Towards Automating Simulation-Based Design Verification using ILP

Kerstin Eder, Peter Flach, and Hsiou-Wen Hsueh

Department of Computer Science, University of Bristol
MVB, Woodland Road, Bristol BS8 1UB, UK
Kerstin.Eder@bristol.ac.uk, Peter.Flach@bristol.ac.uk,
hsueh@cs.bris.ac.uk

Abstract. This short paper introduces a promising new application area for ILP, namely simulation-based semiconductor design verification, and indicates the potential that ILP has to offer in the context of functional coverage closure.

1 Introduction to Design Verification and Coverage Closure

Verification of industrial designs still relies heavily on simulation; it can take up to 70% of the entire design effort [1]. The increasing complexity of real-world semiconductor designs makes exhaustive simulation prohibitive. [1] In reality, tight time-to-market constraints force verification engineers to be selective with respect to the tests they run to gain confidence in the functional correctness of a design. To keep track of verification progress engineers use functional coverage models which contain four components: a semantic description (story), the list of attributes mentioned in the story, the set of all possible values (domain) for each attribute and a list of restrictions on the legal combinations in the Cartesian product of the attribute domains. The elements in the Cartesian product are referred to as *coverage tasks*. The restrictions of the coverage model identify which coverage tasks are legal and hence need to be covered during verification. Uncovered legal tasks are referred to as *coverage holes*. The task of constructing tests which will cover a hole is called *coverage closure*.

Generating stimulus to increase functional coverage is a key challenge in simulation-based verification. In practice up to 90% of coverage tasks can be reached via *biased pseudo-random tests* which are automatically generated based on a set of user-defined constraints, called directives. However, even supplying the directives requires significant engineering skill and is often only accomplished through many trial-and-error runs. At the end engineers resort to writing *directed tests* by hand aiming to cover the missing cases. Consequently as much as 90% of a verification team's time and resources can be spent on closing the remaining 10% coverage manually.

Figure 1 shows the typical long flat-tailed curve when plotting the coverage rate achieved by random simulation (y-axis) against the number of simulation runs (x-axis). The data for this figure originates from our case study and is representative for many industrial verification projects. It clearly shows that the number of simulations necessary to obtain the last few coverage tasks is excessive in comparison to the number of simulations needed to get most of the coverage. This is one reason why verification has

[1] In most cases the sun would burn out before even a fraction of the tests can be simulated [3].

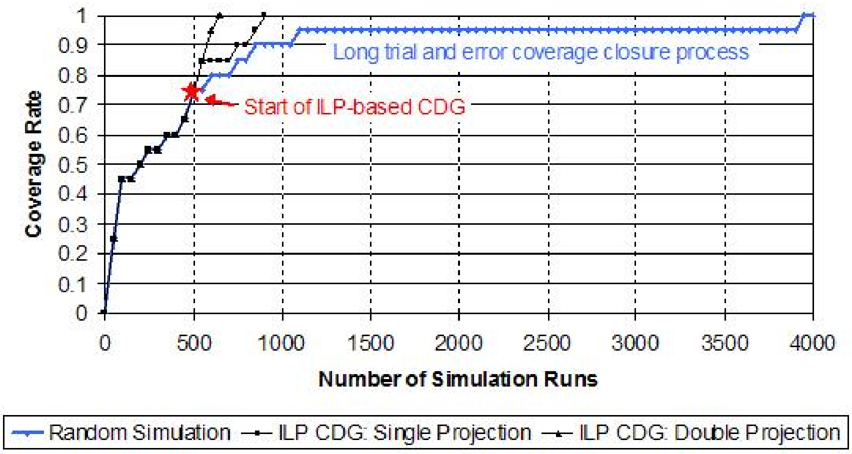

Fig. 1. Coverage Progress for Random Simulation compared to ILP-based CDG.

become the dominant cost in the design process and many verification projects run over time and budget. The latest version of the International Technology Roadmap for Semiconductors [1] calls for more *automation* in the process of functional coverage closure to reach verification targets faster and with less engineering effort.

2 The Test Generation Challenge

The *automatic* generation of the test generation directives is the most demanding aspect of automating functional coverage closure. Coverage-directed stimulus generation (CDG) techniques [2] aim to achieve exactly this. Feedback-based CDG approaches use machine learning in order to automatically generate new directives that bias stimulus generation towards producing tests which target specific coverage tasks. Machine learning techniques employed in this context include Bayesian networks, evolutionary techniques such as genetic algorithms and genetic programming as well as Markov chains. The underlying assumption is that the learning mechanism can identify, from existing tests and coverage, how best to bias stimulus generation such that the resulting new tests can reach outstanding coverage tasks. As a result, the curve in Figure 1 should climb significantly faster than random simulation thus saving a large number of simulation runs and hence verification effort. All existing approaches have shortcomings which have so far prevented them from being used in practice.

3 ILP-based Coverage-Directed Test Generation

This paper proposes a novel CDG technique based on an inductive machine learning method that discovers relational information from structured data. Inductive Logic Programming (ILP) is applied to tests and their related coverage in order to induce general rules which describe the characteristics of these tests. The resulting rules can be used directly as directives, to obtain tests that are structurally similar to the examples presented to the learning system. Coverage closure can be automated by applying rule learning to clusters of a target coverage task (such as a coverage hole) and combining the resulting rules to obtain directives for test generation. As the tests and associated coverage are supplied to the ILP system in a declarative representation, the induced rules are also declarative and in principle human readable. This gives engineers an insight into the knowledge discovered by the ILP system and is also an excellent basis for automatic translation of these rules into test generation directives.

A case study undertaken with Progol demonstrates the fundamental principles of ILP-based CDG in two steps. The first step evaluates the consistency and reliability of the induced directives for existing coverage in a rediscovery experiment. The second step documents the results of the application of our novel cluster-based coverage closure method. To open this interesting application up for the ILP community the data used in the experiment is available on `http://www.cs.bris.ac.uk/~eder/ILP_CDG/`. This site also contains further information on the encoding of the tests and coverage data as well as on the Progol setup including background knowledge and mode declarations.

4 Discussion, Results and Conclusion

In contrast to other machine learning applications where the measure of success is achieving a very high accuracy of the learning output resulting in a large lift when comparing system performance with and without learning, this application is slightly different in that the number of examples to learn from is variable and depends on when the learning is kicked off during simulation. From a machine learning viewpoint, the later in the simulation phase learning is started the more examples are available, hence a higher accuracy can be expected. Conversely, the earlier learning is started the fewer examples are available, resulting in a lower accuracy. From a verification viewpoint, however, the earlier the curve starts to *climb faster than random simulation* the more verification effort can be saved. These two conflicting interests need to be traded off carefully with the verification interests dominating in this context. For example, the lift achieved in our case study, although in machine learning terms not impressive, was good enough to save a significant number of simulations as shown in the two steeper curves in Figure 1.

In conclusion, this paper indicates that ILP-based CDG can make a significant contribution towards automating coverage closure. The declarative representation of both data and learning output gives ILP-based CDG a competitive edge in comparison to other machine learning CDG approaches.

References

1. International Technology Roadmap for Semiconductors, Design Chapter, 2005 Edition. Available at `http://public.itrs.net/`.
2. G. Nativ, S. Mittermaier, S. Ur, and A. Ziv. Cost evaluation of coverage directed test generation for the IBM mainframe. In *International Test Conference*, pp 793–802, Oct 2001.
3. C.-J. H. Seger. An introduction to formal verification. TR-92-13, UBC, Department of Computer Science, Vancouver, B.C., Canada, June 1992.