# Functional Programming and Hardware Design: Still Interesting after All These Years

Mary Sheeran

Chalmers University of Technology

ms@chalmers.se

## Abstract

Higher order functions provide an elegant way to express algorithms designed for implementation in hardware [1, 6–9]. By showing examples of both classic and new algorithms, I will explain why higher order functions deserve to be studied.

Next, I will consider the extent to which ideas from functional programming, and associated formal verification methods, have influenced hardware design in practice [3–5, 10]. What can we learn from looking back?

You might ask "Why are methods of hardware design still important to our community?". Maybe we should just give up? One reason for not giving up is that hardware design is really a form of parallel programming. And here there is still a lot to do! Inspired by Blelloch's wonderful invited talk at ICFP 2010 [2], I still believe that functional programming has much to offer in the central question of how to program the parallel machines of today, and, more particularly, of the future. I will briefly present some of the areas where I think that we are poised to make great contributions. But maybe we need to work harder on getting our act together?

***Categories and Subject Descriptors*** D.1.1 [*Applicative (Functional) Programming*]; B.6.3 [*Hardware Description Languages*]; D.1.3 [*Concurrent Programming*]: Parallel Programming

***Keywords*** Hardware design, parallel algorithms, functional programming, higher order functions, parallel programming

## References

[1] P. Bjesse, K. Claessen, M. Sheeran, and S. Singh. Lava: hardware design in Haskell. In *Int. Conf. on Functional Programming*, pages 174–184. ACM, 1998.

[2] G. Blelloch. Functional Parallel Algorithms, invited talk. In *Int. Conf. on Functional Programming*. ACM, 2010.

[3] K. Claessen, N. Een, M. Sheeran, and N. Sorensson. SAT-solving in practice. In *Discrete Event Systems, 2008. WODES 2008. 9th International Workshop on*, pages 61–67. IEEE, 2008.

[4] C. Seger. Integrating design and verification-from simple idea to practical system. In *Formal Methods and Models for Co-Design, 2006. MEMOCODE'06. Proceedings. Fourth ACM and IEEE International Conference on*, pages 161–162. IEEE, 2006.

[5] 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. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(9):1381–1405, 2005.

[6] M. Sheeran. muFP, a language for VLSI design. In *Proceedings of the ACM Symposium on LISP and Functional Programming*, pages 104–112. ACM, 1984.

[7] M. Sheeran. Finding regularity: describing and analysing circuits that are not quite regular. In *Correct Hardware Design and Verification Methods*, volume 2860 of *LNCS*, pages 4–18. Springer, 2003.

[8] M. Sheeran. Generating Fast Multipliers Using Clever Circuits. In *Formal Methods in Computer-Aided Design*, volume 3312 of *LNCS*, pages 6–20. Springer, 2004.

[9] M. Sheeran. Functional and dynamic programming in the design of parallel prefix networks. *J. Funct. Program.*, 21(1):59–114, 2011.

[10] M. Sheeran, S. Singh, and G. Stålmarck. Checking safety properties using induction and a SAT-solver. In *Formal Methods in Computer-Aided Design*, volume 1954 of *LNCS*, pages 127–144. Springer, 2000.



*ICFP'15*, August 31 – September 2, 2015, Vancouver, BC, Canada
ACM. 978-1-4503-3669-7/15/08
http://dx.doi.org/10.1145/2784731.2789053