IDFix: Infrastructure for the Design of Fixed-point systems

IDFix or *Infrastructure for the Design of Fixed-point systems* is a source to source tool which convert a C source code specified with floating data type to a C/C++ source code specified with fixed-point data type based on a analytical approach. IDFix was developed by the Cairn team at INRIA and IRISA, first as part of a joint collaboration with STMicroelectronics, and then with support from INRIA and ANR funding agency.

IDFix is developed in the GeCoS (Generic Compiler Suite) framework, an open source compiler infrastructure developed by the Cairn team.

Why IDFix

Efficient implementation of digital signal processing (DSP) applications in embedded systems requires the use of *fixed point arithmetic*. Thus, DSP applications are designed and simulated using *floating-point data types* but they are finally implemented using *fixed-point arithmetic*. The fixed-point conversion process has been identified as one of the most difficult aspect of implementing a signal processing algorithm in an embedded platform. In finite precision arithmetic, signal quantization leads to unavoidable errors which degrade the final computation quality. Thus, the design of a fixed-point system is a trade-off between the computation quality and the implementation cost. Tools to automate this phase and to explore the design space are required.

The objective of IDFix tool is therefore to determine automatically the correct fixed-point specification which minimizes the implementation cost for a given computation quality (i.e. a maximal degradation of the approximation due to reduced precision). This tool is developed in the context of source-to-source transformation for high-level synthesis. Commercial tools (like Catapult or Vivado) can be used to synthesize the architecture to the RT-Level from a fixed-point description of the application using C/C++. Thus, the goal is to bridge the gap between the floating-point description developed by the algorithm designers and the fixed-point description used as input for high-level synthesis tools.

How IDFix works

IDFix uses an analytical approach to realize the floating-point to fixed-point conversion. The user provides some information to IDFix. Obviously, the first is the C source code using floating-point data types. The C source code needs to be expanded with some #pragma. Some pragmas are necessary while others are optional. In the previous exemple, *MAIN_FUNC, OUTPUT* and *DYNAMIC* are mandatory pragmas. *WIDTH* and *DELAY* are optional.
The second information the user needs to provide is the main feature of the targeting hardware. In our case, it is an architecture model which describes available operators on the hardware platform. The last information is the accuracy loss (or accuracy constraint) allowed by the user. During this transformation, the word-length for the integer part and the fractional part are determined.

References