATE
06EA 002C          WORD 54
;
06EC 4284 5341 A045 HEAD 204,BASE,240,BASE,DOUSE ; ***** BASE
06F6 002E          WORD 56
;
06F8 4483 CC50 06EC HEAD 203,DP,314,DPL,DOUSE ; ***** DPL
; OFFSET OF DECIMAL POINT AFTER DOUBLE-INTEGGER INPUT.
0700 0030          WORD 60
;
0702 4683 C44C 06F8 HEAD 203,FL,304,FLD,DOUSE ; ***** FLD
; OUTPUT FIELD WIDTH.
070A 0032          WORD 62

```

070C	4383	D053	0702	HEAD 203,CS,320,CSP,DOUSE	; ***** CSP
				; USED BY COMPILER TO HOLD CURRENT STACK POSITION,	
				; FOR ERROR CHECKING.	
0714	0034			WORD 64	
				;	
0716	5282	A023	070C	HEAD 202,R#,240,RNUM,DOUSE	; ***** R#
				; CURSOR POSITION (FOR SOME EDITORS).	
071E	0036			WORD 66	
				;	
0720	4883	C44C	0716	HEAD 203,HL,304,HLD,DOUSE	; ***** HLD
				; POINTS TO LAST CHARACTER HELD IN 'PAD'	
0728	0038			WORD 70	
				;	
072A	5583	C553	0720	HEAD 203,US,305,USE,DOUSE	; ***** USE
0732	003A			WORD 72	
				;	
0734	5084	4552	A056	HEAD 204,PREV,240,PREV,DOUSE	; ***** PREV
073E	003C			WORD 74	
				;	
0740	4F86	4952	4947	HEAD 206,ORIGIN,240,ORIGI,DOUSE	; ***** ORIGIN
074C	003E			WORD 76	
				;	
				; END OF USER AREA	
				;	
				;	
074E	3182	A02B	0740	HEAD 202,1+,240,ONEP	; ***** 1+
0756	0C8D			INC (S)	
0758	B502	73DA		NEXT	
				;	
075C	3282	A02B	074E	HEAD 202,2+,240,TWOP	; ***** 2+
0764	0D0D			(S)	
0766	B502	73DA		NEXT	
				;	
076A	4884	5245	A045	HEAD 204,HERE,240,HERE,DOCOL	; ***** HERE
0774	0662	04FA	03C0	WORD DP,AT,SEMIS	
				;	
077A	4185	4C4C	D44F	HEAD 205,ALLO,324,ALLOT,DOCOL	; ***** ALLOT
0784	0662	04D2	03C0	WORD DP,PSTOR,SEMIS	
				;	
078A	AC81	077A	0554	HEAD 201,,254,COMMA,DOCOL	; ***** ,
0790	0772	051C	05CC	WORD HERE,STORE,TWO,ALLOT,SEMIS	
				;	
				; THIS SYSTEM DOES NOT USE 'C,'	
				;	
079A	AD81	078A	07A0	HEAD 201,,255,SUB	; ***** -
07A0	254D			SUB (S)+,(S)	
07A2	B502	73DA		NEXT	
				;	
07A6	BD81	079A	07AC	HEAD 201,,275,EQUAL	; ***** =
07AC	954D			CMP (S)+,(S)	
07AE	0302			BEQ K1\$	
07B0	0B4D			CLR (S)	
07B2	0102			BR K2\$	

```

07B4 B5CD 0001 K1$: MOV #1,(S)
07B8 B502 73DA K2$: NEXT
;
07BC BC81 07A6 07C2 HEAD 201,,274,LESS ; ***** <
07C2 9D55 0002 CMP 2(S),(S)+
07C6 0502 BLT L1$
07C8 0B4D CLR (S)
07CA 0102 BR L2$
07CC B5CD 0001 L1$: MOV #1,(S)
07D0 B502 73DA L2$: NEXT
;
07D4 BE81 07BC 07DA HEAD 201,,276,GREAT ; ***** >
07DA 9D55 0002 CMP 2(S),(S)+
07DE 0602 BGT M1$
07E0 0B4D CLR (S)
07E2 0102 BR M2$
07E4 B5CD 0001 M1$: MOV #1,(S)
07E8 B502 73DA M2$: NEXT
;
07EC 5283 D44F 07D4 HEAD 203,RO,324,ROT ; ***** ROT
07F4 B340 MOV (S),RO
07F6 BD4D 0004 MOV 4(S),(S)
07FA BD75 0002 0004 MOV 2(S),4(S)
0800 B035 0002 MOV R0,2(S)
0804 B502 73DA NEXT
;
0808 5385 4150 C543 HEAD 205,SPAC,305,SPACE,DOCOL ; ***** SPACE
0812 0120 0020 02E8 WORD LIT,40,EMIT,SEMIS
;
081A 2D84 5544 A050 HEAD 204,-DUP,240,DDUP ; ***** -DUP
0824 0A8D TST (S)
0826 0301 BEQ N1$
0828 B365 MOV (S),-(S)
082A B502 73DA N1$: NEXT
;
082E 5488 4152 4556 HEAD 210,TRAVERSE,240,TRAV,DOCOL ; ***** TRAVERSE
; MOVE (FORWARDS OR BACKWARDS) ACROSS A (VARIABLE LENGTH)
; DICTIONARY NAME FIELD.
083C 04B0 WORD SWAP
083E 048E 043A 0120 XXN1: WORD OVER,PLUS,LIT,177,OVER,CAT,LESS,ZBRAN,XXN1-.
0850 04B0 04A0 03C0 WORD SWAP,DROP,SEMIS
;
0856 4C86 5441 5345 HEAD 206,LATEST,240,LATES,DOCOL ; ***** LATEST
0862 06DC 04FA 04FA WORD CURR,AT,AT,SEMIS
;
; THE NEXT 4 OPERATORS CAN DEPEND ON COMPUTER WORD SIZE.
; THEY CONVERT ADDRESSES WITHIN THE NAME FIELDS OF FORTH
; DICTIONARY ENTRIES.
;
086A 4C83 C146 0856 HEAD 203,LF,301,LFA,DOCOL ; ***** LFA
0872 0120 0004 079E WORD LIT,4,SUB,SEMIS
;
087A 4383 C146 086A HEAD 203,CF,301,CFA,DOCOL ; ***** CFA
0882 05CC 079E 03C0 WORD TWO,SUB,SEMIS

```

