APPENDIX A. SUMMARY OF FORTH RULES

1. FORTH WORDS ARE MADE UP OF AN ARBITRARY NUMBER OF CHARACTERS, SEPARATED BY SPACES.

2. MOST WORDS REQUIRE PARAMETERS ON A PUSH-DOWN STACK.

3. ANY ERROR MESSAGE EMPTIES BOTH STACKS.

4. ALL PARAMETERS PUT ONTO A STACK MUST BE REMOVED WHEN THEY ARE NO LONGER NEEDED. THE ORDER WILL BE LAST IN, FIRST OUT.

5. ALL WORDS MUST BE DEFINED BEFORE THEY CAN BE USED.

6. COMPILING WORDS MUST NEVER BE USED OUTSIDE A DEFINITION.

7. EVERY IF MUST BE FOLLOWED BY A THEN .

8. ANYTHING PUSHED ONTO THE RETURN STACK MUST BE REMOVED WITHIN THE SAME DEFINITION.

9. WHEN NESTING STRUCTURES IN FORTH, YOU MUST NEST EACH STRUCTURE COMPLETELY WITHIN ANY OUTER STRUCTURE.

10. NEVER PUT AN UNTESTED ROUTINE INTO A LOOP.

11. A STUB MUST REPRODUCE THE BEHAVIOR OF ITS INTENDED COUNTERPART WITH RESPECT TO STACK USAGE.

APPENDIX B. microFORTH GLOSSARY

This glossary includes all words, definitions, and screen assignments that are common to all CPUs. Because of the flexibility of the FORTH language, however, you may find a few exceptions on your diskette. These will have been caused by our programmers' making improvements to the microFORTH system you have received.

Within this glossary there are also a few words whose exact behavior varies from chip to chip because the implementation of each is machine dependent. The end behavior of these words, however, is the same on all machines; the most obvious variations of implementation occur in M* and M/MOD . They are used by */ /MOD */MOD and MOD . Do not use M* and M/MOD unless you understand exactly how these words modify the stack pointer on your particular CPU. Use */ /MOD */MOD and MOD to perform the appropriate arithmetic.

Because they are CPU-dependent, no ASSEMBLER, CROSS-COMPILER, or RECONFIGURE words are given in this glossary. The microFORTH Technical Manual gives specifics of these three application vocabularies by chip type, while the Development System Documentation that you received with your system contains information about any additions or other changes.

Short glossaries for the microFORTH vocabularies that pertain only to Options (such as Extended-Precision Math or File Management) are provided with the options when the number of words warrants it.

The order followed here is that of the ASCII character codes.

WORD VOCABULARY SCREEN STACK: IN OUT

! FORTH 0 2 0

Stores the second number on the stack into the address which is on the top of the stack. For example, if VALUE is a VARIABLE , then 32767 VALUE ! changes VALUE to 32767.

" EDITOR 14 0 0

Used to enter a line of text into PAD; the text is terminated by the delimiter " .

Usage: " TEXT" 1 I

This example inserts TEXT in Line 2 of the current screen.

FORTH 12 1 1

Converts the least significant digit of a 16-bit binary number to its ASCII equivalent using the current BASE. The ASCII character is placed in the output string.

#> FORTH 12 1 2

Terminates the pictured numeric output, leaving the byte count of the string on top of the stack and its address beneath for TYPE .

#LEFT EDITOR 21 0 1

Computes the number of characters remaining in the source text line.

#S FORTH 12 1 1

Converts any remaining digits of a 16-bit binary number on the stack to their ASCII equivalents, using the current BASE. The ASCII characters are placed in the output string. At least one digit will be converted if the number is zero.

' FORTH 11 0 1

Places the address of the parameter field of the next word in the current input stream onto the top of the stack. Searches first the CONTEXT vocabulary, then the CURRENT vocabulary, before giving an error message.

'S FORTH 10 0 1

Places the address of the top of the stack on the stack, i.e., the address of the top of the stack before 'S was invoked.

(FORTH 3 0 0

Begins a comment, which is terminated by) . Comments are ignored by the system and may appear inside or outside a definition. They may not, however, cross an even line boundary in source text screens.

(.) FORTH 12 1 2

Converts a sixteen-bit signed number on top of the stack to its ASCII equivalent, leaving the byte count of the string on the top of the stack and its address beneath for TYPE .

