Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
doLIST	(a --)	DOCOL	COG Word		X	X	X				$10		Process colon list.
doDOIT	(a --)	DODOIT	COG Word		X	X					$23		Do what DOES> does
doVAR	(-- a)	DOVAR	COG Word		X	X	X				$5		Run time routine for VARIABLE and CREATE.
doCONST	(-- w)	DOCONST	COG Word		X	X					$1A		Run time routine for CONSTANT
doUSER	(-- a)	DOUVAR	COG Word		X	X					$2		Run time routine for user variables.
doLITB	(-- c)	DOLIT_B	COG Word		X	X				X	$2B		Push an inline byte literal.
doLIT	(-- w)	DOLIT_W	COG Word		X	X	X				$30		Push an inline word literal.
doLITL	(-- l)	DOLIT_L	COG Word		X	X				X	$2D		Push an inline long literal.
EXECUTE	(ca --)	EXECUTE	COG Word			X	X				$36		Execute the word at ca.
next	(--)	LNEXT	COG Word		X	X	X				$41		Run time code for the single index loop.
?branch	(f --)	QBRANCH	COG Word		X	X	X				$4B		Branch if flag is zero.
branch	(--)	BRANCH	COG Word		X	X	X				$51		Branch to an inline address.
EXIT	(--)	EXIT	COG Word			X	X				$53		Terminate a colon definition.
SP0	(-- a)	CLEARD	COG Word			X					$5B		Pointer to bottom of the data stack.
RP0	(-- a)	CLEARR	COG Word			X					$5D		Pointer to bottom of the return stack.
DUP	(w -- w w)	DUP	COG Word			X	X				$66		Duplicate the top stack item.
?DUP	(w -- w w | 0)	QDUP	COG Word			X	X				$5F		Dup tos if its is not zero.
DROP	(w --)	DROP	COG Word			X	X		X		$70	40	Discard top stack item.
SWAP	(w1 w2 -- w2 w1)	SWAP	COG Word			X	X		X		$71	64	Exchange top two stack items.
OVER	(w1 w2 -- w1 w2 w1)	OVER	COG Word			X	X		X		$7F	64	Copy second stack item to top.
SP@	(-- a)	SPFETCH	COG Word			X	X		X		$86		Push the current data stack pointer.
SP!	(a --)	SPSTORE	COG Word			X	X				$8D		Set the data stack pointer.
RP!	(a --)	RPSTORE	COG Word		X	X	X				$91		Set the return stack pointer.
RP@	(-- a)	RPFETCH	COG Word			X	X				$95		Push the current RP to the data stack.
R>	(-- w)	RPSP	COG Word			X	X				$9A		Pop the return stack to the data stack.
R@	(-- w)	CPYRPSP	COG Word			X	X				$A0		Copy top of return stack to the data stack.
I	(-- w)	CPYRPSP	COG Word			X					$A0		Alias of R@
>R	(w --)	SPRP	COG Word		X	X	X				$A9		Push the data stack to the return stack.
COG!	(w cog --)	COGSTORE	COG Word			X				X	$AF		Pop the data stack to COG ram memory.
2!	(w a --)	LSTORE	COG Word			X	X				$B5		Pop the data stack to long memory.
C!	(c b --)	CSTORE	COG Word			X	X				$B7		Pop the data stack to byte memory.
!	(w a --)	STORE	COG Word			X	X				$B9	72	Pop the data stack to memory.
COG@	(cog -- w)	COGFETCH	COG Word			X				X	$C0		Push COG ram memory location to the data stack.
2@	(a -- w)	LFETCH	COG Word			X	X				$C8		Push long memory location to the data stack.
C@	(b -- c)	CFETCH	COG Word			X	X				$CA		Push byte memory location to the data stack.
@	(a -- w)	FETCH	COG Word			X	X				$CC	48	Push word memory location to the data stack.