```

;
0888 4E83 C146 087A HEAD 203,NF,301,NFA,DOCOL ; ***** NFA
0890 0120 0005 079E WORD LIT,5,SUB,LIT,-1,TRAV,SEMIS
;
089E 5083 C146 0888 HEAD 203,PF,301,PFA,DOCOL ; ***** PFA
08A6 05C4 083A 0120 WORD ONE,TRAV,LIT,5,PLUS,SEMIS
;
; THE NEXT 7 OPERATIONS ARE USED BY THE COMPILER, FOR
; COMPILE-TIME SYNTAX-ERROR CHECKS.
;
08B2 2184 5343 A050 HEAD 204,!CSP,240,SCSP,DOCOL ; ***** !CSP
08BC 0392 0712 051C WORD SPAT,CSP,STORE,SEMIS
;
08C4 3F86 5245 4F52 HEAD 206,?ERROR,240,QERR,DOCOL ; ***** ?ERROR
08D0 04B0 0154 0008 WORD SWAP,ZBRAN,XXN2-.,ERROR,BRAN,XXN3-.
08DC 04A0 XXN2: WORD DROP
08DE 03C0 XXN3: WORD SEMIS
;
08E0 3F85 4F43 D04D HEAD 205,?COM,320,QCOMP,DOCOL ; ***** ?COMP
08EA 06E8 04FA 0408 WORD STATE,AT,ZEQU,LIT,21,QERR,SEMIS
;
08F8 3F85 5845 C345 HEAD 205,?EXE,303,QEXEC,DOCOL ; ***** ?EXEC
0902 06E8 04FA 0120 WORD STATE,AT,LIT,22,QERR,SEMIS
;
090E 3F86 4150 5249 HEAD 206,?PAIRS,240,QPAIR,DOCOL ; ***** ?PAIRS
091A 079E 0120 0013 WORD SUB,LIT,23,QERR,SEMIS
;
0924 3F84 5343 A050 HEAD 204,?CSP,240,QCSP,DOCOL ; ***** ?CSP
092E 0392 0712 04FA WORD SPAT,CSP,AT,SUB,LIT,24,QERR,SEMIS
;
093E 3F88 4F4C 4441 HEAD 210,?LOADING,240,QLOAD,DOCOL ; ***** ?LOADING
094C 0694 04FA 0408 WORD BLK,AT,ZEQU,LIT,26,QERR,SEMIS
;
095A 4387 4D4F 4950 HEAD 207,COMPIL,305,COMP,DOCOL ; ***** COMPIL
; COMPIL THE EXECUTION ADDRESS FOLLOWING.
0966 08E8 03EE 04C4 WORD QCOMP,FROMR,DUP,TWOP,TOR,AT,COMMA,SEMIS
;
0976 DBC1 095A 0554 HEAD 301,,333,LBRAC,DOCOL ; ***** [
; STOP COMPILATION, ENTER EXECUTION STATE.
097C 05BC 06E8 051C WORD ZERO,STATE,STORE,SEMIS
;
0984 DD81 0976 0554 HEAD 201,,335,RBRAC,DOCOL ; ***** ]
; ENTER COMPILATION STATE.
098A 0120 00C0 06E8 WORD LIT,300,STATE,STORE,SEMIS
;
0994 5386 554D 4744 HEAD 206,SMUDGE,240,SMUDG,DOCOL ; ***** SMUDGE
; ALTER LATEST WORD NAME (SO THAT DICTIONARY SEARCH
; WON'T FIND A PARTIALLY-COMPLETE ENTRY.
09A0 0860 0120 0020 WORD LATES,LIT,40,TOGGL,SEMIS
;
09AA 4883 D845 0994 HEAD 203,HE,330,HEX,DOCOL ; ***** HEX
09B2 0120 0010 06F4 WORD LIT,20,BASE,STORE,SEMIS
;
09BC 4487 4345 4D49 HEAD 207,DECIMA,314,DEC,DOCOL ; ***** DECIMAL
uA/FORTH [170,10] FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

```

uA/FORTH      [170,10]
09C8          0120    000A    06F4          WORD      LIT,12,BASE,STORE,SEMIS
;
09D2          4F85    5443    CC41          HEAD      205,OCTA,314,OCTAL,DOCOL      ; ***** OCTAL
09DC          0120    0008    06F4          WORD      LIT,10,BASE,STORE,SEMIS
;
09E6          2887    433B    444F          HEAD      207,CODE,251,PSCOD,DOCOL,,,,LT4      ; ***** (;CODE)
; USED ONLY BY COMPILER; COMPILED BY `;CODE'.
09F2          03EE    0860    08A4          WORD      FROMR,LATES,PFA,CFA,STORE,SEMIS
;
;
09FE          3BC5    4F43    C544          HEAD      305,COD,305,SEMIC,DOCOL,,,LT3      ; ***** ;CODE
; CREATE NEW DATA TYPE WITH CODE ROUTINE WRITTEN IN ASSEMBLY.
0A08          092C    0964    09F0          WORD      QCSP,COMP,PSCOD,LBRAC,SMUDG,SEMIS
; NOTE: LATER, THE ASSEMBLER WILL PATCH THIS DEFINITION.
;
;
0A14          3C87    5542    4C49          HEAD      207,BUILD,323,BUILD,DOCOL,,LT2      ; ***** <BUILDS
; CREATE NEW DATA TYPE WITH CODE ROUTINE IN HIGHER-LEVEL FORTH.
0A20          05BC    057A    03C0          WORD      ZERO,CON,SEMIS
;
;
0A26          4485    454F    BE53          HEAD      205,DOES,276,DOES,DOCOL      ; ***** DOES>
0A30          03EE    0860    08A4          WORD      FROMR,LATES,PFA,STORE,PSCOD
DODOE: 0A3A    B126          MOV      IP,-(RP)
0A3C          B484          MOV      (W)+,IP
0A3E          B0A5          MOV      W,-(S)
0A40          B502    73DA          NEXT
;
;
0A44          4385    554F    D44E          HEAD      205,COUN,324,COUNT,DOCOL      ; ***** COUNT
; CONVERT STRING TO THE FORMAT USED BY `TYPE'.
0A4E          04C4    0754    04B0          WORD      DUP,ONEP,SWAP,CAT,SEMIS
;
;
0A58          5484    5059    A045          HEAD      204,TYPE,240,TYPE,DOCOL      ; ***** TYPE
0A62          0822    0154    0018          WORD      DDU, ZBRAN,XXL2-.,OVER,PLUS,SWAP,XDO
0A70          01D4    0508    02E8          XXL1:    WORD      I,CAT,EMIT,XLOOP,XXL1-.,BRAN,XXL3-.
0A7E          04A0          XXL2:    WORD      DROP
0A80          03C0          XXL3:    WORD      SEMIS
;
;
0A82          3D86    4543    4C4C          HEAD      206,=CELLS,240,ECELL,DOCOL      ; ***** =CELLS
; NOTE--NEEDED TO FORCE EVEN ADDRESS.
0A8E          04C4    05C4    0368          WORD      DUP,ONE,AND,PLUS,SEMIS
;
;
0A98          2D89    5254    4941          HEAD      211,-TRAILIN,307,DTRAI,DOCOL      ; ***** -TRAILING
0AA6          04C4    05BC    01C2          WORD      DUP,ZERO,XDO
0AAC          048E    048E    043A          XXW6:    WORD      OVER,OVER,PLUS,ONE,SUB,CAT
0AB8          05DE    079E    0154          WORD      BL,SUB,ZBRAN,XXW7-.,LEAVE,BRAN,XXWA-.
0AC6          05C4    079E          XXW7:    WORD      ONE,SUB
0ACA          0170    FFE0    03C0          XXWA:    WORD      XLOOP,XXW6-.,SEMIS
;
;
0AD0          2884    222E    A029          HEAD      204,(.),240,PDOTQ,DOCOL      ; ***** (".)
; USED ONLY BY COMPILER. COMPILED BY `.'"
0ADA          03FA    0A4C    04C4          WORD      R,COUNT,DUP,ONEP,ECELL
0AE4          03EE    043A    03E0          WORD      FROMR,PLUS,TOR,TYPE,SEMIS
;
;
0AEE          2EC2    A022    0AD0          HEAD      302,.",240,DOTQ,DOCOL      ; ***** ."
uA/FORTH      [170,10]
FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```



```

; TYPE ASCII MESSAGE.
0AF6 0120 0022 06E8 WORD LIT,34.,STATE,AT,ZBRAN,XXL6-.
0B02 0964 0AD8 0C4C WORD COMP,PDOTQ,WORD,HERE,CAT,ONEP,ECELL
0B10 0782 0142 000A WORD ALLOT,BRAN,XXL7-.
0B16 0C4C 0772 0A4C XXL6: WORD WORD,HERE,COUNT,TYPE
0B1E 03C0 XXL7: WORD SEMIS
;
0B20 3F86 4C41 4749 HEAD 206,?ALIGN,240,QALIG,DOCOL ; ***** ?ALIGN
0B2C 0772 05C4 0368 WORD HERE,ONE,AND,ALLOT,SEMIS
;
;
0B36 4586 5058 4345 HEAD 206,EXPECT,240,EXPEC ; ***** EXPECT
; READ N CHARACTERS TO MEMORY (AND TERMINATE WITH NULLS).
; ( ADDRESS N ==>).
0B42 0044 KBD ; LEAVES INPUT BUFFER INDEX IN R2
; RETURNS LINE OF INPUT
0B44 B540 MOV (S)+,R0 ; PUT COUNT IN R0
0B46 B541 MOV (S)+,R1 ; PUT DEST ADDR IN R1
0B48 0060 0024 CTRLC XXX4
0B4C C5CA 000A CMPB #12,(R2) ; IS FIRST CHAR LF?
0B50 030A BEQ XXX3 ; IF YES FINISH UP
0B52 D491 XXL1: MOVB (R2)+,(R1)+
0B54 C5CA 000A CMPB #12,(R2) ; TEST FOR LF
0B58 0301 BEQ XXX2 ; QUIT MOVING IF LF FOUND
0B5A 7605 SOB R0,XXX1
0B5C C5F1 000D FFFF XXL2: CMPB #15,-1(R1) ; TEST FOR CR
0B62 0201 BNE XXX3
0B64 0850 SUBI 1,R1 ; REMOVE IT IF FOUND
0B66 8B51 XXL3: CLRB (R1)+ ; AND APPEND NULLS
0B68 8B51 CLRB (R1)+
0B6A 8B49 CLRB (R1)
0B6C B502 73DA NEXT
0B70 0084 0017 0000 XXL4: JOBIDX R0,JOBSTS
0B76 45C8 0080 BIC #J.CCC,(R0) ; REMOVE IMPENDING CNTL-C
0B7A 0046 435E 000D TYPEOCR <^C>
0B80 01F2 BR XXX3
;
;
0B82 5185 4555 D952 HEAD 205,QUER,331,QUERY,DOCOL ; ***** QUERY
0B8C 0632 04FA 0120 WORD TIB,AT,LIT,120,EXPEC,ZERO,IN,STORE,SEMIS
;
0B9E 80C1 0B82 0554 HEAD 301,,200,NULL,DOCOL ; ***** THE NULL
; THE NULL OPERATION (ASCII 0) STOPS INTERPRETATION/COMPILATION
; AT END OF A TERMINAL INPUT LINE, OR A DISK SCREEN. ALL DISK
; BUFFERS MUST TERMINATE WITH NULLS, AND 'EXPECT' PLACES NULLS
; AFTER EACH TERMINAL INPUT LINE.
0BA4 0694 04FA WORD BLK,AT
0BA8 0154 0026 05C4 WORD ZBRAN,XXJ2-. ,ONE,BLK,PSTOR,ZERO,IN,STORE
0BB8 0694 04FA 0600 WORD BLK,AT,BSCR,MOD,ZEQU,ZBRAN,XXJ1-. ,QEXEC,FROMR,DROP
0BCC 0142 0006 XXL1: WORD BRAN,XXJ4-.
0BD0 03EE 04A0 XXL2: WORD FROMR,DROP
0BD4 03C0 XXL4: WORD SEMIS
;

```

