Compiling Forth for Performance

Thomas Almy

Tektronix, Inc.

Conventional Forth environments use threaded code interpretation techniques. While this makes compilation fast, and keeps the compiler size extremely small, the execution performance tends to be five to ten times slower than that obtainable with compiled languages. Subroutine threaded code, combined with more intelligent compiling words can increase performance markedly on some systems, and several companies sell code optimizers which turn colon definitions into code words either by copying primitives to make inline code or by more sophisticated techniques.

The author has for over a year been compiling complete Forth applications directly into machine code on 8086 and Z-80 based systems. Of course, the compiler is written in Forth!

By compiling the complete application, many code optimizations are possible that are not available to mixed systems. For instance, no memory space is allocated for constants, and colon definitions become machine language subroutines. Compiled programs run faster than equivalent programs compiled in C or Pascal, yet have the same code compactness of conventional Forth. Additionally, the compiler compilers faster than most C or Pascal compilers. Performance on an IBM/PC was measured to be roughly 0.25 MOPS (using the sieve benchmark), or about 19 clock cycles per Forth primitive.

The paper will discuss the user interface and implementation concepts a Forth compiler, and will give examples of some possible optimizations that can be performed.