Supporting testimonial to the ArchC infrastructure (2004)

NOTE: I was an early PhD student at the time and was quite impressed with what ArchC had to offer; I still am.

1. What ArchC is all about

ArchC is an architecture description language that exploits the already mature SystemC standard legacy. It targets the automated generation of:

a) functional and cycle-accurate instruction set simulators and at a later stage of

b) tools from the entire application development toolchain for a custom processor (compiler, assembler, linker, debugger, etc).

Simulator generation is at a very good and usable state, however needs to be extended towards the system-level (bus generation ?). Toolset generation which naturally comes next, requires more hard effort and therefore the appropriate support of the open-source community. In my personal opinion, compiler generation is so hard that it can only be possible through funding. Other issues as automated bus generation also need support.

2. Strengths of ArchC

Important points regarding ArchC are:

- Nice integration with SystemC. The ArchC user codes instruction decoding, hardware resources, etc in small ArchC files and the corresponding SystemC are automatically generated, which removes heavy burden from the user.

- Unique approach to the design of a retargetable system call library. This makes possible to benchmark real-sized applications with realistic demands, i.e. many input arguments. Many of the other simulators can't cope with that.

- Ease of introducing resource elements, e.g. multi-banked memories.

- Reuse possibilities for existing SystemC codes.

- Models of 3 processors, 2 of them with complete retargeted GNU toolchains! Plus binaries of very popular benchmarks (MediaBench, MiBench).

3. How has ArchC helped me in my research

- Ease of designing variations of a base processor. Most of my papers require tradeoff analysis on alternative architectures.
- My "drproc" model is a 5-pipeline stage RISC processor with specialized instructions for data-reuse transformations [work of Catthoor et al, IMEC]. I originally had coded an assembler (~few days) and the entire processor in plain SystemC (~month). When I found out about ArchC, it required only 3-4 8-hour days to deliver to my colleagues a much more robust cycle-accurate simulator!
- I have ran reasonably sized applications (e.g. one SourceForge application, of 25K C lines with intensive use of pointers) with no problem on the R3000 and MIPS models of ArchC.

4. What needs to be done

This work will be incomplete unless the "ArchC roadmap" is fulfilled. While I believe the compiler retargeting is a very hard issue, [addressing UNICAMP team] I am certain that your team is experienced in DSP compilation techniques (I have read some of the papers). Certainly, assembler generation, and bus wrapper generation (e.g. for OCP) plus some more models are awaited by many users, including myself.

Thank you http://www.archc.org for all the (free) fish!