```

0BD6 4684 4C49 A04C HEAD 204,FILL,240,FILL,DOCOL ; ***** FILL
0BE0 04B0 03E0 048E WORD SWAP,TOR,OVER,CSTOR,DUP,ONEP,FROMR
0BEE 05C4 079E 0310 WORD ONE,SUB,CMOVE,SEMIS
;
0BF6 4585 4152 C553 HEAD 205,ERAS,305,ERASE,DOCOL ; ***** ERASE
0C00 05BC 0BDE 03C0 WORD ZERO,FILL,SEMIS
;
0C06 4286 414C 4B4E HEAD 206,BLANKS,240,BLANK,DOCOL ; ***** BLANKS
0C12 05DE 0BDE 03C0 WORD BL,FILL,SEMIS
;
0C18 4884 4C4F A044 HEAD 204,HOLD,240,HOLD,DOCOL ; ***** HOLD
0C22 0120 FFFF 0726 WORD LIT,-1,HLD,PSTOR,HLD,AT,CSTOR,SEMIS
;
0C32 5083 C441 0C18 HEAD 203,PA,304,PAD,DOCOL ; ***** PAD
0C3A 0772 0120 0044 WORD HERE,LIT,104,PLUS,SEMIS
;
0C44 5784 524F A044 HEAD 204,WORD,240,WORD,DOCOL ; ***** WORD
0C4E 0694 04FA 0154 WORD BLK,AT,ZBRAN,XXI1-. ,BLK,AT,BLOCK,BRAN,XXI2-.
0C60 0632 04FA XXI1: WORD TIB,AT
0C64 069E 04FA 043A XXI2: WORD IN,AT,PLUS,SWAP,ENCL,HERE,LIT,42,BLANK,IN
0C78 04D2 048E 079E WORD PSTOR,OVER,SUB,TOR,R,HERE,CSTOR,PLUS
0C88 0772 0754 03EE WORD HERE,ONEP,FROMR,CMOVE,SEMIS
;
;
0C92 2888 554E 424D HEAD 210,(NUMBER),240,PNUMB,DOCOL ; ***** (NUMBER)
0CA0 0754 04C4 03E0 XXF3: WORD ONEP,DUP,TOR,CAT,BASE,AT,DIGIT
0CAE 0154 002C 04B0 WORD ZBRAN,XXG4-. ,SWAP,BASE,AT,USTAR,DROP
0CBC 07F2 06F4 04FA WORD ROT,BASE,AT,USTAR,DPLUS
0CC6 06FE 04FA 0754 WORD DPL,AT,ONEP,ZBRAN,XXG5-. ,ONE,DPL,PSTOR
0CD6 03EE 0142 FFC6 XXG5: WORD FROMR,BRAN,XXF3-.
0CDC 03EE 03C0 XXG4: WORD FROMR,SEMIS
;
;
0CE0 4E86 4D55 4542 HEAD 206,NUMBER,240,NUMB,DOCOL ; ***** NUMBER
0CEC 05BC 05BC 07F2 WORD ZERO,ZERO,ROT,DUP,ONEP,CAT,LIT,55,EQUAL
0CFE 04C4 03E0 043A WORD DUP,TOR,PLUS,LIT,-1
0D08 06FE 051C 0C9E XXF6: WORD DPL,STORE,PNUMB,DUP,CAT,BL,SUB
0D16 0154 0016 04C4 WORD ZBRAN,XXF7-. ,DUP,CAT,LIT,56,SUB
0D24 05BC 08CE 05BC WORD ZERO,QERR,ZERO,BRAN,XXF6-.
0D2E 04A0 03EE 0154 XXF7: WORD DROP,FROMR,ZBRAN,XXFA-. ,DMINU
0D38 03C0 XXFA: WORD SEMIS
;
;
0D3A 2D85 4946 C44E HEAD 205,-FIN,304,DFIND,DOCOL ; ***** -FIND
0D44 05DE 0C4C 0772 WORD BL,WORD,HERE,COUNT,UPPER,HERE,CONT,AT,AT,PFIND
0D58 04C4 0408 0154 WORD DUP,ZEQU,ZBRAN,XXE3-. ,DROP,HERE,LATES,PFIND
0D68 03C0 XXE3: WORD SEMIS
;
;
0D6A 5585 5050 D245 HEAD 205,UPPE,322,UPPER,DOCOL ; ***** UPPER
; SETS STRINGS TO UPPER CASE--TO ALLOW
; LOWER AS WELL AS UPPER CASE FROM TERMINAL.
0D74 048E 043A 04B0 WORD OVER,PLUS,SWAP,XDO
0D7C 01D4 0508 0120 XXE2: WORD I,CAT,LIT,140,GREAT,I,CAT,LIT,173,LESS
0D90 0368 0154 000A WORD AND,ZBRAN,XXE1-. ,I,LIT,40,TOGGL
0D9E 0170 FFDC 03C0 XXE1: WORD XLOOP,XXE2-. ,SEMIS
;

```

```

ODA4 2887 4241 524F HEAD 207,(ABORT,251,PABOR,DOCOL ; ***** (ABORT)
ODB0 1022 03C0 WORD ABORT,SEMIS

;
ODB4 4585 5252 D24F HEAD 205,ERRO,322,ERROR,DOCOL ; ***** ERROR
ODBE 064C 04FA 0422 WORD WARN,AT,ZLESS,ZBRAN,XXN4-. ,PABOR
ODCA 0772 0A4C 0A60 XXN4: WORD HERE,COUNT,TYPE,PDOTQ
ODD2 03 BYTE 3
ODD3 3F20 20 ASCII / ? /
ODD6 EVEN
ODD6 1404 03A0 069E WORD MESS,SPSTO,IN,AT,BLK,AT,QUIT,SEMIS

;
ODE6 4983 AE44 0DB4 HEAD 203,ID,256,IDDOT,DOCOL ; ***** ID.
ODEE 0C38 0120 0020 WORD PAD,LIT,40,LIT,137,FILL,DUP
ODFC 08A4 0870 048E WORD PFA,LFA,OVER,SUB,PAD,SWAP,CMOVE
OE0A 0C38 0A4C 0120 WORD PAD,COUNT,LIT,37,AND,TYPE,SPACE,SEMIS

;
OE1A 4386 4552 5441 HEAD 206,CREATE,240,CREAT,DOCOL ; ***** CREATE
OE26 0D42 0154 0010 WORD DFIND,ZBRAN,XXD2-. ,DROP,NFA,IDDOT
OE32 0120 0004 1404 WORD LIT,4,MESS,SPACE
OE3A 0772 04C4 0508 XXD2: WORD HERE,DUP,CAT,WIDTH,AT,MIN,ONEP,ALLOT
OE4A 0B2A 04C4 0120 WORD QALIG,DUP,LIT,240,TOGGL,HERE,ONE,SUB
OE5A 0120 0080 04E8 WORD LIT,200,TOGGL,LATES,COMMA
OE64 06DC 04FA 051C WORD CURR,AT,STORE
OE6A 0772 0762 078E WORD HERE,TWOP,COMMA,SEMIS

;
OE72 5BC9 4F43 504D HEAD 311,[COMPILE,335,BCOMP,DOCOL ; ***** [COMPILE]
OE80 0D42 0408 05BC WORD DFIND,ZEQU,ZERO,QERR,DROP
OE8A 0880 078E 03C0 WORD CFA,COMMA,SEMIS

;
OE90 4CC7 5449 5245 HEAD 307,LITERA,314,LITER,DOCOL ; ***** LITERAL
OE9C 06E8 04FA 0154 WORD STATE,AT,ZBRAN,XXD6-. ,COMP,LIT,COMMA
OEAA 03C0 XXD6: WORD SEMIS

