

ID.Fix: an EDA Tool for Fixed-Point Refinement of Embedded Systems

O. Sentieys¹, D. Menard², N. Simon¹

¹University of Rennes I, INRIA. ²INSA Rennes, IETR
olivier.sentieys@inria.fr

I. INTRODUCTION

Most of digital image and signal processing algorithms are implemented into architectures based on fixed-point arithmetic to satisfy cost and power consumption constraints associated with most of embedded and cyber-physical systems. The fixed-point conversion process (or refinement) is crucial for reducing the time-to-market and design tools to automate this phase and to explore the design space are still lacking. The ID.Fix EDA tool, based on the compiler infrastructure GECOS, allows for the conversion of a floating-point C source code into a C code using fixed-point data types. The data word-lengths are optimized by minimizing the implementation cost under accuracy constraint. To achieve low optimization time, an analytical approach is used to evaluate the fixed-point computation accuracy. This approach is valid for systems made-up of any (smooth) arithmetic operations. Commercial tools can then be used to synthesize the architecture or to perform software compilation from the output fixed-point description of the application. Thus, the goal is to bridge the gap between the floating-point description developed by algorithm designer and the fixed-point description use as input for high-level synthesis or compilation tools.

II. FLOATING-POINT TO FIXED-POINT AUTOMATIC CONVERSION IN ID.FIX

The tool inputs and output are described in Figure 1. Input is the C source code describing the application using floating-point data types. Some pragmas are used to express specific behavior and to control the design flow. This tool determines the integer and fractional part word-length of each data. The tool generates a new C code in which fixed-point data types, such as Mentor Graphics' `ac_fixed` or SystemC `sc_fixed`, are used. The implementation cost is minimized under an accuracy constraint specified by the user. An architecture model is used to determine the cost of each operator according to the operand word-length.

The tool has been developed in the GECOS framework [4], which is a compiler infrastructure leveraging source to source transformations. The different transformations carried-out in ID.Fix are defined through a GECOS script. The detailed tool flow is presented in Figure 1. The user has to define the numerical accuracy constraint applied to the output. **Output quantization noise power** is used as the numerical accuracy constraint. From the C source code, an intermediate representation is generated with the GECOS front-end. This intermediate representation, used for fixed-point conversion, is a Control and Data Flow Graph (CDFG).

A. Fixed-point conversion

In the fixed-point conversion, the dynamic range is first determined with a technique based on interval arithmetic. Then, the data word-lengths are optimized by minimizing the implementation cost under the numerical accuracy. The architecture model is defined in an XML file. Algorithm used for word-length optimization can be chosen among several. Thanks to the analytical expression of the accuracy metric, the optimization time is significantly lower than classical approaches based on fixed-point simulations.

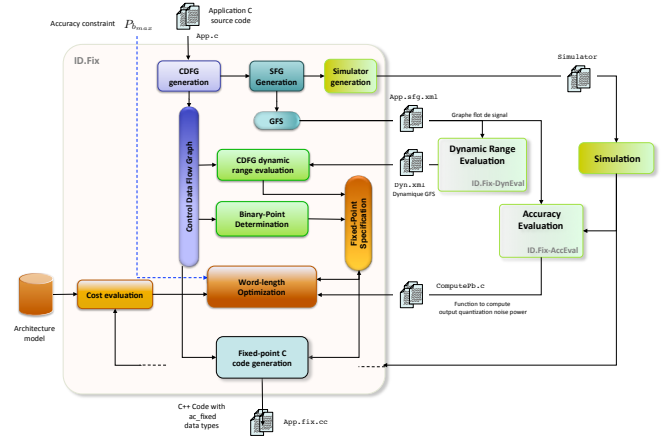


Fig. 1. Detailed Flow of ID.Fix

Finally, a fixed-point C code is regenerated at the output of the tool. For each data contributing to the computation of the output, fixed-point data types are used. This data type defines the total number of bits, the number of bits for the fractional part, the rounding mode, and the overflow mode.

B. Accuracy evaluation

The originality of our tool is the use of an analytical approach to evaluate numerical accuracy. This approach is valid for any system made-up of smooth operations. The accuracy evaluation module generates a C code describing the analytical expression of the output quantization noise power according to the operand word-length. The intermediate representation used to evaluate the accuracy is a single Signal Flow Graph representing the application. To determine the analytical expression of $P_{e,y}$, three transformations are applied to the SFG. First, the CDFG at the noise level is determined. Second, linear recurrence equations describing the application are determined. Finally, the expression of $P_{e,y}$ is computed.

REFERENCES

- [1] R. Rocher, D. Menard, O. Sentieys, and P? Scalart. Analytical approach for numerical accuracy estimation of fixed-point systems with smooth operations. *IEEE Transactions on Circuits and Systems I*, 59(10):2326 - 2339, 2012.
- [2] D. Menard, R. Rocher, and O. Sentieys. Analytical fixed-point accuracy evaluation in linear time-invariant systems. *IEEE Transactions on Circuits and Systems I*, 55(10):3197–3208, 2008.
- [3] D. Menard, R. Rocher, O. Sentieys, and O. Serizel. Accuracy Constraint Determination in Fixed-Point System Design. *EURASIP Journal on Embedded Systems*, 2008:Article ID 23197, 13 pages, 2008.
- [4] A. Floch et al.. GeCoS: A framework for prototyping custom hardware design flows. in *Proc. 13th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pp. 100-105, 2013.