METATOR – A look into processor synthesis

This is 2014. We have tools to assist the designer (the processor design engineer per se) to speedup his/her development. Still, the processor must be designed explicitly. What would go beyond the state-of-the-art is not to have to design the golden model of the processor at all.

What I am proposing is an application-specific processor synthesis tool that goes beyond the state-of-the-art. A model generator for producing the high-level description of the processor, based only on application analysis and user-defined constraints. And for the fun of it, let's codename it METATOR, because I tend to watch too much Supernatural these days, and METATOR (messenger) is a possible meaning for METATRON, an angelic being from the Apocrypha with a human past. So think of METATOR as an upgrade (spiritual or not) to the current status of both academic and commercial ASIP design tools.

ASIPs are tuned for cost-effective execution of targeted application sets. An ASIP design flow involves profiling, architecture exploration, generation and selection of functionalities and synthesis of the corresponding hardware while enabling the user taking certain decisions.

The state-of-the-art in ASIP synthesis includes commercial efforts from Synopsys which has accumulated three relevant portfolios: the ARC configurable processor cores, Processor Designer (previously LISATek) and the IP Designer nML-based tools (previously Target Compiler Technologies); ASIPmeister by ASIP Solutions (site down?), Lissom/CodAL by Codasip, and the academic TCE and NISC toolsets. Apologies if I have missed any other ASIP technology provider!

The key differentiation point of METATOR against existing approaches is that ASIP synthesis should not require the explicit definition of a processor model by a human developer. The solution implies the development of a novel scheme for the extraction of a common denominator architectural model from a given set of user applications (accounting for high-level constraints and requirements) that are intended to be executed on the generated processor by the means of graph similarity extraction. From this automatically generated model, an RTL description, verification IP and a programming toolchain would be produced as part of an automated targeting process, in like "meta-": a generated model generating models!.

METATOR would accept as input the so-called algorithmic soup (narrow set of applications) and generate the ADL (Architecture Description Language) description of the processor. My first aim would be for ArchC but this could also expand to the dominant ADLs, LISA 2.0 and nML.

METATOR would rely upon HercuLeS high-level synthesis technology and the YARDstick profiling and custom instruction generation environment. In the past, YARDstick had been used for generating custom instructions (CIs) for ByoRISC (Build Your Own RISC) soft-core processors. ByoRISC is a configurable in-order RISC design, allowing the execution of multiple-input, multiple-output custom instructions and achieving higher performance than typical VLIW architectures. CIs for ByoRISC where generated by YARDstick, which purpose is to perform application analysis on targeted codes, identify application hotspots, extract custom instructions and evaluate their potential impact on code performance for ByoRISC.

To sum this up, METATOR is a mind experiment in ASIP synthesis technology. It automatically generates a full-fledged processor and toolchain merely from its usage intent, expressed as indicative targeted application sets.