;
OEAC 44C8 494C 4554 HEAD 310,DLITERAL,240,DLITE,DOCOL ; ***** DLITERAL
OEBA 06E8 04FA 0154 WORD STATE,AT,ZBRAN,XXN5-. ,SWAP,LITER,LITER
OEC8 03C0 XXN5: WORD SEMIS

;
OECA 5582 A03C 0EAC HEAD 202,U,240,ULESS,,LT1 ; ***** U<
; UNSIGNED LESS-THAN, NEEDED FOR ADDRESSES.
; : U< >R 0 R> 0 DMINUS D+ SWAP DROP 0< ;
OED2 954D CMP (S)+,(S)
OED4 8203 BHI CC1$ ;BRANCH IF TRUE
OED6 0B4D CLR (S)
OED8 B502 73DA NEXT
OEDC B5CD 0001 CC1$: MOV #1,(S)
OEE0 B502 73DA NEXT

;
OEE4 3F86 5453 4341 HEAD 206,?STACK,240,QSTAC,DOCOL ; ***** ?STACK
; ERROR CHECK.
OEF0 061E 04FA 05CC WORD SZERO,AT,TWO,SUB,SPAT,ULESS,ONE,QERR
OF00 0392 0772 0120 WORD SPAT,HERE,LIT,200,PLUS,ULESS,TWO,QERR
OF10 03C0 WORD SEMIS

;
OF12 4989 544E 5245 HEAD 211,INTERPRE,324,INTER,DOCOL ; ***** INTERPRET
uA/FORTH [170,10] FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

```

OF20 0D42
OF22 0154 001E 06E8
OF2C 0154 000A 0880
OF38 0880 0132
OF3C 0EEE 0142 001C
OF42 0772 0CEA 06FE
OF56 04A0 0E9A
OF5A 0EEE
OF5C 0142 FFC2
OF60 03C0

;
OF62 4989 4D4D 4445
OF70 0860 0120 0040

;
OF7A 568A 434F 4241
OF8A 0A1E 0120 A081
OF9A 0772 0672 04FA
OFA8 0762 06CE 051C
DOVOC: WORD TWOP,CONT,STORE,SEMIS

;
;
OFB0 46C5 524F C854
OFBA OFA8
OFBC A081
OFBE 1916
OFC0 0000
XXVOC: WORD 0 ; THE VOCABULARY LINK (FOR FUTURE USE)

;
;
OFC2 448B 4645 4E49
OFD2 06CE 04FA 06DC

;
OFDC A8C1 OFC2 0554
OFE2 0120 0029 0C4C

;
OFEA 5184 4955 A054
OFF4 05BC 0694 051C
OFFC 03B0 0306 0B8A
1008 0408 0154 0008
1010 03
1011 4F20 4B
1014
1014 0142 FFE6
1018 03C0
XXB2: WORD BRAN,XXB1-.
WORD SEMIS ; ONLY FOR RELOCATION CODE

;
;
101A 4185 4F42 D452
1024 03A0 09C6 0810
102A 0306 0AD8
102E 10
102F 4175 462F 524F
1040
1040 0FB8 OFD0 OFF2
1046 03C0
WORD FORTH,DEFIN,QUIT
WORD SEMIS ; ONLY FOR RELOCATION CODE

;
; COLD AND WARM STARTS
;

```

```

1048 0010 4D3A 0200 FDDB: WORD ^H010,XUP+100,512.,0,0,0,0,0
1058 1C03 RAD50 /DSK/
105A 00 BYTE 0
105B FF BYTE -1
105C 27EA RAD50 /FOR/
105E 7E40 RAD50 /TH /
1060 774A RAD50 /SCR/
1062 7808 0000 BYTE 10,170,0,0
1066 0000 0000 WORD 0,0
106A 63B7 514A WORD 61667,50512
106E 4144 2F51 WORD 40504,27521
1072 2837 2742 WORD 24067,23502
1076 2020 2020 463A ASCII / :FORTH .SCR/
1085 00 BYTE 0
1086 EVEN

;
1086 4384 4C4F A044 HEAD 204,COLD,240,COLD ; ***** COLD
1090 CENT: ; COLD START ENTRY POINT
1090 BDF7 EF78 FF28 MOV ORIGIN+14,FORTH+6 ; SET `FORTH` VOCABULARY FROM STARTUP TABLE
1096 BDC3 EF76 MOV ORIGIN+20,U ; INITIALIZE USER POINTER
; NOTE--INITIALIZES AREAS IN HIGH MEMORY
; CLEAR DISK BUFFERS ON FIRST TIME THROUGH
109A BDC0 EF84 MOV ORIGIN+42,R0 ; `FIRST`--BEGINNING OF DISK BUFFERS
109E BDC1 EF82 MOV ORIGIN+44,R1 ; `LIMIT`--JUST BEYOND DISK BUFFERS
10A2 0B50 01$: CLR (R0)+
10A4 9001 CMP R0,R1
10A6 81FD BMI 01$
; NOW INITIALIZE `OUT`, `OFFSET`, `USE` AND `PREV`
10A8 0B73 0022 CLR 42(U) ; CLEAR `OUT`
10AC 0B73 0026 CLR 46(U) ; CLEAR `OFFSET`
10B0 BDF3 EF6E 003A MOV ORIGIN+42,72(U) ; TO `USE`
10B6 BDF3 EF68 003C MOV ORIGIN+42,74(U) ; TO `PREV`
; END OF SPECIAL HIGH-MEMORY INITIALIZE
10BC 087B MOVI 14,R1 ; ON COLD START, MOVE 12. WORDS
10BE 0115 BR W2
10C0 WENT: ; WARM START ENTRY POINT

10C0 BDC3 EF4C MOV ORIGIN+20,U ; INITIALIZE USER POINTER
10C4 7237 EF38 LEA R0,ORIGIN ; CHECK FOR ATTEMPT TO RELOCATE
; WITHOUT RELOADING SYSTEM
10C8 9033 003E CMP R0,76(U) ; TO `ORIGIN`
10CC 030D BEQ 02$
10CE 0046 4E49 4156 TYPECR <INVALID MEMORY BASE>
10E6 0049 EXIT
10E8 0874 02$: MOVI 5,R1 ; ON WARM START, MOVE FIVE WORDS
10EA 7377 EF24 W2: LEA R5,ORIGIN+22 ; START MOVING FROM HERE
10EE BDC0 EF1E MOV ORIGIN+20,R0 ; MOVE TO THE USER AREA
10F2 0805 ADDI 6,R0 ; PLUS 6
10F4 B550 P1$: MOV (R5)+,(R0)+
10F6 7642 SOB R1,P1$
10F8 0084 1017 002C JOBGET R0,JOBTRM ; FETCH INDEX TO TERMINAL LINE TABLE
10FE B5C1 0010 MOV #^H010,R1 ; BIT 4 = LOWER CASE INPUT OK
1102 5048 BIS R1,@R0 ; SET TERMINAL STATUS WORD
1104 BDC6 EF0C MOV ORIGIN+24,RP ; INITIALIZE THE RETURN-STACK POINTER
uA/FORTH [170,10] FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

