23

The World of I/O: A Rich Application Area for Formal Methods

Francisco Corella
Hewlett Packard Co.
8000 Foothills Blvd., Roseville, CA 95747-5649, USA
phone: (916)785-3504
fax: (961)785-3096
email: fcorella@rosemail.rose.hp.com

Abstract

Proponents of formal methods are making a great effort to demonstrate the impact that formal methods can have on the hardware design process. To this purpose, they are concentrating on a few hot areas where the need for formal techniques seems more obvoius, and the potential impact seems greater. Such areas include, for example, arithmetic circuits, pipelined and superscalar processor architectures, cache coherence, formal memory models, and ATM switches. There is one area, however, that has been somewhat neglected, even though it is perhaps the area where formal methods could have the greatest impact. This is the area of I/O, including the specification and verification of I/O architectures, I/O bus protocols, I/O bus adaptors, I/O processors and I/O devices.

The world of I/O is evolving rapidly. Multimedia and networking require faster I/O devices, higher interconnection bandwith, and flexible peer-to-peer communication among I/O devices. This in turn fuels fast innovation in the area of I/O architectures. At the same time, backwards compatibility constraints require support for older devices and older protocols. All this amounts to great conceptual complexity, which causes protocol flaws and design errors that result in deadlock, starvation or data corruption. Formal methods, especially at the system level, can help master this complexity, prevent some errors, find others, and, in some cases, provide an assurance of correctness that is very welcome by I/O designers and architects. The small formal verification group at Hewlett-Packard has been applying formal methods to I/O problems for a few years, and this has helped considerably with the integration of formal techniques in the design process at HP.

An example of both complexity and conceptual richness is the PCI 2.1 protocol. It has evolved from an earlier and simpler protocol, which explains some awkward features. At the same time, however, it has introduced a most interesting transaction ordering mechanism for communication among devices

located anywhere on an arbitratry acyclic network of buses. We have formally proved a result that shows how this ordering mechanism implements a shared memory paradigm that is fundamentally different, and in some sense more general, than both sequential consistency and the weaker memory consistency models found in shared memory multiprocessor systems.