1-	(w -- w)	ONESUB	COG Word			X					$D5		Decrements TOS
1+	(w -- w)	ONEADD	COG Word			X					$D9	40	Increments TOS
2-	(w -- w)	TWOSUB	COG Word			X					$F2	40	Subtract two from TOS

Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
2+	(w -- w)	TWOADD	COG Word			X					$F4	40	Add two to TOS
2*	(w -- w)	TWOMUL	COG Word			X					$DD	40	Multiply TOS by two
2/	(w -- w)	TWODIV	COG Word			X					$E1	40	Divide TOS by two
CELL+	(a -- a)	CELLADD	COG Word			X	X				$F4		Add cell size in byte to address.
CELL-	(a -- a)	CELLSUB	COG Word			X	X				$F2		Subtract cell size in byte from address.
2	(-- n)	TWO	COG Word			X					$FF	56	Push two to TOS
1	(-- n)	ONE	COG Word			X					$101	56	Push one to TOS
0	(-- n)	ZERO	COG Word			X					$103	40	Push zero to TOS
FALSE	(-- F)	ZERO	COG Word			X				X	$101		Push zero to TOS
-1	(-- n)	XTRUE	COG Word			X	X				$F9		Push -1 to TOS
TRUE	(-- T)	XTRUE	COG Word			X				X	$F9		Push -1 to TOS
+	(w w -- w)	ADD	COG Word			X	X				$109	72	Add top two items.
0<	(n -- t)	TESTNEG	COG Word			X	X				$123		Return true if n is negative.
=	(w w -- t)	TESTEQ	COG Word			X	X				$111		Return true if top two are equal.
<	(n1 n2 -- t)	TESTLT	COG Word			X	X				$11A		Signed compare of top two items.
ABS	(n -- u)	ABS	COG Word			X	X				$12E	72	Absolute Value from n
NEGATE	(n -- -n)	INVERT	COG Word			X	X	X			$138	72	Two's complement of tos.
OR	(w w -- w)	OR	COG Word			X	X	X			$141	64	Bitwise OR.
AND	(w w -- w)	AND	COG Word			X	X	X			$143	64	Bitwise AND.
XOR	(w w -- w)	XOR	COG Word			X	X	X			$147	48	Bitwise XOR.
NOT	(w -- w)	NOT	COG Word			X		X			$145	48	Logical Not of TOS
ROR	(w n -- w)	ROR	COG Word			X				X	$F0	64	Logically rotate w by n bits right.
ROL	(w n -- w)	ROL	COG Word			X				X	$E9	64	Logically rotate w by n bits left.
SHL	(w n -- w)	SHL	COG Word			X				X	$E5	64	Logically shift w by n bits left.
SHR	(w n -- w)	SHR	COG Word			X				X	$E7	64	Logically shift w by n bits right.
/MOD	(n n -- r q)	DIV	COG Word			X	X				$151		Signed divide. Return mod and quotient.
*	(n1 n2 --nt)	MUL	COG Word			X	X				$167		Signed Multiply.
>DIRA	(n --)	_DIRA	COG Word			X				X	$186	56	Put TOS to direction states for P0- P31
DIRA>	(-- n)	_DIRAG	COG Word			X				X	C$	64	Push direction states for P0- P31 to TOS
INA	(-- n)	_INA	COG Word			X				X	$18E	48	Push input states for P0-P31 to TOS
OUTA	(n --)	_OUTA	COG Word			X				X	$188	48	Put TOS to output states for P0- P31
SETTICKS	(--)	SETTICKS	COG Word			X				X	$178		Store CNT to internal register TIME
GETTICKS	(-- n)	GETTICKS	COG Word			X				X	$17A		Push the result of subtract internal register TIME from CNT
WAITTICKS	(n --)	WAITTICKS	COG Word			X				X	$194		Wait for n ticks.
HUPOP	(n1 n2 -- n)	HUPOP	COG Word			X				X	$19A		Perform a hub operation.
NOP	(--)	JNEXT	COG Word			X	X				$C	32	Do nothing.
GETCOGID	(-- n)	GETCOGID	COG Word			X				X	$181		Get the COG number.
