FME Fellow Nomination of Warren A. Hunt, Jr.

J Strother Moore
Department of Computer Science
University of Texas at Austin
Austin, TX 78712 USA
moore@cs.utexas.edu

December 13, 2018

1 Preface

It is my pleasure to nominate Warren A. Hunt, Jr., for the FME Fellowship.

I describe his technical contributions and cite concrete evidence of his contributions using two styles of references. Boldface references starting with an alphabetic character, e.g., [**B1**], refer to publications by Hunt (possibly with co-authors) and listed under those labels in his CV. I have attached his CV, which may also be found online at `www.cs.utexas.edu/users/hunt/vita/-may-2018.pdf`. References that are entirely numeric, e.g., [1], refer to the bibliography at the end of this nomination. These are by authors other than Hunt.

By way of full disclosure, Hunt's PhD dissertation was supervised by Bob Boyer and me at the University of Texas and virtually all of his work has been done with the the theorem provers Nqthm and ACL2, of which I am a co-author with Boyer and Matt Kaufmann. Nqthm (`www.cs.utexas.edu/users/boyer/-ftp/nqthm/index.html`) and ACL2 (`www.cs.utexas.edu/users/moore/acl2`) are distributed in source code form and without charge.

2 Name and Affiliation of Nominee

Professor Warren A. Hunt, Jr
Computer Science Department
University of Texas at Austin
2317 Speedway, Stop D9500
Austin, TX 78712 - 1757
hunt@cs.utexas.edu

3 Technical Contributions

Hunt has been a leader in the mechanized verification of microprocessor designs since his groundbreaking dissertation of 1985. Indeed, it is fair to say that Hunt founded the field of mechanized verification of functional correctness of industrial microprocessor designs.

The story starts over 30 years ago with Hunt's 1985 dissertation in which, using Nqthm [3, 4], he proved the functional correctness of a gate-level design of a 16-bit wide full function microprocessor he called the FM8501 [**T1,B1**]. In particular, he proved that the gate-level design implemented a certain machine code.

Between 1987 and 1992, he and others (including me) at Computation Logic, Inc. (CLI) verified a hardware/software stack. The bottom of the initial stack was Hunt's FM8501, although midway through the project Hunt produced a 32-bit version called FM8502 on which the final stack was based. I implemented and verified an assembler/linker/loader for it, and students implemented various extremely simple high-level language compilers, an operating system, and applications programs.

All the pieces fit together in the sense that theorems were mechanically proved establishing that high-level applications ran correctly on the hardware. The CLI stack was first published in 1989 [1].

However, between 1989 and 1992, Hunt developed a formal hardware design language modeled on NDL by LSI Logic, Inc. and used it to verify a more sophisticated 32-bit microprocessor, FM9001 [**P8,P10,P12**]. Because NDL was a commercial design language it was possible to fabricate FM9001, which CLI did (via LSI Logic, Inc.). We then ported the stack to FM9001 producing the world's first verified stack running on a verified microprocessor design that had been fabricated.

This inspired – and continues to inspire – many projects world wide whose common denominator is the mechanically verified stacking of various intermediate languages and components into a coherent system. The first such project outside CLI was the European ESPRIT-sponsored Provably Correct Systems (ProCoS) project. In Appendix B of *Towards Verified Systems* [2], Langmaack and Ravn explicitly state that ProCoS was inspired by the CLI stack. More recently, both the SEL-4 effort at NICTA [9] and the CompCert project at Inria [10] focus on much more realistic system components than those used in the CLI stack but are not (yet) verifiably linked to hardware at the bottom level. Both groups acknowledge the influence of the CLI stack.

Hunt, however, had his sights on industrial penetration of formal methods.

In 1993, he landed a contract for CLI to verify the microcode engine of a commercial digital signal processor, the Motorola CAP DSP [**P15,P17**] and, with Bishop Brock, he formalized the engine's design and verified that it implemented the microcode semantics, discovering numerous pipeline hazards in the process. This was done with ACL2 [8, 7]. They created an executable ACL2 predicate to scan microcode to detect problematic sequences. As a formal predicate the tool was a necessary hypothesis in the verified statement of correctness;

but as an executable algorithm it was used within the design group for several more years simply to confirm the lack of hazards.