Used by . (i.e., dot).

(MARK) FORTH 9 1 0

Compiles a backward jump in a logical structure.

(MATCH) FORTH 22 4 2

Usage: string-A count string-B count (MATCH)
Counts must be <256. Searches for the 1st occurrence of A in B. Returns the end byte plus 1 of the matched string in B and a truth value: zero if no match and non-zero if match.

(MATCH) EDITOR 22 4 2

In the EDITOR vocabulary on COSMACs only. Behaves like the FORTH vocabulary (MATCH) .

(MOVE) FORTH 22 3 0

Only exists on 6800s and COSMACs; in the EDITOR vocabulary on COSMACs. Same as MOVE except there is an intermediate move to HERE .

(NUMBER) FORTH 10 1 2

Same as NUMBER except that the ASCII string may begin with a minus sign. Also, if the terminating character is not a space, (NUMBER) will exit with an error message. The top of the stack is either the terminator or garbage.

(THEN) FORTH 9 1 0

Completes a forward jump in a logical structure.

* FORTH 5 2 1

Performs an unsigned multiply of the low-order byte of the top number on the stack with the sixteen-bit number beneath it, leaving a sixteen-bit product.

*/ FORTH 5 3 1

Multiplies the second and third numbers on the stack, then divides by the top number, leaving the quotient on top of the stack. This is an unsigned operation with a twenty-three-bit intermediate result.

*/MOD FORTH 5 3 2

Multiplies the second and third numbers on the stack, then divides by the top number, leaving the quotient on top of the stack and the remainder beneath. This is an unsigned operation with a twenty-three-bit intermediate result.

+ FORTH 0 2 1

Replaces the two numbers on the stack by their sum.

+! FORTH 0 2 0

Increments the sixteen-bit word whose address is on the top of the stack by the amount in the second word of the stack.

+LOOP FORTH 0 1 0

Terminates the range of a DO ... LOOP. Increments the index by an unsigned sixteen-bit number on top of the stack, removing the number. The loop is terminated if the new index equals or exceeds the limit (unsigned compare).

WORD VOCABULARY SCREEN STACK: IN OUT

+LOOP FORTH 9 1 0
Defines the compile-time behavior of +LOOP .

, FORTH 0 1 0
Places the sixteen-bit value on top of the stack into the next dictionary position (at HERE) and advances H by two.

- FORTH 0 2 1
Subtracts the top stack item from the second stack item, leaving the difference on the stack.

-' FORTH 0 0 2
Returns a nonzero value if the next word in the current input stream cannot be found in the dictionary, and 0 if it can be found. If the word is found, the second item on the stack is the address of the word's parameter field.

-DUP FORTH 3 1 2
Reproduces the top of the stack only if it is non-zero.

-MOVE FORTH 22 3 0
Same as MOVE except that the count must be less than 256 and the block of memory is moved in reverse order, beginning at its highest address. (8080s and Z80s only.)

-TRAILING FORTH 13 2 2
Reduces the byte count on the top of the stack by the number of trailing blanks found in the string whose address is the second item on the stack.

. FORTH 12 1 0
Outputs a signed sixteen-bit number from the top of the stack.

.R FORTH 13 2 0
Outputs the second number on the stack, right-adjusted in a field whose width is specified on the top of the stack.

/ FORTH 5 2 1
Unsigned division of the second word (full sixteen bits) of the stack by the top (max value 128), leaving the quotient on the top of the stack.

/MOD FORTH 5 2 2
Performs an unsigned division of the second stack item by the first, leaving a quotient on the top of the stack and a remainder beneath.

0< FORTH 0 1 1
If the top stack item is less then zero, replaces it with one; leaves zero otherwise.

0= FORTH 0 1 1
If the top stack item equals zero, replaces it with one; leaves zero otherwise.

1+ FORTH 0 1 1

Adds one to the top stack item.

1LINE EDITOR 21 0 1

Given a string in PAD, searches for the string in the current line. Leaves zero if the string is not found and one if it is. Leaves the cursor positioned at the end of the matched string or at the end of line if not found.

2* FORTH 0 1 1

Doubles the value of the top item on the stack.

2+ FORTH 0 1 1

Adds two to the top stack item.

8* FORTH 3 1 1

Multiplies the top value on the stack by eight.

: FORTH 0 0 0

Creates a dictionary entry for the word following : . Puts the interpreter into compile mode.