RXHEAD	(-- n)		Constant							X			
RXTAIL	(-- n)		Constant							X			

Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
TXHEAD	(-- n)		Constant							X			
TXTAIL	(-- n)		Constant							X			
RXBUFF	(-- a)		Constant							X			
TXBUFF	(-- a)		Constant							X			
VERSION	(-- n)		Constant							X			
PAR	(-- a)		Constant							X	$1F0		Boot Parameter
CNT	(-- a)		Constant							X	$1F1		System Counter
CTRA	(-- a)		Constant							X	$1F8		Counter A Control
CTRB	(-- a)		Constant							X	$1F9		Counter B Control
FRQA	(-- a)		Constant							X	$1FA		Counter A Frequency
FRQB	(-- a)		Constant							X	$1FB		Counter B Frequency
PHSA	(-- a)		Constant							X	$1FC		Counter A Phase
PHSB	(-- a)		Constant							X	$1FD		Counter B Phase
VCFG	(-- a)		Constant							X	$1FE		Video Configuration
VSCL	(-- a)		Constant							X	$1FF		Video Scale
COGSTAT	(-- a)		Multiprocessing							X			User variable for internal use.
BASE	(-- a)		Variable				X						Storage of the radix base for numeric I/O.
tmp	(-- a)		Variable				X						A temporary storage location used in parse and find.
>IN	(-- a)		Variable				X						Hold the character pointer while parsing input stream.
#TIB	(-- a)		Variable				X						Hold the current count and address of the terminal nput buffer.
EVAL	(-- a)		Variable				X						Execution vector of EVAL.
HLD	(-- a)		Variable				X						Hold a pointer in building a numeric output string.
CONTEXT	(-- a)		Variable				X						A area to specify vocabulary search order.
CP	(-- a)		Variable				X						Point to the top of the code dictionary.
LAST	(-- a)		Variable				X						Point to the last name in the name dictionary.
CEND	(-- a)		Variable							X			Hold the end of a code definition. (SEE)
RND	(-- a)		Variable							X			Hold the Random Seed
COG3	(-- a)		Multiprocessing							X			Multiprocessing variable for COG 3
COG4	(-- a)		Multiprocessing							X			Multiprocessing variable for COG 4
COG5	(-- a)		Multiprocessing							X			Multiprocessing variable for COG 5
COG6	(-- a)		Multiprocessing							X			Multiprocessing variable for COG 6
COG7	(-- a)		Multiprocessing							X			Multiprocessing variable for COG 7
COGCOMM	(--)		Propeller							X			Adresslist for internal use (SEE word)
RESET	(--)		Propeller							X			Reboot the Propeller Chip
CLK	(-- l)		Propeller							X			Current system clock frequency in Hz.
CLKREG>	(-- n)		Propeller							X			Push the current state from CLK register to TOS
>CLKREG	(n --)		Propeller							X			Put TOS to the CLK register.
I+	(n -- w)		Stack										Add n to the number on top of the return stack.
I-	(n -- w)		Stack										Subtract the number on top of the return stack from n.

Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
TUCK	((w1 w2 -- w2 w1 w2)		Stack										Copy TOS to third position.
NIP	(w1 w2 -- w2)		Stack										Remove second item from stack.
ROT	(w1 w2 w3 -- w2 w3 w1)		Stack				X						Rot 3rd item to top.
2DROP	(w w --)		Stack				X						Discard two items on stack.
2DUP	(w1 w2 -- w1 w2 w1 w2)		Stack				X						Duplicate top two items.
-	(n1 n2 -- n)		Math				X						Subtract two items. n= n1-n2
EMIT	(c --)		Terminal				X						Send character c to the output device.
?KEY	(-- c T | F)		Terminal				X						Return input character and true, or a false
U<	(u u -- t)		Compare				X						Unsigned compare of top two items.
MAX	(n n -- n)		Math				X						Return the greater of two top stack items.
MIN	(n n -- n)		Math				X						Return the smaller of top two stack items.
WITHIN	(u ul uh -- t)		Math				X						Return true if u is within the range of ul and uh. (ul <= u < uh)
ALIGNED	(b -- a)		Global Memory				X						Align address to the cell boundary.