In 1995, Hunt convinced AMD to contract with CLI to verify the floating point division microcode for the AMD K5 microprocessor, AMD's chip competing with Intel's Pentium I. My colleague Matt Kaufmann and I verified that design prior to fabrication[11] and Hunt helped convince AMD to start a formal verification team consisting of three people who had worked on the CLI stack.

Meanwhile, Hunt continued to work on the formalization of commercial hardware design languages (HDLs) such as Verilog. His formal language was named DE (for "Dual Eval") **[P18]**. In addition, he and his student Sawada used DE to design and verify the FM9801, an out-of-order microprocessor with speculative execution, exceptions, and self-modifying code **[P20]**. To the best of my knowledge, the FM9801 project has been the most complete verified microprocessor specification for 20 years now.

But Hunt's real objective was to integrate theorem-proving based formal methods into the industrial microprocessor design workflow. His chance came in May, 2007, when he was challenged by Glenn Henry, the President and cofounder of Centaur Technology[1] to verify the floating-point adder design for the Via Nano.

The unit to be verified was actually composed of four adders so that the unit could simultaneously add four pairs of single-precision numbers, two pairs of double-precision numbers, or one pair of extended-precision numbers. The unit could complete four floating-point additions in two steps (clock cycles), and it was pipelined so it could produce four results every clock cycle. The unit's description involved 33,700 lines of Verilog in 680 modules; its implementation involved 432,322 transistors. The floating-point adder was part of a larger media unit that had 1074 inputs, including 26 clock inputs, and 374 output signals. The unit could perform over 100 other operations – and there were many ($>$ 1000) inputs that had to be set properly to force it to perform the additions of interest.

In order to specify and verify this unit Hunt, and PhD student Sol Swords, had to upgrade the formal HDL DE to handle the Verilog subset used **[P45]**, test the formal model of the design against Centaur's own simulator, implement unique object representation and memoization in ACL2 **[P30]** to handle the huge objects involved, implement symbolic simulation [13] (aka "bit blasting") in ACL2, and then combine symbolic simulation with ACL2 theorem proving. (The mixing of symbolic simulation and theorem proving were done first in Forte at Intel[12, 14], where Forte was subsequently used to verify the whole execution cluster (all 2,000+ μops) in the design of the Willamette P4 processor [6].) Hunt and Swords then verified the Via Nano FPU adder – after finding a bug in the extended precision case that required a mask change to correct. This bug discovery is the event that triggered Centaur's investment in formal

[1] Centaur designs high performance, low-cost x86 compatible microprocessors; it is an independent subsidiary of VIA Technologies, Inc., a Taiwanese chipset manufacturer and is the third largest x86 manufacturer after Intel and AMD.

verification. Hunt helped Centaur hire an experienced formal verification team, including Swords.

Subsequently, in his 2010 PhD dissertation, Swords, supervised by Hunt, used ACL2 to prove that his improved ACL2 bit blasting code[16] is a sound proof technique in ACL2, thus eliminating the mix of logics formerly supporting correctness. Everything is now done in the ACL2 logic.

Centaur has included ACL2-based tools in its daily workflow for seven years: every night after designers have checked in their modifications, ACL2 tools are run on the designs, producing feedback every morning on possible bugs introduced the day before. This nightly process involves more than 100 cores checking an in-house corpus of formal results whose size exceeds all of ACL2's public regression suite. They are able to formally verify an extended floating point adder in just minutes. This has fundamentally changed Centaur's workflow, reducing the need for testing while increasing confidence [15].