; FORTH 0 0 0

Terminates a : definition. Toggles the user variable STATE .

;CODE FORTH 4 0 0

Ends the creation portion of a new defining word and begins the code portion (run-time behavior) of it.

;CODE FORTH 0 0 0

When executed, sets the code address of the new wod to point to the code that follows ;CODE .

;S FORTH 0 0 0

Ends the loading of any screen in which ;S is executed. Within a definition causes an exit to the next outer definition.

< FORTH 0 2 1

If the second stack item is less than the top, replaces the top two items on the stack with one, zero otherwise. This is a limited signed compare. Equivalent to - 0<.

<# FORTH 12 0 0

Begins pictured numeric output. Sets HLD to PAD. sixteen-bit binary number must be on the stack.

<BUILDS FORTH 3 0 0

Begins the compile-time behavior of a new "high-level" defining word. Defined as 0 CONSTANT ; used with DOES>.

WORD VOCABULARY SCREEN STACK: IN OUT

<R FORTH 0 1 0
Removes the top item on the parameter stack and places it on the top of the return stack.

= FORTH 0 2 1
If the top two stack items are equal, replaces them with one; leaves zero otherwise.

> FORTH 5 2 1
If the second item on the stack is greater that the top item, replaces both with one; leaves zero otherwise. This is a limited signed compare. Equivalent to SWAP - 0< .

? FORTH 12 1 0
Outputs the contents of the word address which is on the top of the stack. Equivalent to @ . (dot).

?STACK FORTH 10 0 0
Checks for stack underflow and overflow and issues an error message if appropriate.

@ FORTH 0 1 1
Replaces the address on the top of the stack by the contents of the two-byte word at that location.

A EDITOR 14 1 0
In the current screen, adds the line of text that follows A AFTER the line number given. Line 15 is lost. The added line remains in PAD.

ABS FORTH 5 1 1
Replaces the top stack item with its absolute value.

AND FORTH 0 2 1
Performs the logical sixteen-bit AND operation on the top two stack items.

ASSEMBLE FORTH 9 0 1
For COSMACs only, a constant which gives the load screen of the ASSEMBLER vocabulary.

ASSEMBLER FORTH 0 0 0
Sets CONTEXT to the ASSEMBLER vocabulary.

AT EDITOR 21 1 1
Calculates the physical address in memory of the current cursor position within the current screen.

B EDITOR 21 0 0
Positions the cursor in front of the string just found. Used in conjunction with F .

WORD VOCABULARY SCREEN STACK: IN OUT

BACKUP DISKING 24 0 0
Copies an entire diskette from Drive 0 to Drive 1.

BASE FORTH 0 0 1
A user variable that contains the radix for number conversions on input or output. It is one byte long and is used with C@ and C! .

BEGIN FORTH 9 0 1
Marks the beginning of an indefinite loop which is terminated by END . Leaves its address on the stack.

BLANK FORTH 22 2 0
Given an address in the second stack position and the byte count (<256) on top, stores blanks into that region of memory. Also in the EDITOR vocabulary.

BLK FORTH 0 0 1
A user variable that contains the number of the block being interpreted during a LOAD . If BLK contains zero, input is from the terminal. Overlaps the user variable IN .

BLOCK FORTH 3 1 1
Replaces the block number on the top of the stack by the starting address of its block buffer in memory, adding in OFFSET .

BUFFER FORTH 0 0 1
Returns the address of the block ID of a free block buffer. The ID resides two bytes before the beginning of the block buffer.

C EDITOR 21 0 0
Inserts the string that follows C into the current line, beginning at the current cursor position. Extra characters (at the end of the line) will be lost.

C! FORTH 0 2 0
Stores the eight-bit value in the low-order byte of the second item on the stack into the address on the top of the stack.

C# EDITOR 21 0 1
Calculates the character position of the cursor in the current line.

C, FORTH 0 1 0
Places the low-order byte of the top of the stack into the next dictionary position at HERE and advances H by one.

