

Bridging the High-level and Implementation Divide: Mission Impossible?

Victor Konrad, Intel Corporation

Extended Abstract

At Intel microprocessor design, high-level models are extensively for the design of architecture. However, the only model originating from the architectural priesthood that is directly useful for RTL implementers is the architectural model (running on an architecture simulator). The architectural model is instruction-accurate, and therefore it is useful as a golden-model for checking the RTL at instruction-level boundaries. There is, however, a huge gap between the architectural and the RTL model, and even the instruction-boundary requires a tremendous amount of “bridging”, because the actual behavior of a microprocessor is determined by events that are not visible at the architectural level (e.g. cache misses, page faults, partial pipeline flushes, interrupts etc) which have to be modeled using separate event injectors and re-steering mechanisms.

One might wish for a more useful high-level model. First and foremost, the model must be a useful sandbox for architects to experiment in. We might also wish for the high-level model to be close enough to the RTL to obviate the bridging effort described above. We might wish for a mixed-level model, where modules from the high-level and the RTL models can be substituted for each other in a plug-and-play way. We might wish for checkers to be developed and deployed at the architectural level which could later be used, unchanged, in RTL validation. Last but not least, we might wish that the code of the HLM be a starting point for the developers of RTL (as things stand now, it is seldom used in that way, and the RTL designers rely on documents, word-of-mouth communication etc. to implement the intent of the architects). We refer to such a model as *micro-architectural*.

For such wishes to become reality, the micro-architectural model must be clock accurate and match the RTL on major signals. It must therefore be quite detailed, which implies (a) slow simulation time, which limits its usefulness as an architect’s sandbox, (b) a large staff to build, maintain and keep the model and keep it in synch with downstream models,

(c) disruption in the design process in cases where the micro-architecture is in place and the micro-architectural model has to be retrofitted, in many cases to existing RTL, as an afterthought.

The focus of my work on high-level modeling was alleviating difficulty (a). The project started out with a micro-architectural model written in iHDL (Intel’s internal RTL language) and it became very clear that it was way to slow for the architects to work with. One way to deal with this problem was to develop a library of high-level models of standard components used in microprocessor designs, ranging from low-level, logic-design primitives such as priority encoders, through more complex combinational logic elements such as PLAs, and ending at high-level microprocessor-specific components such as a TLB, pseudo-LRU etc. It was not enough to merely recode these components in C or C++ (although that in itself provided a significant speedup over iHDL): the models had to use special software algorithms for speedup, which bear little resemblance to hardware implementation. In one case (that of PLAs), a new algorithm was developed which substantially improved over Espresso minimization. Modeling of TLBs revealed that the problem is isomorphic of one encountered in networking, namely subnet routing, for which special algorithms have been developed.

Although the components of the library created for this exercise showed a significant speedup over straightforward implementation, the project eventually came to naught, because of reasons (b) and (c) above. It is my feeling, based on this and other experiments at Intel, that high-level modeling will not provide microprocessor design with the hoped-for productivity increase until there is a seamless synthesis path from architecture all the way to layout. This is already becoming reality in certain domains of electronic design, but in the highly complex design of microprocessors it is several years away.