

Functional Coverage Metric Generation from Temporal Event Relation Graph

Young-Su Kwon and Chong-Min Kyung
VLSI Systems Lab.

Korea Advanced Institute of Science and Technology
GuSong-Dong, YuSong-Gu, Taejon, Korea(ROK)
{yskwon, kyung}@vslab.kaist.ac.kr

Abstract

Functional coverage is a technique which can be used for checking the completeness of test vectors. In this paper, automatic generation of temporal events for functional coverage is proposed. The TERG(Temporal Event Relation Graph) is the graph where the nodes represent basic temporal property and the edges represent the time-shift value between two properties. Hierarchical temporal events are generated by traversing TERG such that invalid, or irrelevant properties are eliminated. Concurrent edge groups in TERG make it possible to generate more comprehensive temporal properties.

1. Introduction

Simulation still remains as the primary technique for functional validation of designs. During the random simulation of the design, pseudo random generators drive the inputs to the design while the outputs are checked by the monitors[1]. Although random simulation can validate the functionality of the design in a short time, it lacks the measure of quality of verification effort. Simulation and formal verification can be combined to provide the coverage of the verification environment[2].

The functional coverage metric is usually a sequence of events that represents temporal properties of the design described in property specification language. The definition of coverage metric for specific application is crucial for the coverage analysis to be successful in verifying the design. In this paper, we propose a novel coverage task generation from TERG(Temporal Event Relation Graph) for functional coverage. We focus on the functional coverage in which coverage task is described by FLTL [2]. In our approach, TERG is used as the specification for the design.

This paper is organized as follows. The TERG and temporal event generation algorithm are presented in section 2.

Experimental results are shown in section 3 followed by concluding remarks given in section 4.

2. Temporal Event Relation Graph

The TERG(Temporal Event Relation Graph) is the graph where the node represents the Boolean logic or basic temporal event and the edge represents the time-shift value between two nodes. The *temporal property generation algorithm* traverses TERG to combine the temporal expressions of the nodes to form longer or more complex expressions hierarchically. Finally, the hierarchically constructed temporal expressions are defined as assertion properties. The node and its outgoing edges are shown in Figure 1. The tem-

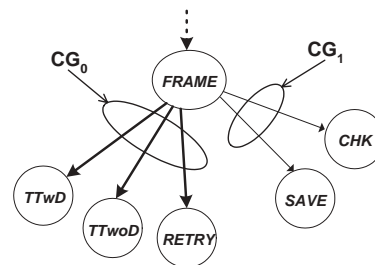


Figure 1. The concurrent edge groups of the node.

poral properties for PCI state machine are explained below. The state *FRAME* can be followed by only one of *TTwD*, *TTwoD* and *RETRY* according to the PCI specification.

The outgoing edges of *FRAME* can be grouped into a set of groups called CG(Concurrent edge Group)'s. In Figure 1, the parent node, *FRAME* has two CG's, i.e., *CG_0* and *CG_1*. A node in TERG may have one or more CG's so that several nodes may be the children of the current node. CG allows several events to occur concurrently after one event has occurred. The advantages of TERG includes the description of

interaction between different state machines as well as valid cross-product exploration. If two nodes are joined to a single node, the validity of the time-shift value for the destination node is checked. The temporal property generation algorithm traverses TERG and generates all possible properties. The algorithm is shown in Figure 2.

```

Ω = 0;
do {
  Ω = Ω ← (Π = cm_gen());
} while (Π ≠ 0);

cm_gen() {
  Π = 0;
  foreach node ni, P(ni) = 0;
  foreach node ns visited in BFS {
    if ns is leaf node, Π ← P(ns) and continue;
    foreach CG G of ns {
      e ← select next edge in G;
      t ← select next time-shift of e;
      nd = dest. node of e;
      if nd was already explored {
        P'(nd) = P(ns) [t] nd;
        if valid, P(nd) = P(nd) & P'(nd);
        else P(nd) = 0;
      } else P(nd) = P(ns) [t] nd;
    }
  }
}

```

Figure 2. Temporal property generation algorithm.

3. Experiments

We have applied TERG property generation algorithm to the “8-channel PCI DMA controller(PDC8)”. PDC8 is DMA controller that transfers data between main memory in PC and eight FIFOs. The verification environment and internal structure of PDC8 is shown Figure 3.

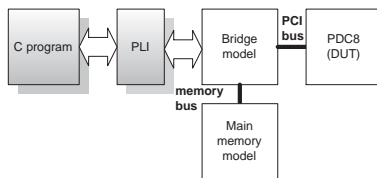


Figure 3. 8-channel PCI DMA controller(PDC8)

We have applied software simulation using VCS in verifying PDC8. We manually described the temporal properties as coverage metrics in VCS before using TERG .

The generated properties are written in OpenVera assertion properties and checked on VCS. We have found that we omitted the temporal properties that are closely related to those two state machines. The manually written temporal properties are not sufficient to disclose all the bugs that are closely related to the temporal properties of several state machines. If we describe all cross-product states, the number of all cross-product states is 222300 states which includes a large number of unreachable states. We described TERG for PDC8 and statistics of properties for PDC8 is shown in Table 1. The number of generated properties is 3423 where all the properties are valid properties according to the specification. This amounts to only 1.5% of 222300 states in the cross-product model. In the experiment using

	# States
# of nodes	18
# of edges	25
# of time-shift values	82
# of generated properties	3423

Table 1. The statistics of properties for PDC8.

TERG , we discovered that automatic generation of temporal properties from TERG makes all possible temporal events. The temporal properties related to bugs caused by the close operation among several state machines were also generated from the proposed coverage metric generation algorithm.

4. Conclusions

In this paper, the novel coverage metric generation methodology is presented based on the proposed TERG which can describe the temporal relation between basic properties. The proposed temporal property generation algorithm traverses the TERG and generates valid temporal properties automatically. The generated properties have hierarchical structure to save the memory consumption by the property checking.

References

- [1] L. Fournier, Y. Arbetman, and M. Levinger. Functional verification methodology for microprocessors using the Genesys test-program generator. In *IEEE Design Automation and Test in Europe*, pages 434–441, 1999.
- [2] J. Ruf, D. W. Hoffmann, T. Kropf, and W. Rosenstiel. Simulation-guided property checking based on multi-valued AR-automata. In *IEEE Design Automation and Test in Europe*, pages 742–748, 2001.