C@ FORTH 0 1 1
Replaces the address on the top of the stack with its contents. The high-order byte is zero filled.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT	
CODE	FORTH	4	0	0	Begins a dictionary entry for the word following it and enters the ASSEMBLER vocabulary.
COMPILE	FORTH	0	0	0	Changes the user variable STATE ; used by : and ; . (Changes the name field in the dictionary entry. The byte changed is machine-dependent.)
CONSTANT	FORTH	0	1	0	A defining word which creates a dictionary entry for a sixteen-bit value. When the name is invoked, the value is placed on the top of the stack.
CONTEXT	FORTH	0	0	1	A user variable whose contents point to the vocabulary in which searches begin.
COPY	EDITOR	14	2	0	Copies one screen to another. The source screen is unchanged. Usage: source-screen destination-screen COPY
COUNT	FORTH	15	1	2	Takes the address of a character string whose first byte is a character count and replaces it with a character count on top of the stack and the address of the first character beneath. In Screen 16 on COSMACs.
CR	FORTH	12	0	0	Sends a carriage return and line feed to the terminal.
CREATE	FORTH	0	0	0	When executed, creates a dictionary header for the word that follows it. Used in the definition of all defining words.
CROSS	FORTH	19	0	1	A CONSTANT that places the load screen number of the cross-compiler on the top of the stack.
CURRENT	FORTH	0	0	1	A user variable whose contents point to the vocabulary in which new definitions are added. The CURRENT vocabulary is searched when the search of the CONTEXT vocabulary ends.
CVARIABLE	FORTH	4	1	0	A defining word which creates a dictionary entry for an eight-bit value. When the CVARIABLE name is invoked, the address of the value is placed on the top of the stack.
CZ	FORTH	0	0	1	Places one byte of zero on the stack. Increments the stack pointer by one byte.

WORD VOCABULARY SCREEN STACK: IN OUT

D EDITOR 14 1 0

In the current screen, deletes the line specified on the top of the stack and places it in PAD. Succeeding lines are moved up; Line 15 is duplicated.

DECIMAL FORTH 5 0 0

Sets BASE to radix ten for number conversion.

DEFINITIONS FORTH 11 0 0

Sets CURRENT to CONTEXT. Used to specify the vocabulary in which definitions will be entered.

DELETE EDITOR 14 1 0

Stores zero into the first two bytes of the specified screen to mark the screen as unused. This screen then will not be listed by INDEX , SHOW , or TRIAD in the PRINTING utility.

DEVICE PRINTER 17 0 0

Marks the load point for the PRINTER vocabulary. (Not available on COSMACs.)

DISKING FORTH 19 0 1

A CONSTANT that gives the load screen number of the DISKING utility.

DO FORTH 9 0 0

Defines the compile-time behavior of DO .

DO FORTH 0 2 0

Begins a finite loop whose index (the top stack item) and limit (the second stack item) are moved to the return stack when it is invoked.

DOES > FORTH 0 0 0

A defining word which marks the beginning of the run-time portion of a new defining word. Used with <BUILDS .

DOWN DISKING 24 2 0

See RIGHT .

DR0 FORTH 19 0 0

Sets the user variable OFFSET to zero for absolute access by BLOCK and LIST .

DR1 FORTH 19 0 0

Sets the user variable OFFSET to 2000 for relative access to Drive 1 by BLOCK and LIST .

DROP FORTH 0 1 0

Removes the top item from the stack.

WORD VOCABULARY SCREEN STACK: IN OUT

DUMP FORTH 13 2 0

Outputs the values contained in a specified region of memory. Usage: start-addr count DUMP

DUP FORTH 0 1 2

Duplicates the top of the stack.

ECHO FORTH 15 1 0

Sends the character in the low-order byte of the top stack item to the terminal.

ECHO PRINTER 17 1 0

Sends the character in the low-order byte of the top stack item to the printer device. (Not available on COSMACs.)

EDIT FORTH 19 0 1

A constant that is the load screen number of the EDITOR vocabulary. For COSMACs only.

EDITOR FORTH 14 0 0

Sets CONTEXT to the EDITOR vocabulary. It is IMMEDIATE so that it may be invoked inside a definition.

ELSE FORTH 0 1 1

Used within the IF ... THEN structure, ELSE begins the "false" part. The words that follow ELSE are executed if the top stack item was zero (false) when IF was invoked.

ELSE FORTH 9 0 0

Defines the compile-time behavior of ELSE .

END FORTH 0 1 0

Terminates an indefinite loop started with BEGIN . Returns to the start of the loop if the top stack item is zero (false); terminates the loop if the top stack item is non-zero (true). (Not available on 6800s.)

END FORTH 9 0 0

Defines the compile-time behavior of END .