```

; NOW SET FORTH'S INSTRUCTION COUNTER, AND GO
1108 7337 FF18      LEA    IP,ABORT+2      ; START EXECUTION WITH `ABORT'
110C B502 73DA      NEXT
;
;
1110 5384 3E2D A044  HEAD    204,S->D,240,STOD      ; ***** S->D
111A 0A8D      TST     (S)
111C 0D65      SXT     -(S)
111E B502 73DA      NEXT
;
; NOTE--THIS SYSTEM DOESN'T NEED THE OPERATIONS `+-' AND `D+-',
; BECAUSE `M*' AND `M/' ARE DEFINED IN CODE.
;
1122 4183 D342 1110  HEAD    203,AB,323,ABS,DOCOL      ; ***** ABS
112A 04C4 0422 0154  WORD    DUP,ZLESS,ZBRAN,XXR5-.,MINUS
1134 03C0      XXR5:  WORD    SEMIS
;
1136 4484 4241 A053  HEAD    204,DABS,240,DABS,DOCOL      ; ***** DABS
1140 04C4 0422 0154  WORD    DUP,ZLESS,ZBRAN,XXRB-.,DMINU
114A 03C0      XXR8:  WORD    SEMIS
;
114C 4D83 CE49 1136  HEAD    203,MI,316,MIN,DOCOL      ; ***** MIN
1154 048E 048E 07D8  WORD    OVER,OVER,GREAT,ZBRAN,XXR7-.,SWAP
1160 04A0 03C0      XXR7:  WORD    DROP,SEMIS
;
1164 4D83 D841 114C  HEAD    203,MA,330,MAX,DOCOL      ; ***** MAX
116C 048E 048E 07C0  WORD    OVER,OVER,LESS,ZBRAN,XXR6-.,SWAP
1178 04A0 03C0      XXR6:  WORD    DROP,SEMIS
;
117C 4D82 A02A 1164  HEAD    202,M*,240,MSTAR      ; ***** M*
1184 BD66 0002      MOV    2(S),-(RP)      ; USE RETURN STACK FOR SAVING SIGN
1188 8002      BPL    SS1$
118A 0C75 0002      NEG    2(S)      ; GET ABSOLUTE VALUE
118E 0A8D      SS1$:  TST    (S)
1190 8002      BPL    SS2$
1192 0C4E      NEG    (RP)      ; ADJUST SIGN WHICH WAS SAVED
1194 0C4D      NEG    (S)      ; GET ABSOLUTE VALUE
1196 71F7 F19C      SS2$:  JSR    PC,UMULT
119A 0A96      TST    (RP)+      ; NEGATIVE RESULT?
119C 8003      BPL    SS3$      ; NO
; IF GET HERE, NEGATE THE DOUBLE-INTEGGER NUMBER ON THE STACK
119E 0C55      NEG    (S)+
11A0 0C4D      NEG    (S)
11A2 8DE5      SBC    -(S)
11A4 B502 73DA      SS3$:  NEXT
;
11A8 4D82 A02F 117C  HEAD    202,M/,240,MSLAS      ; ***** M/
11B0 BD66 0002      MOV    2(S),-(RP)      ; SAVE DIVIDEND SIGN
11B4 0201      BNE    S5$      ; ZERO WOULDN'T INDICATE
11B6 0C8E      INC    (RP)      ; A SIGN CHANGE
11B8 B3A6      S5$:  MOV    (RP),-(RP)      ; DUPLICATE IT
11BA 8008      BPL    S1$
; IF GET HERE, TAKE ABSOLUTE VALUE OF DOUBLE-INTEGGER DIVIDEND
11BC 0C35 0002      COM    2(S)
uA/FORTH [170,10]
FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

11C0	0C35	0004		COM	4(S)		
11C4	0CB5	0004		INC	4(S)		
11C8	8DB5	0002		ADC	2(S)		
11CC	0A8D			TST	(S)		; IS DIVISOR NEGATIVE?
11CE	8002			BPL	S2\$		
11D0	0C4E			NEG	(RP)		; IF YES, NEGATE QUOTIENT SIGN
11D2	0C4D			NEG	(S)		; AND TAKE ABS. VALUE OF DIVISOR
11D4	71F7	F178		JSR	PC,UDIV		
11D8	0A96			TST	(RP)+		; NEGATIVE QUOTIENT?
11DA	8001			BPL	S3\$; NO
11DC	0C4D			NEG	(S)		; NEGATE THE QUOTIENT
11DE	0A96			TST	(RP)+		; NEGATIVE DIVIDEND?
11E0	8002			BPL	S4\$; NEGATE THE REMAINDER
11E2	0C75	0002		NEG	2(S)		
11E6	B502	73DA		NEXT			
11EA	AA81	11A8	0554	HEAD	201,,252,STAR,DOCOL		; ***** *
11F0	1182	04A0	03C0	WORD	MSTAR,DROP,SEMIS		
11F6	2F84	4F4D	A044	HEAD	204,/MOD,240,SLMOD,DOCOL		; ***** /MOD
1200	03E0	1118	03EE	WORD	TOR,STOD,FROMR,MSLAS,SEMIS		
120A	AF81	11F6	0554	HEAD	201,,257,SLASH,DOCOL		; ***** /
1210	11FE	04B0	04A0	WORD	SLMOD,SWAP,DROP,SEMIS		
1218	4D83	C44F	120A	HEAD	203,MO,304,MOD,DOCOL		; ***** MOD
1220	11FE	04A0	03C0	WORD	SLMOD,DROP,SEMIS		
1226	2A85	4D2F	C44F	HEAD	205,*/MO,304,SSMOD,DOCOL		; ***** */MOD
1230	03E0	1182	03EE	WORD	TOR,MSTAR,FROMR,MSLAS,SEMIS		
123A	2A82	A02F	1226	HEAD	202,*/ ,240,SSLA,DOCOL		; ***** */
1242	122E	04B0	04A0	WORD	SSMOD,SWAP,DROP,SEMIS		
124A	4D85	4D2F	C44F	HEAD	205,M/MO,304,MSMOD,DOCOL		; ***** M/MOD
1254	03E0	05BC	03FA	WORD	TOR,ZERO,R,USLAS,FROMR		
125E	04B0	03E0	0346	WORD	SWAP,TOR,USLAS,FROMR,SEMIS		

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

```

; *****
;
;     DISK I/O
;
; *****
;
; `USE' AND `PREV' MOVED TO USER AREA
;
1268    2B84    5542    A046           HEAD    204,+BUF,240,PBUF,DOCOL           ; ***** +BUF
1272    05F4    0120    0004           WORD    BBUF,LIT,4,PLUS,PLUS,DUP,LIMIT,AT,EQUAL
1284    0154    0008    04A0           WORD    ZBRAN,XXT1-. ,DROP,FIRST,AT
128E    04C4    073C    04FA    XXT1:   WORD    DUP,PREV,AT,SUB,SEMIS
;
1298    5586    4450    5441           HEAD    206,UPDATE,240,UPDAT,DOCOL           ; ***** UPDATE
12A4    073C    04FA    04FA           WORD    PREV,AT,AT,LIT,100000,OR,PREV
12B2    04FA    051C    03C0           WORD    AT,STORE,SEMIS
;
12B8    458D    504D    5954           HEAD    215,EMPTY-BUFFER,323,MTBUF,DOCOL           ; ***** EMPTY-BUFFERS
12CA    067E    04FA    068A           WORD    FIRST,AT,LIMIT,AT,OVER,SUB,ERASE,SEMIS
;
12DA    4685    554C    C853           HEAD    205,FLUS,310,FLUSH,DOCOL           ; ***** FLUSH
;   SOME SYSTEMS DEFINE THIS IN THE EDITOR, NOT HERE.
12E4    068A    04FA    067E           WORD    LIMIT,AT,FIRST,AT,XDO
12EE    01D4    04FA    0422    XXTA:   WORD    I,AT,ZLESS,ZBRAN,XXT7-. ,I,TWOP,I,AT
1300    0120    7FFF    0368           WORD    LIT,77777,AND,ZERO,RW
130A    05F4    0120    0004    XXT7:   WORD    BBUF,LIT,4,PLUS,XPLOO,XXTA-. ,MTBUF,SEMIS
;
;
131A    4286    4655    4546           HEAD    206,BUFFER,240,BUFFE,DOCOL           ; ***** BUFFER
1326    0730    04FA    04C4           WORD    USE,AT,DUP,TOR
132E    1270    0154    FFFC    XXT2:   WORD    PBUF,ZBRAN,XXT2-. ,USE,STORE
1338    03FA    04FA    0422           WORD    R,AT,ZLESS,ZBRAN,XXT3-.
1342    03FA    0762    03FA           WORD    R,TWOP,R,AT,LIT,77777,AND
1350    05BC    18EA           WORD    ZERO,RW
135A    03FA    051C    03FA    XXT3:   WORD    R,STORE,R,PREV,STORE,FROMR,TWOP,SEMIS
;
1364    4285    4F4C    CB43           HEAD    205,BLOC,313,BLOCK,DOCOL           ; ***** BLOCK
136E    06C0    04FA    043A           WORD    OFFSET,AT,PLUS,TOR
1376    073C    04FA    04C4           WORD    PREV,AT,DUP,AT,R,SUB,DUP,PLUS,ZBRAN,XXT4-.
138A    1270    0408    0154    XXT5:   WORD    PBUF,ZEQU,ZBRAN,XXT6-.
1392    04A0    03FA    1324           WORD    DROP,R,BUFFE
1398    04C4    03FA    05C4           WORD    DUP,R,ONE,RW,TWO,SUB
13A4    04C4    04FA    03FA    XXT6:   WORD    DUP,AT,R,SUB,DUP,PLUS,ZEQU
13B2    0154    FFD6           WORD    ZBRAN,XXT5-.
13B6    04C4    073C    051C           WORD    DUP,PREV,STORE
13BC    03EE    04A0    0762    XXT4:   WORD    FROMR,DROP,TWOP,SEMIS
;
;
;
;
;
;
;
;
;
13C4    2886    494C    454E           HEAD    206,(LINE),240,PLINE,DOCOL           ; ***** (LINE)
uA/FORTH    [170,10]    FORTH INTEREST GROUP    POB 1105    SAN CARLOS, CA 94070

