# Abstract

## Experience with Practical Formal Verification at an Industrial Scale

Tom Melham

Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford, OX1 3QD, England

10th Workshop on Automated Reasoning
Liverpool, 15–16 April 2003

The subtitle of the Workshop on Automated Reasoning series [1] is *Bridging the Gap between Theory and Practice*. Building such bridges is very challenging: it involves tackling all kinds of messy non-technical problems—issues that theoreticians may find boring or irrelevant—and it may require you to make unpleasant engineering compromises in the implementation of an otherwise elegant theory. On the other hand, it can be tremendously illuminating scientifically and inspire new theoretical ideas. And bridge building can, in my experience, also be enormously fun and rewarding.

For this invited talk at the tenth workshop, I describe some collaborative work with researchers at Intel's Strategic CAD Labs [2] that addresses precisely this theme. The aim of this research, which is published in [3, 4], is to make formal verification a practical, everyday tool for industrial hardware design—specifically high-performance microprocessor design.

Successful application of formal methods in this arena requires the best available theoretical basis for verification technology, embodied in highly tuned and well-engineered software implementations. The latter are, of course, beyond the scope of most research groups; serious implementations require orders of magnitude more development effort than the typical research project can afford. But the research community can and energetically does engage with the former, with much formal hardware verification research aimed at new algorithms and focused on overcoming capacity limits.

But any serious attempt to bridge the gap between theory and practice for industrial verification will face many difficulties other implementation efficiency and algorithm capacity. Equally important is the problem of managing the complexity of the verification activity itself. The work I describe in this talk attacks this problem by coupling implementation engineering and research into verification algorithms with research on verification *methodology*. What is meant by 'methodology' here is a systematic approach to organising a large verification effort. This includes a clearly articulated plan for the sequence and purpose of each of the many interdependent activities of a typical verification project, together with a guiding structure for the verification code artifacts to be produced.

The approach is supported by a formal verification environment called Forte, which combines symbolic trajectory evaluation [5], an efficient, linear temporal logic model-checking algorithm, with lightweight theorem proving. The model checker and theorem prover are tightly integrated through a general-purpose functional programming language. The combination of model checking, theorem proving, and a general-purpose programming language allows the verification environment to be customised and large verification efforts to be organised and scripted effectively.

The talk illustrates the methodology and the Forte environment with the verification of an IEEE-compliant, extended precision floating-point adder. The adder was verified as part of a large scale effort at Intel to verify the IEEE-compliance of the FADD, FSUB, FMUL, FDIV, FSQRT, and FPREM operations of the Intel Pentium Pro processor [6].

## Acknowledgements

I thank the organisers of ARW 2003 for their invitation to speak at the workshop in Liverpool. This talk describes the results of joint research with Robert Jones, John O'Leary, and Carl Seger of Intel Strategic CAD Labs and Mark Aagaard of the University of Waterloo [3, 4].

## References

[1] `http://www.dcs.kcl.ac.uk/staff/endriss/ARW/`

[2] `http://www.intel.com/research/scl/`

[3] M. D. Aagaard, R. B. Jones, T. F. Melham, J. W. O'Leary and C.-J. H. Seger, 'A Methodology for Large-Scale Hardware Verification', in *Formal Methods in Computer-Aided Design: Third International Conference, FMCAD 2000: Austin, November 2000: Proceedings*, edited by W. A. Hunt, Jr. and S. D. Johnson, Lecture Notes in Computer Science, vol. 1954 (Springer-Verlag, 2000), pp. 263–282.

[4] R. B. Jones, J. W. O'Leary, C.-J. H. Seger, M. D. Aagaard and T. F. Melham, 'Practical Formal Verification in Microprocessor Design', *IEEE Design & Test of Computers*, vol. 18, no. 4 (July/August 2001), pp. 16–25.

[5] C.-J. H. Seger and R. E. Bryant, 'Formal verification by symbolic evaluation of partially-ordered trajectories', *Formal Methods in System Design*, vol. 6, no. 2 (March 1995), pp. 147–189.

[6] J. O'Leary, X. Zhao, R. Gerth, and C.-J. H. Seger, 'Formally verifying IEEE compliance of floating-point hardware', *Intel Technology Journal* (First Quarter, 1999). Available online at `http://developer.intel.com/technology/itj/`.