ERASE FORTH 4 2 0

Given the byte count on top of the stack and the address beneath, stores zeros in a region of memory. Usage: start-adr. count ERASE

ERASE-CORE FORTH 3 0 0

Stores zeros in all the block buffers. Does not write to disk any block buffers marked for writing.

ERR EDITOR 21 1 0

Uses the error condition code on top of the stack; if true, moves text from PAD to HERE and invokes 0 QUESTION .

WORD VOCABULARY SCREEN STACK: IN OUT

ERROR DISKING 26 0 1
Leaves the value of STATUS masked for error bits.

EXECUTE FORTH 0 1 0
Executes the word whose parameter field address is on top of the stack.

EXPECT FORTH 16 2 0
Inputs, from the terminal, the number of characters specified on top of the stack and places them into memory at the address given beneath, followed by 2 nulls. The string is ended when the count is exhausted or by a carriage return.

F EDITOR 21 0 0
Beginning at the current cursor position in the current screen, searches for the string that follows F and leaves the cursor positioned immediately after that string.
Multiple lines are searched.

FILL DISKING 24 0 0
Sets a non-zero value into the block IDs of the disk block buffers. Used to force the operating system to read disk Block 0 from disk.

FIND EDITOR 21 0 0
Searches each line of the current screen, beginning at the current cursor position for the string in PAD . Prints an error message if the string is not found.

FLUSH FORTH 3 0 0
Forces all updated blocks to be written to disk.

FMT DISKING 24 0 0
Formats the disk on Drive 1 (where appropriate).

FORGET FORTH 11 0 0
Physically forgets, at execute time, all dictionary entries after and including the word specified in the current input stream.

FORTH FORTH 11 0 0
The name of the innermost vocabulary. Sets CONTEXT to FORTH . It is IMMEDIATE so that it may be invoked inside a definition.

GAP EDITOR 14 1 1
In the current screen, pushes all lines that occur AFTER the specified line down one.

H FORTH 0 0 1
A user variable that contains the address of the top of the dictionary. See HERE .

WORD VOCABULARY SCREEN STACK: IN OUT

HERE FORTH 0 0 1
Places on the stack the address of the next available byte at the top of the dictionary. See H .

HEX FORTH 5 0 0
Sets BASE to radix sixteen for number conversion.

HLD FORTH 12 0 1
A variable that points at the most recent character of the output string during pictured numeric output.

HOLD EDITOR 14 1 0
Transfers the line whose number is on the top of the stack to PAD.

HOLD FORTH 12 1 0
Decrements HLD and places an ASCII character that is on top of the stack into the output string during pictured numeric output. See <# , # and #> .

I EDITOR 14 1 0
In the current screen, inserts the line that is stored in PAD into the line that follows the one whose number is on top of the stack. Succeeding lines are pushed down; Line 15 is lost.

I FORTH 0 0 1
Copies the top of the return stack onto the parameter stack; it does not alter the return stack.

IF FORTH 0 1 0
Begins a conditional structure. Executes the words that immediately follow IF when the top of the stack is true (non-zero); otherwise skips to ELSE (if present) or THEN (if there is no ELSE) or WHILE (instead of THEN).

IF FORTH 9 0 1
Defines the compile-time behavior of IF .

IMMEDIATE FORTH 3 0 0
Marks the word most recently defined as a compiling word. The word is executed when encountered inside of a definition.

IN FORTH 0 0 1
A user variable that points to the relative location in the input stream. IN overlaps the user variable BLK .

IN-LINE FORTH 11 1 0
Given a number on the top of the stack, compiles it as a sixteen-bit literal.

WORD VOCABULARY SCREEN STACK: IN OUT

IN-LINE FORTH 0 0 1
Puts a sixteen-bit literal on the stack at run time.

INC DISKING 24 0 1
A constant that gives the block increment for RIGHT and SWEEP . Must be an odd number.

INDEX PRINTING 27 2 0
Types the first line of each screen in the range given, sixty lines to a page. The copyright and heading are at the base of each page.
Usage: start-screen# end-screen# INDEX .

INTERPRET FORTH 0 0 0
Outer interpreter loop; scans and searches for a word (to be compiled or executed, depending on STATE and precedence) in the dictionary. If not found, converts number and comiles literal form if in compile mode.