```


uA/FORTH

[170,10]

MACRO ASSEMBLY LISTING

13D0 03E0 05E8 05F4
 13DC 11EE 043A 136C

 13E8 2E85 494C C54E
 13F2 13CE 0AA4 0A60

 13FA 4D87 5345 4153
 1406 064C 04FA 0154
 1418 06C0 04FA 0600
 1424 0142 000E
 1428 0AD8
 142A 06
 142B 534D 2047 2023
 1432
 1432 1700
 1434 03C0

 1436 4C84 414F A044
 1440 0694 04FA 03E0
 1452 0600 11EE 0694
 1462 03EE 0694 051C

 146A 2DC3 BE2D 1436
 1472 094A 05BC 069E
 1482 121E 079E 0694

```

WORD TOR,CL,BBUF,SSMOD,FROMR,BSCR
WORD STAR,PLUS,BLOCK,PLUS,CL,SEMIS
;
HEAD 205,.LIN,305,DLINE,DOCOL ; ***** .LINE
WORD PLINE,DTRAI,TYPE,SEMIS
;
HEAD 207,MESSAG,305,MESS,DOCOL ; ***** MESSAGE
WORD WARN,AT,ZBRAN,XXW5-. ,DDUP,ZBRAN,XXW3-. ,LIT,4
WORD OFFSET,AT,BSCR,SLASH,SUB,DLINE
XXW3: WORD BRAN,XXW4-.
XXW5: WORD PDOTQ
      BYTE 6
      ASCII /MSG # /
      EVEN
WORD DOT
XXW4: WORD SEMIS
;
HEAD 204,LOAD,240,LOAD,DOCOL ; ***** LOAD
WORD BLK,AT,TOR,IN,AT,TOR,ZERO,IN,STORE
WORD BSCR,STAR,BLK,STORE,INTER,FROMR,IN,STORE
WORD FROMR,BLK,STORE,SEMIS
;
HEAD 303,--,276,ARROW,DOCOL ; ***** -->
WORD QLOAD,ZERO,IN,STORE,BSCR,BLK,AT,OVER
WORD MOD,SUB,BLK,PSTOR,SEMIS
;
;
;
;
;

```

uA/FORTH

[170,10]

PAGE

```

; *****
;
; MISCELLANEOUS HIGHER LEVEL
; *****
;
;
148C   A7C1   146A   0554   HEAD   301,,247,TICK,DOCOL   ; *****
1492   0D42   0408   05BC   WORD   DFIND,ZEQU,ZERO,QERR,DROP,LITER,SEMIS
;
14A0   4686   524F   4547   HEAD   206,FORGET,240,FORGE,DOCOL   ; ***** FORGET
14AC   06DC   04FA   06CE   WORD   CURR,AT,CONT,AT,SUB,LIT,30,QERR,TICK,DUP
14C0   0658   04FA   07C0   WORD   FENCE,AT,LESS,LIT,25,QERR
14CC   04C4   088E   0662   WORD   DUP,NFA,DP,STORE,LFA,AT,CONT,AT
14DC   051C   03C0           WORD   STORE,SEMIS
;
;
;
14E0   4284   4341   A04B   HEAD   204,BACK,240,BACK,DOCOL   ; ***** BACK
14EA   0772   079E   078E   WORD   HERE,SUB,COMMA,SEMIS
;
14F2   42C5   4745   CE49   HEAD   305,BEGI,316,BEGIN,DOCOL   ; ***** BEGIN
14FC   08E8   0772   05C4   WORD   QCOMP,HERE,ONE,SEMIS
;
1504   45C5   444E   C649   HEAD   305,ENDI,306,ENDIF,DOCOL   ; ***** ENDIF
150E   08E8   05CC   0918   WORD   QCOMP,TWO,QPAIR,HERE,OVER,SUB,SWAP,STORE,SEMIS
;
1520   54C4   4548   A04E   HEAD   304,THEN,240,THEN,DOCOL   ; ***** THEN
152A   150C   03C0           WORD   ENDIF,SEMIS
;
152E   44C2   A04F   1520   HEAD   302,DO,240,DO,DOCOL   ; ***** DO
1536   0964   01C2   0772   WORD   COMP,XDO,HERE,LIT,3,SEMIS
;
1542   4CC4   4F4F   A050   HEAD   304,LOOP,240,LOOP,DOCOL   ; ***** LOOP
154C   0120   0003   0918   WORD   LIT,3,QPAIR,COMP,XLOOP,BACK,SEMIS
;
155A   2BC5   4F4C   D04F   HEAD   305,+LOO,320,PLOOP,DOCOL   ; ***** +LOOP
1564   0120   0003   0918   WORD   LIT,3,QPAIR,COMP,XPLOO,BACK,SEMIS
;
1572   55C5   544E   CC49   HEAD   305,UNTI,314,UNTIL,DOCOL   ; ***** UNTIL
157C   05C4   0918   0964   WORD   ONE,QPAIR,COMP,ZBRAN,BACK,SEMIS
;
1588   45C3   C44E   1572   HEAD   303,EN,304,END,DOCOL   ; ***** END
1590   157A   03C0           WORD   UNTIL,SEMIS
;
1594   41C5   4147   CE49   HEAD   305,AGAI,316,AGAIN,DOCOL   ; ***** AGAIN
159E   05C4   0918   0964   WORD   ONE,QPAIR,COMP,BRAN,BACK,SEMIS
;
15AA   52C6   5045   4145   HEAD   306,REPEAT,240,REPEAT,DOCOL   ; ***** REPEAT
15B6   03E0   03E0   159C   WORD   TOR,TOR,AGAIN,FROMR,FROMR,TWO,SUB,ENDIF,SEMIS
;
15C8   49C2   A046   15AA   HEAD   302,IF,240,IF,DOCOL   ; ***** IF
15D0   0964   0154   0772   WORD   COMP,ZBRAN,HERE,ZERO,COMMA,TWO,SEMIS
;

```

```

;
15DE 45C4 534C A045 HEAD 304,ELSE,240,ELSE,DOCOL ; ***** ELSE
15E8 05CC 0918 0964 WORD TWO,QPAIR,COMP,BRAN,HERE,ZERO,COMMA
15F6 04B0 05CC 150C WORD SWAP,TWO,ENDIF,TWO,SEMIS
;
1600 57C5 4948 C54C HEAD 305,WHIL,305,WHILE,DOCOL ; ***** WHILE
160A 15CE 0762 03C0 WORD IF,TWOP,SEMIS
;
;
;
1610 5386 4150 4543 HEAD 206,SPACES,240,SPACS,DOCOL ; ***** SPACES
161C 05BC 116A 0822 WORD ZERO,MAX,DDUP,ZBRAN,XXR4-. ,ZERO,XDO
162A 0810 0170 FFFC XXRA: WORD SPACE,XLOOP,XXRA-.
1630 03C0 XXR4: WORD SEMIS
;
1632 3C82 A023 1610 HEAD 202,#,240,BDIGS,DOCOL,,LT2 ; ***** <#
163A 0C38 0726 051C WORD PAD,HLD,STORE,SEMIS
;
1642 2382 A03E 1632 HEAD 202,#>,240,EDIGS,DOCOL ; ***** #>
164A 04A0 04A0 0726 WORD DROP,DROP,HLD,AT,PAD,OVER,SUB,SEMIS
;
165A 5384 4749 A04E HEAD 204,SIGN,240,SIGN,DOCOL ; ***** SIGN
1664 07F2 0422 0154 WORD ROT,ZLESS,ZBRAN,XXR1-. ,LIT,55,HOLD
1672 03C0 XXR1: WORD SEMIS
;
1674 A381 165A 0554 HEAD 201,,243,DIG,DOCOL ; ***** #
167A 06F4 04FA 1252 WORD BASE,AT,MSMOD,ROT,LIT,11,OVER,LESS
168A 0154 0008 0120 WORD ZBRAN,XXR2-. ,LIT,7,PLUS
1694 0120 0030 043A XXR2: WORD LIT,60,PLUS,HOLD,SEMIS
;
169E 2382 A053 1674 HEAD 202,#S,240,DIGS,DOCOL ; ***** #S
16A6 1678 048E 048E XXR3: WORD DIG,OVER,OVER,OR,ZEQU,ZBRAN,XXR3-. ,SEMIS
;
16B6 4483 D22E 169E HEAD 203,D.,322,DDOTR,DOCOL ; ***** D.R
16BE 03E0 04B0 048E WORD TOR,SWAP,OVER,DABS,BDIGS,DIGS,SIGN,EDIGS
16CE 03EE 048E 079E WORD FROMR,OVER,SUB,SPACS,TYPE,SEMIS
;
16DA 2E82 A052 16B6 HEAD 202,.R,240,DOTR,DOCOL ; ***** .R
16E2 03E0 1118 03EE WORD TOR,STOD,FROMR,DDOTR,SEMIS
;
16EC 4482 A02E 16DA HEAD 202,D.,240,DDOT,DOCOL ; ***** D.
16F4 05BC 16BC 0810 WORD ZERO,DDOTR,SPACE,SEMIS
;
16FC AE81 16EC 0554 HEAD 201,,256,DOT,DOCOL ; ***** .
1702 1118 16F2 03C0 WORD STOD,DDOT,SEMIS
;
1708 BF81 16FC 0554 HEAD 201,,277,QUEST,DOCOL ; ***** ?
170E 04FA 1700 03C0 WORD AT,DOT,SEMIS
;
1714 5582 A02E 1708 HEAD 202,U.,240,UDOT,DOCOL ; ***** U.
171C 05BC 16F2 03C0 WORD ZERO,DDOT,SEMIS

```

```

; UTILITY SECTION.

