

Twiddle – A DSL for the Functional Bit-hacker

Michael Buch

December 24, 2018

Abstract

It is useful (and fun!) to bit twiddle i.e. perform arithmetic and manipulate data at the granularity of individual bits. Traditionally there is a compromise one has to make between using a low-level unsafe language that allows bit-twiddling versus a safe high-level language in which the type system or language design prohibit operations at bit-level (without additional complexity). The *Twiddle* domain-specific language (DSL) is an embedded language written in Scala using the Lightweight Modular Staging (LMS) framework that generates C code. Bit-twiddling is allowed in one of two forms: (1) Operations on a Bit primitive type in the Twiddle language (2) Using common programming patterns that efficiently transform to bit-twiddling code. Thus our language is a tool for the curious, a tool for safe bit-hackers and a tool for someone looking to get a bit more performance out of his high-level language.