J FORTH 4 0 1
Puts the index of the outer of two nested DO ... LOOPS on the stack. Only the indices of the two innermost nested loops are available. In Screen 5 on COSMACs.

KEY FORTH 16 0 1
Receives and places on the stack a single character from the keyboard. In Screen 15 on COSMACs.

L FORTH 13 0 0
Lists the screen specified in the user variable SCR.

L# EDITOR 21 0 1
Calculates the line number of the cursor in the current screen. Implementation is machine-dependent.

LEAVE FORTH 4 0 0
Sets the limit of a DO ... LOOP equal to zero so that a loop will be terminated. Implementation is machine-dependent.
In Screen 5 on COSMACs.

LEFT DISKING 24 2 0
See RIGHT .

LF PRINTING 27 0 0
Sends one line feed.

LINE EDITOR 14 1 2
Given the number of a line in the current screen on the top of the stack, returns a character count of sixty-four (on top) and the address of the line beneath. The line number is masked by fifteen.

WORD VOCABULARY SCREEN STACK: IN OUT

LINE FORTH 13 2 2
Given a line number beneath and a screen number on top of the stack, calculates the block address with a count of 64 on the top of the stack. Can be used by TYPE or MOVE .

LIST FORTH 13 1 0
Lists the screen whose number is found on the top of the stack and places the screen number in SCR .

LOAD FORTH 3 1 0
Begins interpretation of source text in the screen whose number is on the top of the stack.

LOG DISKING 26 1 0
Logs a disk error by typing the block number that is on top of the stack, followed by the disk error message and the error status.

LOOP FORTH 0 0 0
Terminates the range of a DO ... LOOP. Increments the index by one and exits if the index equals or exceeds the limit.

LOOP FORTH 9 1 0
Defines the compile-time behavior of LOOP .

M EDITOR 21 1 0
Given a count, moves the cursor forward (positive) or backward (negative). The line that contains the cursor is sent to the terminal.

M* FORTH 5 2 2
Multiplies the top two values on the stack, leaving a twenty-four-bit product. The output format is chip-dependent. See M/MOD .

M/MOD FORTH 5 3 2
Divides a twenty-four-bit number by the top stack item, leaving the remainder on top and the dividend beneath.
The input format is chip-dependent. See also M* .

MATCH DISKING 25 2 0
Usage: start-screen# end-screen#-plus-1 MATCH
Compares between DR0 and DR1; does not match screens if both begin with 0. On the first mismatch, types screen# and approximate line# (relative block * 2) of the mismatch.

MAX FORTH 5 2 1
A limited signed compare between the top two values on the stack that leaves the largest value on the stack.

MESSAGE FORTH 10 1 0
Types on the terminal a specified line relative to the start of Scr. 23. Omits trailing blanks. Uses Scr. 23 as the logical base, i.e., Message 16 is Line 0 of Scr. 24, Message 32 is Line 0 of Scr. 25, etc.

WORD VOCABULARY SCREEN STACK: IN OUT

MESSAGE PRINTER 17 1 0

Same as MESSAGE in the FORTH vocabulary. (Not available on COSMACs.

MIN FORTH 5 2 1

A limited signed compare between the top two values on the stack that leaves the smaller value on the stack.

MINUS FORTH 0 1 1

Replaces the top of the stack by its two's complement.

MOD FORTH 5 2 1

Divides the top stack item into the value beneath it, leaving the remainder on the top of the stack.

MOVE FORTH 0 3 0

Moves a specified region of memory to another region of memory; moves the locations with lower addresses first.
The source area remains unchanged.
Usage: source-addr. dest.-addr. byte-count MOVE

MSG FORTH 15 0 0

Defines a word that will type out the string that follows it in the dictionary. The string is preceeded by a character count. In Screen 16 on COSMACs.

MSG PRINTER 17 0 0

Sets ASCII character codes into a named definition in the dictionary. (Not available on COSMACs.)

N EDITOR 21 0 0

Finds the next occurrence of a string (found with an F) in the current screen.

NB DISKING 24 0 1

A constant that gives the number of block buffers.

NEW DISKING 24 0 1

A constant that gives the first block number on Drive 1.

NOT FORTH 5 1 1

Reverses the truth value of the top of the stack.
Identical to 0= .

NOTIFY DISKING 26 1 1

Erases the block ID in the buffer whose address is on top of the stack after first fetching the block number contained in the ID. Invokes LOG with the block number and returns the number less the contents of OFFSET to the stack.