```

```

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

```

;
;
1722 4C84 5349 A054 HEAD 204,LIST,240,LIST,DOCOL ; ***** LIST
; ( N---. LIST GIVEN SCREEN.)
172C 09C6 0306 04C4 WORD DEC,CR,DUP,SCR,STORE,PDOTQ
1738 06 BYTE 6
1739 4353 2052 2023 ASCII /SCR # /
1740 EVEN
1740 1700 0120 0010 WORD DOT,LIT,20,ZERO,XDO
174A 0306 01D4 05D4 XXZ1: WORD CR,I,THREE,DOTR,SPACE
1754 01D4 06B2 04FA WORD I,SCR,AT,DLINE,XLOOP,XXZ1-. ,CR,SEMIS
;
1764 4985 444E D845 HEAD 205,INDE,330,INDEX,DOCOL ; ***** INDEX
; LIST FIRST LINE OF A RANGE OF DISK SCREENS.
176E 0306 0754 04B0 WORD CR,ONEP,SWAP,XDO
1776 0306 01D4 05D4 XXZ2: WORD CR,I,THREE,DOTR,SPACE,ZERO,I,DLINE
1786 02FE 0154 0004 WORD QTERM,ZBRAN,XXZ3-. ,LEAVE
178E 0170 FFE6 03C0 XXZ3: WORD XLOOP,XXZ2-. ,SEMIS
;
1794 5485 4952 C441 HEAD 205,TRIA,304,TRIAD,DOCOL ; ***** TRIAD
; LIST DISK SCREENS THREE PER PAGE.
179E 0120 000C 02E8 WORD LIT,14,EMIT ; FORM FEED
17A4 05D4 120E 05D4 WORD THREE,SLASH,THREE,STAR,THREE
17AE 048E 043A 04B0 WORD OVER,PLUS,SWAP,XDO
17B6 0306 01D4 172A XXZ4: WORD CR,I,LIST,XLOOP,XXZ4-. ,CR,LIT,17,MESS,CR,SEMIS
;
17CC 5685 494C D453 HEAD 205,VLIS,324,VLIST,DOCOL ; ***** VLIST
17D6 0120 0080 06A8 WORD LIT,200,OUT,STORE,CONT,AT,AT
17E4 06A8 04FA 0120 XXZ5: WORD OUT,AT,LIT,100,GREAT,ZBRAN,XXZ6-.
17F2 0306 05BC 06A8 WORD CR,ZERO,OUT,STORE
17FA 04C4 0DEC 0810 XXZ6: WORD DUP,IDDOT,SPACE,SPACE,PFA,LFA,AT
1808 04C4 0408 02FE WORD DUP,ZEQU,QTERM,OR,ZBRAN,XXZ5-. ,DROP,SEMIS
;
;
;
;
;
;
;
PAGE

```

```

; *****
;
; INSTALLATION-DEPENDENT SECTION (TERMINAL AND DISK I/O)
;
; *****
;
1818      PERMIT:
1818      9357      0020      CMP      (S),#40      ; TEST FOR CONTROL CHARACTER
181C      0502
181E      0CB3      0022      BLT      E1$
1822      B541      MOV      (S)+,R1      ; INCREMENT `OUT'
1824      0043      TOUT
1826      B502      73DA      NEXT
;
;
;
182A      PKEY:
182A      0084      1017      002C      JOBGET  R0,JOBTRM      ; GET INDEX TO TERMINAL STATUS WORD.
1830      55C8      0003      BIS      #3,@R0      ; SET IMAGE MODE, NOECHO
1834      0042      TIN      ; GET CHAR IN R1
1836      45C8      0003      BIC      #3,@R0      ; CLEAR IMAGE MODE, NOECHO
183A      0060      0006      CTRLC   F6$      ; BRANCH IF CNTL-C CONDITION
183E      B065      MOV      R1,-(S)      ; PUT CHAR ON DATA STACK
1840      B502      73DA      F5$:    NEXT
1844      B5E5      0003      F6$:    MOV      #3,-(S)      ; PUT 3 (CNTL-C) ON STACK
1848      0106      BR      E4$      ; REMOVE IMPENDING CNTL-C
;
;
;
184A      PQTER:  CLR      -(S)
184C      0060      0004      CTRLC   E3$      ; HAS CNTL-C BEEN PRESSED?
1850      B502      73DA      NEXT      ; NO
1854      0C8D      E3$:    INC      (S)      ; CHANGE FALSE VALUE ON STACK TO TRUE
1856      0084      0017      0000      E4$:    JOBIDX  R0,JOBSTS
185C      45C8      0080      BIC      #J.CCC,(R0)      ; REMOVE IMPENDING CNTL-C
1860      B502      73DA      NEXT
;
;
1864      PCR:    CRLF
1866      B502      73DA      NEXT      ; CARRIAGE RETURN AND LINE FEED
;
;
186A      4283      C559      17CC      HEAD      203,BY,305,BYE      ; ***** BYE
1872      0048      CRLF
1874      0049      EXIT
;
; *****
;
; DISK I/O
;
; *****
;
1876      4E84      5452      A053      HEAD      204,NRTS,240,NRTS      ; ***** NRTS
uA/FORTH [170,10] FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

```

1880 B577 F7CC
1884 B540
1886 B037 F7C0
188A 0B70 0100
188E 0B65
1890 0080 5DC0 F7B2
1896 0301
1898 0C8D
189A B502 73DA

189E 4E84 5457 A053

18A8 B577 F7A4
18AC B577 F79A
18B0 0B65
18B2 0080 6DC0 F790
18B8 0301
18BA 0C8D
18BC B502 73DA

18C0 5283 D354 189E

18C8 187E 0154 0004
18D0 03C0

18D2 5783 D354 18C0

18DA 18A6 0154 0004
18E2 03C0

18E4 5283 D72F 18D2

18EC 03E0 05C4 079E
18FE 048E 0754 03EE

1904 0154 000A 18C6
190C 0142 0006

1910 18D8 18D8
1914 03C0

```

```

; ADDR REC -> ERR
; READ A SINGLE RECORD
  MOV (S)+,FDDB+10 ; LOGICAL BLOCK NUMBER
  MOV (S)+,R0
  MOV R0,FDDB+2 ; ADDR OF BUFFER
  CLR 256.(R0) ; CLEAR NULL CELL AT END OF SCREEN
  CLR -(S)
  INPUT FDDB
  BEQ AA1$
  INC (S) ; ERROR DURING READ
AA1$: NEXT
;
; HEAD 204,NWTS,240,NWTS ; ***** NWTS
; ADDR REC -> ERR
; WRITE A SINGLE RECORD
  MOV (S)+,FDDB+10 ; LOGICAL BLOCK NUMBER
  MOV (S)+,FDDB+2
  CLR -(S)
  OUTPUT FDDB
  BEQ AA2$
  INC (S) ; ERROR DURING WRITE
AA2$: NEXT
;
; HEAD 203,RT,323,RTS,DOCOL ; ***** RTS
; ADDR REC ->
; READ A SINGLE RECORD. QUIT IF ERROR
  WORD NRTS,ZBRAN,AA3$-.,QUIT
AA3$: WORD SEMIS
;
; HEAD 203,WT,323,WTS,DOCOL ; ***** WTS
; ADDR RECORD ->
; WRITE A SINGLE RECORD. QUIT IF ERROR
  WORD NWTS,ZBRAN,AA4$-.,QUIT
AA4$: WORD SEMIS
;
;
; HEAD 203,R/,327,RW,DOCOL ; ***** R/W
; READ OR WRITE 1024-BYTE SCREEN.
; ADDR BLOCK# FLAG(R=1, W=0) ->
  WORD TOR,ONE,SUB,DUP,PLUS,OVER,LIT,512.,PLUS
  WORD OVER,ONEP,FROMR
; CHANGE THE SCREEN # TO FIRST SEQUENCE #.
; IF READ, SETUP AND READ TWO 512-BYTE BLOCKS
  WORD ZBRAN,BB1$-.,RTS,RTS
BB2$: WORD BRAN,BB3$-
; SETUP AND WRITE 2 BLOCKS
BB1$: WORD WTS,WTS
BB3$: WORD SEMIS
;
;
;
PAGE