MOD	(n n -- r)		Math				X						Signed divide. Return mod only.
CELLS	(n -- n)		Global Memory				X						Multiply tos by cell size in bytes.
BL	(-- 32)		Text I/O				X						Return 32, the blank character.
>CHAR	(c -- c)		Text I/O				X						Filter non-printing characters.
DEPTH	(-- n)		Stack					X					Return the depth of the data stack.
PICK	(... +n -- ... w)		Stack					X					Copy the nth stack item to tos.
+!	(n a --)		Global Memory				X						Add n to the contents at address a.
COUNT	(b -- b +n)		Global Memory				X						Return count byte of a string and add 1 to byte address.
HERE	(-- a)		Global Memory				X						Return the top of the code dictionary.
PAD	(-- a)		Global Memory				X						Return the address of the text buffer above the code dictionary.
TIB	(-- a)		Global Memory				X						Return the address of the terminal input buffer.
@EXECUTE	(a --)		Global Memory				X						Execute vector stored in address a.
CMOVE	(b1 b2 u --)		Global Memory				X						Copy u bytes from b1 to b2.
FILL	(b u c --)		Global Memory				X						Fill u bytes of character c to area beginning at b.
BLANK	(-- 32)		Global Memory				X						Return 32, the blank character.
ERASE	(b u --)		Global Memory				X						Erase u bytes beginning at b.
PACK$	(b u a -- a)		Global Memory				X						Build a counted string with u characters from b. Null fill.
DIGIT	(u -- c)		Numeric I/O				X						Convert digit u to a character.
EXTRACT	(n base -- n c)		Numeric I/O				X						Extract the least significant digit from n.
<#	(--)		Numeric I/O				X						Initiate the numeric output process.
HOLD	(c --)		Numeric I/O				X						Insert a character into the numeric output string.
#	(u -- u)		Numeric I/O				X						Extract one digit from u and append the digit to output string.
S#	(u -- 0)		Numeric I/O				X						Convert u until all digits are added to the output string.
SIGN	(n --)		Numeric I/O				X						Add a minus sign to the numeric output string.
#>	(w -- b u)		Numeric I/O				X						Prepare the output string to be TYPE'd.
str	(w -- b u)		Numeric I/O				X						Convert a signed integer to a numeric string.

Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
HEX	(--)		Numeric I/O					X					Use radix 16 as base for numeric conversions.
OCTAL	(--)		Numeric I/O						X				Use radix 8 as base for numeric conversions.
BINARY	(--)		Numeric I/O						X				Use radix 2 as base for numeric conversions.
DECIMAL	(--)		Numeric I/O					X					Use radix 10 as base for numeric conversions.
DIGIT?	(c base -- u t)		Numeric I/O			X							Convert a character to its numeric value. A flag indicates success.
NUMBER?	(a -- n T | a F)		Numeric I/O			X							Convert a number string to integer. Push a flag on tos.
KEY	(-- c)		Text I/O			X							Wait for and return an input character.
NUF?	(-- t)		Text I/O			X							Return false if no input, else pause and if CR return true.
SPACE	(--)		Text I/O			X							Send the blank character to the output device.
SPACES	(n --)		Text I/O			X							Send n spaces to the output device.
TYPE	(b u --)		Text I/O			X							Output u characters from b.
CR	(--)		Text I/O			X							Output a carriage return and a line feed.
do$	(-- a)		Text I/O		X	X							Return the address of a compiled string.
$"|	(-- a)		Text I/O		X	X							Run time routine compiled by $". Return address of a compiled string.
."|	(--)		Text I/O		X	X							Run time routine of ." . Output a compiled string.
.R	(n +n --)		Text I/O			X							Display an integer in a field of n columns, right justified.
U.R	(u +n --)		Text I/O			X							Display an unsigned integer in n column, right justified.
U.	(u --)		Text I/O			X							Display an unsigned integer in free format.
.	(w --)		Text I/O			X							Display an integer in free format, preceeded by a space.