NUMBER FORTH 0 1 2

Given the starting address less 1 of a numeric ASCII string on the stack, converts the string to binary according to the current value of BASE and leaves it in the second stack entry. The top item points to the non-numeric terminator.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
OCTAL	FORTH	5	0	0

Sets BASE to radix eight for number conversion.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
OFFSET	FORTH	3	0	1

A user variable whose contents are added to block numbers in BLOCK to determine the physical block number.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
OK	FORTH	15	0	0

Types the characters O, K, carriage return, and line feed. In Screen 16 on COSMACs.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
OVER	FORTH	0	2	3

Copies the second item on the stack onto the top.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
P	EDITOR	14	1	0

Places the line of text that follows P into the specified line. The previous content of the line is lost. The "put" line remains in PAD.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
PAD	FORTH	12	0	1

The starting address of a holding buffer, PAD resides sixty-five bytes above HERE and moves as definitions are added to and deleted from the dictionary.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
PRINTER	FORTH	19	0	1

A constant that places the load screen number of the PRINTER utility on the stack. (Not available on COSMACs.)

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
PRINTER	PRINTER	17	0	0

Same as CR . (Not available on COSMACs.)

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
PRINTING	FORTH	19	0	1

A constant that places the load screen number of the PRINTING utility on the stack.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
QUESTION	FORTH	10	1	0

Repeats the last word executed by the text interpreter (found at HERE) and issues an error message as specified by MESSAGE , then empties both stacks and returns control to the operator. No OK is issued.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
QUIT	FORTH	16	0	0

Empties the return stack and returns control to the operator. No OK is issued.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
R	EDITOR	14	1	0

Replaces the line specified on the top of the stack with the contents of PAD.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
R	FORTH	4	0	1

A constant that gives the address of the return stack pointer. For COSMACs, in the ASSEMBLER vocabulary.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
R!	FORTH	16	1	0

Moves the contents of Register U to Register R (i.e., resets the return stack). On COSMACs only.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
R#	FORTH	13	0	1

User variable which contains the character position of the cursor in the EDITOR. When file management is in the system R# is the record number of the currently accessed record.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
R>	FORTH	0	0	1

Removes the top of the return stack and places it on the parameter stack.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
REMOVE	EDITOR	21	1	0

Given the character position of the beginning of the string to be deleted, deletes those characters on the line (up to the current cursor position) and moves all characters up.
Trailing blanks are added at the end as needed.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
RIGHT	DISKING	24	2	0

Copies the range of screen given from Drive 0 to Drive 1.
Usage: start-screen# end-screen#-plus-1 RIGHT
May be called UP , DOWN , or LEFT .

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
ROT	FORTH	0	3	3

Rotates the top three stack items, putting the third stack item on the top. On 6800s ROT resides in Screen 5.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
S!	FORTH	10	1	0

Sets the address of the current stack pointer to the one given on the stack.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
S0	FORTH	0	0	1

A user VARIABLE that contains the address of the bottom of the parameter stack and the start of the input message buffer.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
SCR	FORTH	13	0	1

A user variable that holds the current EDITOR screen number.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
SHOW	PRINTING	27	2	0

Types TRIADs of screens in the inclusive range given.
Usage: start-screen end-screen SHOW

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
SIGN	FORTH	12	2	1

Places a minus sign in the pictured numeric output string if the second word on the stack is negative. Deletes this second word on the stack but retains the top word.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
SPACE	FORTH	12	0	0

Sends a single space (blank) to the terminal.

WORD VOCABULARY SCREEN STACK: IN OUT

SPACES FORTH 12 1 0
Sends the number of spaces that is designated by the top stack item. May send zero spaces.

STATE FORTH 0 0 1
A user variable, one byte wide, that indicates whether the interpreter is in compile or execute mode.

STATUS DISKING 26 0 1
Returns on the stack the disk status as of the last operator.

STRING EDITOR 21 0 0
Scans characters in the input stream until the delimiting character (the low-order byte on top of the stack or a carriage return) is encountered. Reads characters from the terminal into PAD with a leading count.

SWAP FORTH 0 2 2
Exchanges the top two stack items.

