

BAT: Bit-Level Analysis Tool

Panagiotis Manolios¹, Sudarshan K. Srinivasan², Daron Vroon¹, and Yimin Zhang¹

¹ College of Computing

² School of Electrical & Computer Engineering
Georgia Institute of Technology
Atlanta, GA-30318

{manolios@cc.gatech.edu, darshan@ece.gatech.edu, vroon@cc.gatech.edu,
ymzhang@cc.gatech.edu}

1 System Description

Bit-level Analysis Tool (BAT) [1] solves RTL-level bounded model checking and k -step induction problems. Given a machine description and LTL property in our own S-Expression based language (similar to Lisp or Scheme in syntax) and a number of steps, BAT checks the property using SAT-based methods and returns a counter-example if one exists.

We carefully designed the BAT tool to be simple to use, expressive, and as general as possible. A formal specification of its syntax and semantics can be found in a tech report linked from the authors' web pages. Because of its generality, the BAT specification language can easily be the target language of synthesizable subsets of Verilog and VHDL. In fact BAT can handle features of these languages not present in other state of the art bit-level analysis tools.

BAT is implemented in Lisp, and can be used interactively using Lisp's built-in development environment. This makes it easy to set up automatic interaction between Lisp and BAT. For example, users can write functions to generate BAT models given a set of input parameters, or even add features to the language by writing code to translate new features into the core BAT language.

We are submitting the BAT tool for the QF.BVUF problem division of SMT-COMP 2006. For our submission, we choose 7 as the seed number.

References

- [1] P. Manolios, S. K. Srinivasan, and D. Vroon. Automatic memory reductions for rtl-level verification. In *ACM-IEEE International Conference on Computer Aided Design (ICCAD'06)*.