After further improvement and elaboration by Centaur's verification team, the ACL2 Verilog Toolkit is distributed for free as part of ACL2's Community Books repository [5]. See the documentation by entering "VL" into the search box of the ACL2+Books link mentioned under the Manuals link of the ACL2 home page [8]. The toolkit is extensively used at Oracle and other groups doing microprocessor design. For example, see the slides by Greg Grohoski VP, Hardware Development, Oracle, at `www.cs.utexas.edu/users/moore/acl2/-workshop-2017/slides-accepted/grohoski-ACL2_talk.pdf`.)

The ACL2 theorem prover has seen sustained industrial use since the mid 1990s. Companies that have used ACL2 regularly include AMD, Arm, Centaur Technology, IBM, Intel, Kestrel Institute, Motorola/Freescale, Oracle, and Rockwell Collins. This industrial penetration would not have occurred were it not for Hunt's constant pushing of the envelope.

Hunt continues to make contributions to formal methods technology. His recent hardware verification efforts have focused on an efficiently executable formal specification of the x86 ISA at both the system and user levels [**P61**], and self-timed and asynchronous circuits [**P64,P65**]. The latter work, and its integration with the formal HDL, may offer applications to more general concurrent systems verification of the sort studied with CSP and CCS.

Hunt's contributions to FM have been both visionary and technical. His leadership in formalizing and verifying increasingly realistic microprocessor designs has demonstrated what is formally possible. That has spurred many groups to action. In addition, by leading the effort to produce a tool chain in the functional language of a publicly available, free, well-documented, flexible, and fast verification environment in which all proofs were conducted in a single mathematical logic powerful enough to capture most functional properties of industrial microprocessor designs Hunt helped demonstrate that even a relatively small design group, such as that at Centaur, could leverage formal methods with publicly available tools.

Hunt has also worked extensively with Marijn Heule and Matt Kaufmann to develop an efficient SAT proof checking formalism and ACL2-verified checker. [**P50,P51,P53,P54,P58,P60,P63**]. Their formalism has been adopted by

most competitive SAT checkers and the ACL2-verified checker is routinely used at SAT competitions today to check the claims of the winners.

4 Supporting the Worldwide FM Community

Hunt has also played a crucial role in the success of the Formal Methods in Computer Aided Design (FMCAD) conference.

The FMCAD Conference is held (generally) on alternate years in Europe (see `www.fmcad.org`), and many Europeans have published and participated in FMCAD. When CHARME was declining, FMCAD absorbed the CHARME Conference after CHARME 2005 was held.

His engagement with FMCAD started when he was co-chair of the conference in 2000. He found there was no organization behind the conference and so he personally guaranteed hotel bookings, etc. Shortly thereafter he incorporated the FMCAD company and set up business processes, banking, payment systems, etc. As chairman of the five person Steering Committee for the past 18 years he has had a hand in the selection of conference chairs but otherwise participates in FMCAD as any member of the community might. Since FMCAD's incorporation, Hunt has dispersed funds to the conference chair, guaranteed bookings, collected registration payments, and paid the taxes for FMCAD, Inc. This stable financial and technical support, together with the fact that FMCAD presents papers across the whole spectrum of mechanized formal methods (e.g., model checking, equivalence checkers, SAT, SMT, general-purpose theorem proving, probabilistic methods, etc.) in both hardware and software verification, helps explain why FMCAD has become a premier conference in verification.

Hunt has also trained many students who went on to lead industrial verification efforts including Jun Sawada (IBM), Erik Reeber (Intel), Sol Swords and Shilpi Goel (Centaur), and David Rager (Oracle). In each case these former students have gone on to play key roles in the verification of commercial products.

In addition, Hunt's CV lists 25 occasions on which he has been an Invited or Keynote speaker at conferences other than FMCAD, and 39 conferences (other than FMCAD and the ACL2 conference, in which I play a major role) in which he has been a member of PC or otherwise involved in organizing.