?	(a --)		Text I/O			X							Display the contents in a memory cell.
parse	(b u c -- b u n ; <string>)		Parser			X							Scan string delimited by c. Return found string and its offset.
PARSE	(c -- b u ; <string>)		Parser			X							Scan input stream and return counted string delimited by c.
.(	(--)		Parser			X							Output following string up to next) .
(	(--)		Parser			X							Ignore following string up to next) . A comment.
\	(--)		Parser			X							Ignore following text till the end of line.
WORD	(c -- a ; <string>)		Parser			X							Parse a word from input stream and copy it to code dictionary.
TOKEN	(-- a ; <string>)		Parser			X							Parse a word from input stream and copy it to name dictionary.
NAME>	(na -- ca)		Dictionary			X							Return a code address given a name address.
SAME?	(a a u -- a a f \ -0+)		Dictionary			X							Compare u cells in two strings. Return 0 if identical.
find	(a va -- ca na | a F)		Dictionary			X							Search a vocabulary for a string. Return ca and na if succeeded.
NAME?	(a -- ca na | a F)		Dictionary			X							Search all context vocabularies for a string.
^H			Terminal			X							Backup the cursor by one character.
TAP			Terminal			X							Accept and echo the key stroke and bump the cursor.
kTAP			Terminal			X							Process a key stroke, CR or backspace.
accept	(b u -- b u)		Terminal			X							Accept characters to input buffer. Return with actual count.
QUERY	(--)		Terminal			X							Accept input stream to terminal input buffer.
ABORT	(--)		Terminal			X							Reset data stack and jump to QUIT.
abort"	(f --)		Terminal			X							Run time routine of ABORT" . Abort with a message.

Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
$INTERPRET	(a --)		Interpreter				X						Interpret a word. If failed, try to convert it to an integer.
[	(--)		Interpreter				X						Start the text interpreter.
.OK	(--)		Interpreter				X						Display 'ok' only while interpreting.
?STACK	(--)		Interpreter				X		X				Abort if the data stack underflows.
EVAL	(--)		Interpreter				X						Interpret the input stream.
PRESET	(--)		Interpreter				X						Reset data stack pointer and the terminal input buffer.
QUIT	(-- ca)		Interpreter				X						Reset return stack pointer and start text interpreter.
'	(-- ca)		Compiler				X						Search context vocabularies for the next word in input stream.
ALLOT	(n --)		Compiler				X	X	X				Allocate n bytes to the code dictionary.
,	(w --)		Compiler				X						Compile an integer into the code dictionary.
[COMPILE]	(-- ; <string>)		Compiler	X			X						Compile the next immediate word into code dictionary.
COMPILE	(--)		Compiler		X		X						Compile the next address in colon list to code dictionary.
LITERAL	(w --)		Compiler	X			X						Compile tos to code dictionary as an integer literal.
CHAR	(-- c)		Compiler										Parse next word and return its first character.
[CHAR]	(-- ch | --)		Compiler	X									Compile the first char of the following string.
$,"	(--)		Compiler				X						Compile a literal string up to next ".
FOR	(-- a)		Structures	X			X						Start a FOR-NEXT loop structure in a colon definition.
BEGIN	(-- a)		Structures	X			X						Start an infinite or indefinite loop structure.
NEXT	(a --)		Structures	X			X						Terminate a FOR-NEXT loop structure.
UNTIL	(a --)		Structures	X			X						Terminate a BEGIN-UNTIL indefinite loop structure.
AGAIN	(a --)		Structures	X			X						Terminate a BEGIN-AGAIN infinite loop structure.
IF	(-- A)		Structures	X			X						Begin a conditional branch structure.
AHEAD	(-- A)		Structures	X			X						Compile a forward branch instruction.
REPEAT	(A a --)		Structures	X			X						Terminate a BEGIN-WHILE-REPEAT indefinite loop.
THEN	(A --)		Structures	X			X						Terminate a conditional branch structure.
AFT	(a -- a A)		Structures	X			X						Jump to THEN in a FOR-AFT-THEN-NEXT loop the first time through.
