 Table of Contents

A.8 The optional Double-Number word set

Forth systems on 8-bit and 16-bit processors often find it necessary to deal with double-length numbers. But many Forths on small embedded systems do not, and many users of Forth on systems with a cell size of 32-bits or more find that the necessity for double-length numbers is much diminished. Therefore, we have factored the words that manipulate double-length entities into this optional word set.

Please note that the naming convention used in this word set conveys some important information:

1. Words whose names are of the form 2xxx deal with cell pairs, where the relationship between the cells is unspecified. They may be two-vectors, double-length numbers, or any pair of cells that it is convenient to manipulate together.

2. Words with names of the form Dxxx deal specifically with double-length integers.

3. Words with names of the form Mxxx deal with some combination of single and double integers. The order in which these appear on the stack is determined by long-standing common practice.

Refer to A.3.1 for a discussion of data types in Forth.

A.8.6 Glossary

A.8.6.1.0360 2CONSTANT

Typical use: `x1 x2 2CONSTANT name`

A.8.6.1.0390 2LITERAL

Typical use: `: X ... [ x1 x2 ] 2LITERAL ... ;`

A.8.6.1.0440 2VARIABLE

Typical use: `2VARIABLE name`

A.8.6.1.1070 D.R

In D.R, the **R** is short for RIGHT.

A.8.6.1.1090 D2*

See: A.6.1.0320 2* for applicable discussion.

A.8.6.1.1100 D2/

See: A.6.1.0330 2/ for applicable discussion.

A.8.6.1.1140 D>S

There exist number representations, e.g., the sign-magnitude representation, where reduction from double- to single-precision cannot simply be done with DROP. This word, equivalent to DROP on two's complement systems, desensitizes application code to number representation and facilitates portability.

A.8.6.1.1820 M*/

M*/ was once described by Chuck Moore as the most useful arithmetic operator in Forth. It is the main workhorse in most computations involving double-cell numbers. Note that some systems allow signed divisors. This can cost a lot in performance on some CPUs. The requirement for a positive divisor has not proven to be a problem.

A.8.6.1.1830 M+

M+ is the classical method for integrating.

 Table of Contents

 Next Section