SWEEP DISKING 24 2 0
Reads each screen in the range given to check for disk errors.
Usage: start-screen# end-screen#-plus-1 SWEEP

T EDITOR 14 1 1
Types the line specified (on the top of the stack) of the current screen and transfers it to PAD. The line number is left on the stack.

TASK FORTH 3 0 0
Marks the beginning of the application vocabulary.

TEXT FORTH 13 1 0
Scans characters in the input stream until delimiter (low-order byte as top stack item or carriage return) is encountered. Leading occurrences of the delimiter are skipped over. Input is placed in PAD and is blank filled.

THEN FORTH 0 0 0
Marks the end of an IF ... THEN structure.

THEN FORTH 9 0 0
Defines the compile-time behavior of THEN .

TILL EDITOR 21 0 0
Beginning at the current cursor position on the current line, deletes all characters up to and including the string that follows TILL .

TOP EDITOR 14 0 0
Positions the cursor at the beginning of the current screen.

WORD VOCABULARY SCREEN STACK: IN OUT

TRIAD PRINTING 27 1 0

Types a set of three screens, given one screen number. The screen number may be any of the three screens on a page; the top screen is always the screen number modulo three. Copyright and heading appear at page bottom.

TYPE FORTH 15 2 0

Uses a character count on top of the stack and an address beneath to send characters to the terminal. May TYPE zero characters. In Screen 16 on COSMACs.

TYPE PRINTER 17 2 0

Uses a character count on top of the stack and an address beneath to send characters to the printer device.
(Not available on COSMACs.)

U FORTH 4 0 1

A constant that gives the address of the pointer to the start of the user area. For COSMACs, in ASSEMBLER vocabulary.

U* FORTH 0 2 1

Unsigned multiply of the low-order bytes of the top two words on the stack, leaving a sixteen-bit product.

U/ FORTH 0 2 2

Unsigned divide of the second word on the stack by the top word, leaving a quotient on top and a remainder beneath.

UP DISKING 24 2 0

See RIGHT .

UPDATE FORTH 0 0 0

Marks the last buffer returned by BLOCK for writing. The block is rewritten on the disk either by the next FLUSH or automatically when the buffer is needed for another block.

USER FORTH 0 1 0

A defining word, used to name locations at fixed relative addresses within the user area.

VARIABLE FORTH 4 1 0

A defining word that creates a dictionary entry for a sixteen-bit value. When the VARIABLE name is invoked, the address of the value is placed on top of the stack.

VOCABULARY FORTH 11 0 0

Defines a word whose parameter field plus two points to the most recent entry of that vocabulary's set of definitions.
Executing a vocabulary name points CONTEXT to that vocabulary's parameter field plus two.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
WHILE	FORTH	0	0	0

Terminates an indefinite loop of the following form:
BEGIN (condition) IF WHILE or BEGIN (condition) IF ELSE WHILE
Allows a test at the beginning of an indefinite loop.
(Not available on 6800s.)

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
WHILE	FORTH	9	2	0

Defines the compile-time behavior of WHILE .

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
WORD	FORTH	0	1	0

Reads forward in the current input stream until the delimiter given on the stack. The byte count and text are stored at HERE with the byte count in the first byte.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
X	EDITOR	21	0	0

Beginning at the current cursor position, searches for and deletes the string that follows X . Multiple lines are searched.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
[	FORTH	13	0	0

Defines the run-time behavior of [, which types out text on the CRT. The string resides in the dictionary, preceded by a count. It was laid down at compile time by use of the compiling word [.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
[	FORTH	13	0	0

A compiling word which causes the string of characters until the delimiter], following it to be typed when the defined word is invoked.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
[']	FORTH	0	0	1

During compilation, pushes onto the stack the sixteen-bit value that follows it.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
[']	FORTH	11	0	0

Defines the compile-time behavior of ['] .

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
[BLOCK]	DISKING	26	1	1

Invokes BLOCK and, in case of read errors, retries up to ten times. Invokes LOG for all but the last retry.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
[SWAP]	FORTH	11	1	1

A compiling word which swaps the top two words of the stack during compilation.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
\	FORTH	0	0	0

A compiling word that places the address of the word that follows it into a new definition. Used to help define the run-time and compile-time behavior of a compiler word.

WORD	VOCABULARY	SCREEN	STACK: IN	OUT
eot	FORTH	0	0	0

An ASCII null character that terminates scanning in the current input stream. Null controls the sequencing of the block buffers of a screen.