Project 22703: SYRF

Deliverable 6.2.1

Mixed controller/datapath design

---

1 Approach

We have explored the generation of behavioral VHDL code from functional SIGNAL source specification. Behavioral VHDL is the highest level of VHDL, from which software or alternatively RTL hardware can be generated. Roughly speaking, behavioral VHDL compares to VHDL-RTL as C compares to assembly code. This route is in contrast to the one used in **task 6.1.1**, in which VHDL-RTL was directly generated from SIGNAL source architectural specification. The present work can be regarded as a complementary route to the former one. However, SIGNAL is still used for simulations, verifications, optimizations, and exploration of the design space.

As for optimizations, the purpose of this work was to explore how different optimizations to generate C code or VHDL code should be. In particular we have explored

- sharing/reuse of variables, vs.
- sharing/reuse of modules or components.

To this end, in generating some specific internal format to represent dependency graph labelled by clocks (the notion of dependency graph labelled by clocks is described in detail in **task 4.1**) in a way suited for the particular work done, we have, either

- attached clocks to variables, or
- attached clocks to modules or components.

As for design space exploration, the following approach has been developed. Assuming we have a SIGNAL model of the architectural specification, cf. **task 6.1.1**, we can re-interpret the functions performed by each module in a different way. For instance, when activated in a considered round, a module would compute the actual physical date of delivery of its outputs, given the date of availability of its inputs. Or, alternatively, corresponding power consumption could be computed. The idea is that everything regarding non-functional descriptions that can be modelled via a homomorphism from SIGNAL to itself can be obtained in this way. We have done this for physical time performance evaluation (cf. **task 5.2.1** for associated symbolic evaluation techniques), and are currently exploring it for power consumption performance evaluation.

2 Achieved Results

1. A pre-prototype implementing this general approach has been developed. It uses its own specific abstract machine for manipulating dependency graphs labelled by clocks, however, and is written in Java. Thus some significant work is needed in order to have the tool as a module attached to the SIGNAL toolsuite.

2. A pre-prototype implementing the physical time performance evaluation approach has been developed. However,

   (a) it does not implement the general "homomorphism" approach described above, but only implements the particular service of timing evaluation, and

   (b) it is based on an early version of SIGNAL-V4 and thus needs some addaptation to the final version, now available.

Some small or medium size examples of circuits have been designed in this way.

References

[1] A.A. KOUNTOURIS, PhD Thesis, Univ. Rennes I, in preparation.

[2] A.A. KOUNTOURIS AND P. LE GUERNIC, Profiling of SIGNAL Programs and its Application in the Timing Evaluation of Design Implementations, Proceed. of the IEE Colloq. on HW-SW Cosynthesis for Reconfigurable Systems, pp. 6/1-6/9, Feb. 22 1996, HP Labs Bristol UK http://www.irisa.fr/prive/kountour/Docs/iee-paper-final.ps.gz *(compressed postscript 45K)*