```

MACRO ASSEMBLY LISTING

1916 5484 5341 A04B ; HEAD 204,TASK,240,TASK,DOCOL ; ***** TASK
1920 03C0 ; WORD SEMIS ;

ua/FORTH [170,10] PAGE FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

; *****
;
; STACKS AND BUFFERS
;
; *****
;
; NOTE--`UP' AND DISK BUFFERS ARE
; INITIALIZED AT COLD START, OR AT FIRST TIME THROUGH.
;
1922          EVEN
1922          XDP:          ; DICTIONARY STARTS HERE
1922          RELOC:      ; OVERLAY RELOCATION CODE AND DICTIONARY SPACE
1922          0B77      E6DA      CLR      ORIGIN      ; REMOVE BRANCH AT ORIGIN SO THAT uA/FORTH
1926          0B77      E6D8      CLR      ORIGIN+2    ; CAN BE LOADED AND RERUN FROM MONITOR WITHOUT
; LOSING LOADED DICTIONARY ENTRIES
192A          7237      FFEE      LEA      R0,TASK-2    ; RUN THROUGH DICTIONARY MAKING CFA AND
; LFA ABSOLUTE
;
192E          7277      E6CE      LEA      R1,ORIGIN    ; R1 HOLDS MEMORY BASE ADDRESS
1932          BDC3      E6DA      MOV      ORIGIN+20,U  ; PUT BASE OF USER AREA IN U
1936          1043      ADD      R1,U          ; ADD MEMORY BASE TO U
1938          B073      003E      MOV      R1,76(U)    ; 76(U) IS LOCATION OF `ORIGIN'
193C          B004      Y0:      MOV      R0,R4        ; R4 IS TEMPORARY POINTER INTO PARAMETER FIELD
193E          0901      ADDI     2,R4          ; POINT R4 AT CFA
1940          9517      0554      CMP      (R4)+,#DOCOL ; RELOCATE ADDRESSES IN PARAMETER FIELD
1944          022B      BNE      Y2            ; IF CFA IS `DOCOL'
;
1946          9317      03C0      Y4:      CMP      (R4),#SEMIS    ; CONTINUE UNTIL `SEMIS' OR `PSCOD' FOUND
194A          0327      BEQ      Y6
194C          9317      09F0      CMP      (R4),#PSCOD
1950          0324      BEQ      Y6
1952          9317      0120      CMP      (R4),#LIT    ; CHECK FOR CONSTANTS WHICH APPEAR IN
1956          031B      BEQ      Y3            ; PARAMETER FIELDS
1958          9317      0142      CMP      (R4),#BRAN   ; SEVEN PRIMITIVES LEAVE CONSTANTS IN THE
195C          0318      BEQ      Y3            ; PARAMETER FIELD
195E          9317      0154      CMP      (R4),#ZBRAN
1962          0315      BEQ      Y3
1964          9317      0170      CMP      (R4),#XLOOP
1968          0312      BEQ      Y3
196A          9317      0192      CMP      (R4),#XPLOO
196E          030F      BEQ      Y3
1970          9317      0964      CMP      (R4),#COMP
1974          030F      BEQ      Y7
1976          9317      0AD8      CMP      (R4),#PDOTQ
197A          0207      BNE      Y5
197C          1054      ADD      R1,(R4)+      ; ADD BASE TO PARAMETER
197E          D302      MOVB     (R4),R2      ; R2 CONTAINS LENGTH OF STRING
1980          0881      ADDI     2,R2        ; (SO THAT R2 CAN BE ROUNDED DOWN)
1982          0B82      ASR      R2
;
1984          0AC2      ASL      R2            ; FORCE EVEN COUNT
1986          1084      ADD      R2,R4        ; MOVE POINTER PAST STRING
; WORD AND LOW BYTE ARE EVEN ADDRESSES
;
1988          01DE      BR       Y4
198A          1054      Y5:      ADD      R1,(R4)+    ; ADD BASE TO PARAMETER
;
uA/FORTH     [170,10]      FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```



```

198C 01DC          BR      Y4
198E 1054          Y3:    ADD    R1,(R4)+      ; ADD BASE TO PARAMETER
1990 0901          ADDI   2,R4          ; MOVE POINTER PAST CONSTANT
1992 01D9          BR      Y4
1994 1054          Y7:    ADD    R1,(R4)+
1996 1054          ADD    R1,(R4)+      ; RELOCATE EMBEDDED ADDRESS
1998 01D6          BR      Y4
199A 104C          Y6:    ADD    R1,(R4)
                                ; END OF PARAMETER RELOCATION LOOP
                                ;
199C 1070 0002     Y2:    ADD    R1,2(R0)      ; ADD BASE TO CFA
19A0 0A88          TST    (R0)          ; ZERO EQUALS END OF DICT
19A2 0305          BEQ    Y1
19A4 1048          ADD    R1,(R0)      ; ADD BASE TO LINK
19A6 B200          MOV    (R0),R0
                                ; R0 NOW POINTS TO NFA
19A8 0A90          YMATCH: TST   (R0)+
19AA 80FE          BPL    YMATCH
                                ; R0 NOW POINTS TO LINK FIELD ADDRESS (LFA)
19AC 01C7          BR      Y0
                                ; CONTINUE WITH RELOCATION AND INITIALIZATION
19AE 72B7 F6C4     Y1:    LEA    R2,FDDB+56      ; PREPARE AFORTH.SCR
19B2 E5F7 0001 F691  BISB   #1,FDDB+1      ; TRAP FILE--ERROR ABORTS
19B8 0089 0DD7 F68A  FSPEC  FDDB,SCR
19C0 0080 0DC0 F682  INIT   FDDB
19C6 0080 1DD7 F67C  OPENR  FDDB
19CE 1077 E63A     ADD    R1,ORIGIN+14    ; MAKE ADDRESSES IN BOOT-UP TABLE ABSOLUTE
19D2 1077 E63A     ADD    R1,ORIGIN+20
19D6 1077 E638     ADD    R1,ORIGIN+22
19DA 1077 E636     ADD    R1,ORIGIN+24
19DE 1077 E634     ADD    R1,ORIGIN+26
19E2 1077 E636     ADD    R1,ORIGIN+34
19E6 1077 E634     ADD    R1,ORIGIN+36
19EA 1077 E632     ADD    R1,ORIGIN+40
19EE 1077 E630     ADD    R1,ORIGIN+42
19F2 1077 E62E     ADD    R1,ORIGIN+44
19F6 1077 F5C0     ADD    R1,FORTH+2      ; FIX VOCABULARY LINK STRUCTURE TO ABSOLUTE ADDR
19FA 73F7 F692     JMP    CENT
                                ;
19FE          BLKB   10000.-<.-XDP>
                                ; ADJUST THIS NUMBER FOR A DIFFERENT DICTIONARY SIZE
4032          EVEN
4032          XS0:   ; START OF COMPUTATION STACK
4032          BLKW   2      ; IN CASE OF EMPTY STACK
                                ;
                                ;
4036          DSKBUF: ; THREE 1K DISK BUFFERS ( TWO REQUIRED)
                                ; INITIALIZE BUFFERS' UPDATE BITS, AND TERMINATING NULLS, TO ZERO.
                                ; NOTE--THESE BUFFERS ARE CLEARED AT COLD START
                                ; ALSO THE NUMBER OR LOCATION OF BUFFERS CAN BE
                                ; CHANGED AT RUN TIME.
4036 0000          WORD   0
4038          BLKB   1024.
uA/FORTH [170,10]  FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

```

uA/FORTH

[170,10]

4438
443A
443C
483C
483E
4840
4C40
4C42

MACRO ASSEMBLY LISTING

WORD 0
WORD 0
BLKS 1024.
WORD 0
WORD 0
BLKS 1024.
WORD 0

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 MOVED AT RUN TIME.
XTIB: BLKW 42. ; TERMINAL INPUT BUFFER
BLKW 50. ; FOR RETURN STACK

XR0=.
XUP: BLKW 100 ; ROOM FOR 100 USER VARIABLES
;

END ORIGIN

FORTH INTEREST GROUP POB 1105 SAN CARLOS, CA 94070

4D7A

uA/FORTH

[170,10]

