Using Wired for Design Exploration

Emil Axelsson Koen Claessen Mary Sheeran

Chalmers University of Technology
{emax,koen,ms}@cs.chalmers.se

Lava has been used successfully in structural circuit description and layout generation for FPGAs [2]. It has been demonstrated that Lava is suitable for describing so-called "clever" or adaptive circuits – circuits whose structure adapts to properties of its context [4].

The rising problems associated with chip design in the deep sub-micron era call for new methods that are able to account for low-level effects already in higher-level descriptions. In particular, interconnect wires need to be modeled in order to get reliable estimates of non-functional properties, such as delay and power consumption. We are working on a system – Wired [1] – which aims to bring the ideas from Lava down to lower levels. What distinguishes Wired from other layout-aware languages is that wires are modelled explicitly. Also, we have settled on a relational model of information flow, giving some decoupling of information flow from the structural combinators, in the style of Ruby [3].

Wired supports a number of different circuit analyses. We can, for example, analyse for signal flow direction and delay. Our most advanced delay model uses Elmore approximation to appropriately take account of fanout and load capacitance. This analysis is, we feel, an important advance on the version of Wired presented at DCC 2004. It makes essential use of the relational aspect of Wired, with resistances flowing in one direction and capacitances in the other. We have, in Wired, described some tiny circuits that adapt to the delays in their inputs using the idea of clever circuits from Lava [4]. But scaling this up has not been as straight-forward as we had hoped. While adapting to forwards properties (like Lava-style unit delay) can usually be done greedily, adapting to bi-directional properties (like Elmore delay) seems to require some back-tracking, which does not come very naturally in Wired. Still, we hope to tackle more complicated delay-adaptive circuits in time for DCC.

In the talk, we will concentrate not so much on the analyses as on our ideas about a flow that combines Lava and Wired to support design exploration. The descriptions start off in Lava, and are then transformed, surprisingly smoothly, into corresponding Wired descriptions. The key point is to enable switching from the functional to the relational view, and back – corresponding to the mixing of logical and physical design. To keep things concrete, we will consider some classical parallel prefix structures as well as a family of new ones [5].

The nearest related work that we know of is that by Seger and his co-workers on Intel's IDV system. IDV does not have the explicitly placed wires that we currently have (and we too may need to reconsider the exact level of detail that we wish to support). IDV is not, as far as we know, relational in the sense that Wired is. So the pros and cons of being relational will be a theme of this talk.

Acknowledgements

This research has received Intel-custom funding from the Semiconductor Research Corporation. It is also funded by the Swedish research agency Vetenskapsrådet.

References

1. Emil Axelsson, Koen Claessen, and Mary Sheeran. Wired: Wire-Aware circuit design. In *Correct Hardware Design and Verification Methods (CHARME)*, volume 3725 of *LNCS*, pages 5–19. Springer, 2005.
2. K. Claessen, M. Sheeran, and S. Singh. The Design and Verification of a Sorter Core. In *Correct Hardware Design and Verification Methods (CHARME)*, volume 2144 of *LNCS*. Springer, 2001.
3. G. Jones and M. Sheeran. Circuit design in Ruby. In J. Staunstrup, editor, *Formal methods for VLSI Design*. Elsevier, 1990.
4. M. Sheeran. Generating fast multipliers using clever circuits. In *Formal Methods in Computer-Aided Design, FMCAD*, volume 3312 of *LNCS*, pages 6–20. Springer, 2004.
5. M. Sheeran. Hardware Design and Functional Programming: a Perfect Match. *Journal of Universal Computer Science*, 11(7):1135–1158, July 2005. `http://www.jucs.org/jucs_11_7/hardware_design_and_functional`.