























## REFERENCES

- [1] M. Aagaard and M. Leese. 1991. A formally verified system for logic synthesis. In *[1991 Proceedings] IEEE International Conference on Computer Design: VLSI in Computers and Processors*. 346–350. <https://doi.org/10.1109/ICCD.1991.139915>
- [2] Clark Barrett, Pascal Fontaine, and Cesare Tinelli. 2017. *The SMT-LIB Standard: Version 2.6*. Technical Report. Department of Computer Science, The University of Iowa.
- [3] Robert Brayton and Alan Mishchenko. 2010. ABC: An academic industrial-strength verification tool. In *Computer Aided Verification*, Tayssir Touili, Byron Cook, and Paul Jackson (Eds.). Springer, Berlin, Heidelberg, 24–40.
- [4] Cadence. 2019. Conformal Equivalence Checker. <https://bit.ly/2mkp0aa>
- [5] Joonwon Choi, Muralidaran Vijayaraghavan, Benjamin Sherman, Adam Chlipala, and Arvind. 2017. Kami: a Platform for High-Level Parametric Hardware Specification and Its Modular Verification. *Proc. ACM Program. Lang.* 1, ICFP, Article 24 (Aug. 2017), 30 pages. <https://doi.org/10.1145/3110268>
- [6] Robert L. Constable et al. 1986. Implementing Mathematics with The Nuprl Proof Development System.
- [7] Leonardo de Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, C. R. Ramakrishnan and Jakob Rehof (Eds.). Springer, Berlin, Heidelberg, 337–340.
- [8] João Paulo Pizani Flor, Wouter Swierstra, and Yorick Sijsling. 2018. Pi-Ware: Hardware Description and Verification in Agda. In *21st International Conference on Types for Proofs and Programs (TYPES 2015) (Leibniz International Proceedings in Informatics (LIPIcs))*, Tarmo Uustalu (Ed.), Vol. 69. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 9:1–9:27. <https://doi.org/10.4230/LIPIcs.TYPES.2015.9>
- [9] Yann Herklotz and John Wickerson. 2019. *Verismith: FPGA '20 Artifact*. <https://doi.org/10.5281/zenodo.3559802>
- [10] IEEE Std 1364.1 2005. *IEEE Standard for Verilog Register Transfer Level Synthesis Standard*. 1–116 pages. <https://doi.org/10.1109/IEEESTD.2005.339572>
- [11] Intel. 2019. Intel FPGA SDK for OpenCL software technology. <https://intel.ly/2WXoTj8>
- [12] Intel. 2019. Intel Quartus. <https://intel.ly/2m7wbCs>
- [13] Xavier Leroy. 2006. Formal Certification of a Compiler Back-end or: Programming a Compiler with a Proof Assistant. In *Conference Record of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '06)*. ACM, New York, NY, USA, 42–54. <https://doi.org/10.1145/1111037.1111042>
- [14] Christopher Lidbury, Andrei Lascu, Nathan Chong, and Alastair F. Donaldson. 2015. Many-core Compiler Fuzzing. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '15)*. ACM, New York, NY, USA, 65–76. <https://doi.org/10.1145/2737924.2737986>
- [15] Andreas Löföw, Ramana Kumar, Yong Kiam Tan, Magnus O. Myreen, Michael Norrish, Oskar Abrahamsson, and Anthony Fox. 2019. Verified Compilation on a Verified Processor. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '19)*. ACM, New York, NY, USA, 1041–1053. <https://doi.org/10.1145/3314221.3314622>
- [16] William McDonald and Janny Liao. 2006. Logic Equivalence Checking has Arrived for FPGA Developers. In *Design and Verification Conference (DVCon)*.
- [17] William M McKeeman. 1998. Differential Testing for Software. *Digital Technical Journal* 10, 1 (1998), 100–107.
- [18] Ghassan Misherghe and Zhendong Su. 2006. HDD: Hierarchical Delta Debugging. In *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*. ACM, New York, NY, USA, 142–151. <https://doi.org/10.1145/1134285.1134307>
- [19] Boris Ratchev, Mike Hutton, Gregg Baeckler, and Babette van Antwerpen. 2003. Verifying the Correctness of FPGA Logic Synthesis Algorithms. In *Proceedings of the 2003 ACM/SIGDA Eleventh International Symposium on Field Programmable Gate Arrays (FPGA '03)*. ACM, New York, NY, USA, 127–135. <https://doi.org/10.1145/611817.611837>
- [20] John Regehr, Yang Chen, Pascal Cuoq, Eric Eide, Chucky Ellison, and Xuejun Yang. 2012. Test-case Reduction for C Compiler Bugs. In *Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '12)*. ACM, New York, NY, USA, 335–346. <https://doi.org/10.1145/2254064.2254104>
- [21] Stephen Williams. 2019. Icarus Verilog. <http://iverilog.icarus.com/>
- [22] Clifford Wolf. 2019. VlogHammer. <https://bit.ly/2kCxjO3>
- [23] Clifford Wolf. 2019. Yosys Open SYnthesis Suite. <https://bit.ly/2kAXg0q>
- [24] Xilinx. 2019. Vivado Design Suite. <https://bit.ly/2wZAmld>
- [25] Xilinx. 2019. XST Synthesis Overview. <https://bit.ly/2lGtKjL>
- [26] Xuejun Yang, Yang Chen, Eric Eide, and John Regehr. 2011. Finding and Understanding Bugs in C Compilers. In *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '11)*. ACM, New York, NY, USA, 283–294. <https://doi.org/10.1145/1993498.1993532>
- [27] Michal Zalewski. 2015. American fuzzy lop. <http://lcamtuf.coredump.cx/afl/>
- [28] A. Zeller and R. Hildebrandt. 2002. Simplifying and Isolating Failure-Inducing Input. *IEEE Transactions on Software Engineering* 28, 2 (Feb 2002), 183–200. <https://doi.org/10.1109/32.988498>