ELSE	(A -- A)		Structures	X			X						Start the false clause in an IF-ELSE-THEN structure.
WHILE	(a -- A a)		Structures	X			X						Conditional branch out of a BEGIN-WHILE-REPEAT loop.
ABORT"	(-- ; <string>)		Structures	X			X						Conditional abort with an error message.
$"	(-- ; <string>)		Structures	X			X						Compile an inline string literal.
."	(-- ; <string>)		Structures	X			X						Compile an inline string literal to be typed out at run time.
?UNIQUE	(a -- a)		Name Compiler				X						Display a warning message if the word already exists.
$,n	(na --)		Name Compiler				X						Build a new dictionary name using the string at na.
$COMPILE	(a --)		Forth Compiler				X						Compile next word to code dictionary as a token or literal.
OVERT	(--)		Forth Compiler				X						Link a new word into the current vocabulary.
;	(--)		Forth Compiler	X	X				X				Terminate a colon definition.
]	(--)		Forth Compiler				X						Start compiling the words in the input stream.
:	(-- ; <string>)		Forth Compiler						X				Start a new colon definition using next word as its name.
IMMEDIATE	(--)		Forth Compiler						X				Make the last compiled word an immediate word.

Word	Stack Comment	Label	Category	Flags: Immediate	Compile only	COG (native)	eForth	Forth-79	Forth-83	FPForth	Address	Ticks	Description
CREATE	(-- ; <string>)		Defining word					X					Compile a new array entry without allocating code space.
VARIABLE	(-- ; <string>)		Defining word					X					Compile a new variable initialized to 0.
2VARIABLE	(-- ; <string>)		Defining word										Compile a new long variable initialized to 0.
CONSTANT	(w -- ; <string>)		Defining word					X					Compile a constant
2CONSTANT	(l -- ; <string>)		Defining word										Compile a long constant
USER	(-- ; <string>)		Defining word						X				Compile a new user variable.
dodoes	(--)		Defining word		X								Compile time Routine for DOES>
DOES>	(-- addr)		Defining word	X			X						Modify latest created word to jump into the DOES part of a defining word with its pfa on the data stack.
_TYPE	(b u --)		Tools			X							Display a string. Filter non-printing characters.
dm+	(a u -- a)		Tools			X							Dump u bytes from , leaving a+u on the stack.
DUMP	(a u --)		Tools			X							Dump u bytes from a, in a formatted manner.
CDUMP	(a u --)		Tools						X				Dump u bytes from a. a is a address in COG RAM
.S	(--)		Tools						X				Display the contents of the data stack.
>COGNAME	(		Tools						X				Run time routine for >NAME. Searchs native words.
>NAME	(ca -- na | F)		Tools						X				Convert code address to a name address.
FORGET	(-- ; <string>)		Dictionary					X					Forget word <string> and everything that follows it.I93
.ID	(na --)		Tools			X							Display the name at address.
SEE	(-- ; <string>)		Tools			X							A simple decompiler.
WORDS	(--)		Tools						X				Display the names in the context vocabulary.
MEMFREE	(-- n)		Tools						X				Push free memory size to TOS
.MEMFREE	(--)		Tools						X				Print free memory size.
LOCK	(a --)		Multiprocessing						X				Lock a semaphore variable
UNLOCK	(a --)		Multiprocessing						X				Unlock a semaphore variable
GETCOGSTAT	(n -- n)		Multiprocessing						X				Push the current state from COG n to TOS
COGLOOP	(--)		Multiprocessing						X				Internal Mainloop for multiprocessing.
START	(-- ; <string>)		Multiprocessing						X				Start word <string> on next free COG
ISPROPTERM	(-- f)		Propeller						X				Return true if connect to PROPTERMINAL
ISHYDRA	(-- f)		Propeller						X				Return true if HYDRA Board
BOOT	(--)												The application startup point.
$			Prefix						X				Prefix for hex values
#			Prefix						X				Prefix for decimal values
%			Prefix						X				Prefix for binary values