5 Names and Affiliations of Nominators

- Armin Biere, Johannes Kepler University Linz, Austria.
- Roderick Bloem, Graz University of Technology, Austria
- Dominique Borrione, (retired) University Grenoble Alpes, France
- Alessandro Cimatti, Fondazione Bruno Kessler, Trento, Italy
- Jo Ebergen, Oracle Labs, Redwood Shores, California, USA

- Glenn Henry, President, Centaur Technology, Austin, Texas, USA
- Tony Hoare, (retired) Oxford University and Microsoft Cambridge, England
- Barbara Jobstmann, EPFL, Lausanne, Switzerland, and Cadence Design Systems
- Viktor Kuncak, EPFL, Lausanne, Switzerland
- J Strother Moore, (retired) University of Texas at Austin, Texas, USA
- Carl Seger, Chalmers University of Technology, Göteborg, Sweden
- Natasha Sharygina, Universita della Svizzera Italiana, Lugano, Switzerland
- Mary Sheeran, Chalmers University of Technology, Göteborg, Sweden
- Ivan Sutherland, Portland State University, Portland, Oregon, USA
- Georg Weissenbacher, TU Wien, Vienna, Austria

References

[1] W.R. Bevier, W. A. Hunt, Jr., J S. Moore, and W.D. Young. Special issue on system verification. *Journal of Automated Reasoning*, 5(4):409–530, 1989.

[2] J. Bowen, editor. *Towards Verified Systems*. Elsevier Science, 1994.

[3] R. S. Boyer and J S. Moore. *A Computational Logic*. Academic Press, New York, 1979.

[4] R. S. Boyer and J S. Moore. *A Computational Logic Handbook, Second Edition*. Academic Press, New York, 1997.

[5] ACL2 User Community. ACL2 community books.

[6] Roope Kaivola, Rajnish Ghughal, Naren Narasimhan, Amber Telfer, Jesse Whittemore, Sudhindra Pandav, Anna Slobodová, Christopher Taylor, Vladimir Frolov, Erik Reeber, and Armaghan Naik. Replacing testing with formal verification in intel© coretm i7 processor execution engine validation. In *Proceedings of the 21st International Conference on Computer Aided Verification*, CAV '09, pages 414–429, Berlin, Heidelberg, 2009. Springer-Verlag.

[7] M. Kaufmann, P. Manolios, and J S. Moore. *Computer-Aided Reasoning: An Approach*. Kluwer Academic Press, Boston, MA., 2000.

[8] M. Kaufmann and J S. Moore. The ACL2 home page. In *http://www.cs.utexas.edu/users/moore/acl2/*. Dept. of Computer Sciences, University of Texas at Austin, 2018.

[9] Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood. seL4: Formal verification of an os kernel. In *ACM Symposium on Operating Systems Principles*, pages 207–220, October 2009.

[10] Xavier Leroy. Formal verification of a realistic compiler. *Commun. ACM*, 52(7):107–115, July 2009.

[11] J S. Moore, T. Lynch, and M. Kaufmann. A mechanically checked proof of the correctness of the kernel of the AMD5K86 floating point division algorithm. *IEEE Transactions on Computers*, 47(9):913–926, September 1998.

[12] J. OLeary, X. Zhao, R. Gerth, and C.-J.H. Seger. Formally verifying ieee compliance of floating-point hardware. *Intel Technology Journal*, 3(1):1–14, 1999.

[13] C.-J. H. Seger and R. E. Bryant. Formal verification by symbolic evaluation of partially-ordered trajectories. *Formal Methods in System Design*, 6(2):147–190, 1995.

[14] C.-J.H. Seger, R.B. Jones, J.W. O'Leary, T. Melham, M.D. Aagaard, C. Barrett, and D. Syme. An industrially effective environment for formal hardware verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(9):1381–1405, 2005.

[15] Anna Slobodova, Jared Davis, Sol Swords, and Jr. Warren Hunt. A flexible formal verification framework for industrial scale validation. In Satnam Singh, editor, *9th IEEE/ACM International Conference on Formal Methods and Models for Codesign (MEMOCODE)*, pages 89–97. IEEE, 2011.

[16] Sol Swords. A verified framework for symbolic execution in the ACL2 theorem prover. Ph.D. thesis, University of Texas at Austin, 2010. `http://hdl.handle.net/2152/ETD-UT-2010-12-2210`.