UNUM: A General Microprocessor Framework Using Guarded Atomic Actions

Nirav Dave, Michael Pellauer & Arvind
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Email: {ndave, pellauer, arvind}@mit.edu

May 19, 2006

Introduction

We present UNUM: the UNified Universal Microprocessor framework. UNUM's primary benefits are the generality and reusability of components, which facilitate architectural exploration. UNUM has been used to model a PowerPC 405 processor, with the goal of extending the model to a multiprocessor environment.

Motivation

Given the complexity of modern microprocessors architects often turn to simulations to aid in design analysis. Ideally these simulations should offer fast and accurate processor models, and allow easy extension to a multiprocessor environment. They should provide a general, extensible framework for architectural exploration, and allow for component generalization and reuse.

In pursuit of these goals designers often turn to high-level languages such as C. These have the advantages of being easy to develop and result in fast simulators. However, C-based models often bear no relation to actual hardware and thus make it difficult to accurately estimate physical features such as circuit area, timing or power consumption.

An alternative approach is to use RTL-level simulation, which increases the designer's faith in the results and has a wide range of tools available for area and timing estimation. Most importantly, the resulting RTL code can serve as a golden model for equivalence checking as project development progresses towards synthesis.

The disadvantage of RTL is that it can be extremely difficult to develop and debug. The level of detail of RTL means that it is extremely difficult

to swap modules with different characteristics. For example, the simple act of replacing an unpipelined adder with a partially-pipelined ALU capable of two addition operations per clock cycle can result in massive alterations to surrounding control logic and interconnects.

The UNUM Framework

Recent work has focused on bringing high-level languages closer to hardware through frameworks such as SystemC [3]. We present an alternative approach using the guarded atomic actions of the Bluespec methodology [2]. This raises the level of abstraction and allows the designer to consider module actions in isolation with the knowledge that the atomicity of a transaction is guaranteed.

Additionally, previous work on hardware synthesis using guarded atomic actions has shown them to be powerful abstractions of communication between hardware modules [1]. The Bluespec compiler reorganizes System Verilog interfaces into collections of "methods" accessible to other modules. UNUM is a framework of reusable interface definitions which represent components of a microprocessor.

By adhering to standard interfaces, the hardware designer can seamlessly swap modules with different capabilities, timing characteristics or instruction widths without explicit changes in control logic. Currently we are implementing a library of such modules that can be used as standard components. Finally, the Bluespec compiler can generate either C or Verilog RTL, allowing the designer to perform rapid architectural exploration that still results in a usable golden model.

Currently UNUM is being used to model single PowerPC processors. We have developed a reusable library of components such as the Decoder or ALU, and have successfully modelled an in-order PowerPC 405. In the future we wish to extend UNUM to support multiprocessor environments and cache coherence protocols. We are also looking for ways to add support for formal verification.

References

[1] Arvind, Rishiyur S. Nikhil, Daniel L. Rosenband, and Nirav Dave. High-level Synthesis: An Essential Ingredient for Designing Complex ASICs. In *Proceedings of ICCAD'04*, San Diego, CA, 2004.

[2] James C. Hoe and Arvind. Operation-Centric Hardware Description and Synthesis. *IEEE TRANSACTIONS on Computer-Aided Design of Integrated Circuits and Systems*, 23(9), September 2004.

[3] S. Y. Liao. Towards a new standard for system level design. In *Proceedings of the Eighth International Workshop on Hardware/Software Codesign*, pages 2–7, San Diego, CA, May 2000.