

SLAC PUB-2735  
May 1981  
(I)

FORTH-79 LIMITATIONS ON CODE THREADING TECHNIQUES

Terry Holmes  
Stanford Linear Accelerator Center  
Stanford, California 94305

(Presented at the FORTH Standards Conference, Rochester, N.Y.,  
May 12-15, 1981)

\*Work supported by Department of Energy under contract number  
DE-AC03-76SF00515

## PURPOSE

The matter of code threading is not treated by the 1979 FORTH Standard and hence is left as an implementation design decision. This paper examines the Standard for ambiguities in terminology and possible limitations which face the system designer. The first issue considered is whether existing systems can conform to the guidelines given in the standard. A second issue is the implementation of other varieties of threaded code to extend the capabilities of FORTH.

## TERMINOLOGY

Ritter and Walker [1, p. 212] identify four varieties of code threadings applicable to FORTH systems. This paper will refer to direct threaded code (DTC) which consists of strings of addresses of machine code, indirect threaded code (ITC) which uses pointers to such addresses, and token threaded code (TTC) which uses only indices into a table of addresses. Since the 1979 Standard states that compiled numerical values are used by the interpreter to locate machine code to be executed [2,p.2], it is clear that DTC, ITC, and TTC are all representatives of token threaded code as defined by the Standard. Hence, we rename the latter as table threaded code, retaining the abbreviation TTC.

## DTC VERSUS ITC

Since DTC and ITC are the prevalent forms of FORTH implementation at present, it is instructive to consider how the Standard applies to them. The confusion begins with the term "compilation address" which, as has been noted, the Standard defines to be a numerical value, not necessarily

an address. The alternative term "code field address" is even worse, as the code field location corresponds to the value of the compiled token only in the case of DTC FORTH. This value could be better labeled the code reference token.

Likewise, the "parameter field address" is ambiguous in both DTC and ITC FORTH. Most present ITC implementations utilize the coincidental identity of the machine code field address and the parameter (or compilation storage) field address. Thus, the word ' (tick) returns the value of the "parameter field address" even though no such address exists for assembled code words. In DTC FORTH, this is even more obvious. I suggest that the word ' return the identifying token of a word (in ITC implementations, the "code field address") to undo the accidental and unnecessary coupling of the "code field" with the "parameter field".

#### TABLE THREADED CODE

A TTC FORTH system composed of a table of addresses pointing to dictionary entries and a dictionary structure in which all addresses are replaced by 16-bit tokens (indices into the table) can be reconfigured by properly adjusting the address table whenever code is moved in the dictionary. Since this system would be highly mutable, the capability of redefining I/O vectors that many systems have adopted is unified and extended to the entire FORTH system. One further advantage of this system is that the address space of processors like the M68000 can be gracefully extended into the megabyte range with 65,536 identifiable objects, while retaining the compactness of and compatibility with 16-bit FORTH systems.

One attribute of a reconfigurable system is that the order of entry of words into the dictionary is much less binding. The insistence upon preserving the order of entry that is implied in the definition of FORGET could be an unnecessary complication in such a system. In its place may be the capability to purge individual words or entire vocabularies.

#### CONCLUDING REMARKS

At one level, it can be said that all threaded code techniques are logically equivalent. It is this level that the FORTH standard must address. By not specifying threading technique, the 1979 Standard gives designers a great deal of flexibility, but much of this is lost because of the Standard's ties to its origins in present ITC FORTH. A clarification of terminology and intent is needed with regard to this fundamental implementation consideration.

#### REFERENCES

1. Terry Ritter and Gregory Walker, "Varieties of Threaded Code for Language Implementation," Byte Magazine, September 1980.
2. FORTH Standards Team, "FORTH--79", FORTH Interest Group, October 1980.