

DECIDER: Test and Verification at the Register-Transfer Level

Jaan Raik, Maksim Jenihhin, Raimund Ubar

[jaan|maksim|raib}@pld.ttu.ee](mailto:{jaan|maksim|raib}@pld.ttu.ee)

Department of Computer Engineering – Tallinn University of Technology – Estonia

<http://ati.ttu.ee/>

Abstract

DECIDER is a tool for test and verification at RTL VHDL. It combines hierarchical and functional fault models reaching high fault coverages for sequential designs and it supports commercial CAD flows.

1. Introduction

DECIDER (DECISION Diagram based genERator) is a tool for Automated Test Pattern Generation (ATPG) at the Register-Transfer Level (RTL). It has been created at Tallinn University of Technology, Estonia. The novelty of the tool lies in utilization of an efficient circuit representation of High-Level Decision Diagrams (HLDD). It combines hierarchical and functional fault models, including dedicated fault models for functional units, multiplexers and comparison operators. The tool has high-level design interfaces to VHDL. At the logic-level it also supports all the main CAD vendors via an EDIF interface. This logic-level interface allows performing fault simulation in order to assess the fault coverage achieved by DECIDER.

The prototype of DECIDER was first presented as a research paper at DATE 1999 [1]. Since then, it has been further developed in numerous research projects. Currently, the application area of the tool is being expanded into the verification field in the framework of the EC 6th framework STREP project VERTIGO (See www.vertigo-project.eu). Experiments comparing DECIDER to available sequential circuit test generation software has shown that it is capable of reaching higher fault coverages, mostly in orders of magnitude shorter run times than its counterparts.

2. Test and Verification Flow

We present a test generation system that operates either on RTL or hierarchically at RTL and gate-level model descriptions. At present, the system implements Design Compiler [2] from Synopsys for logic-level synthesis. In Figure 1, the basic design flow and the place of test generation in it is shown.

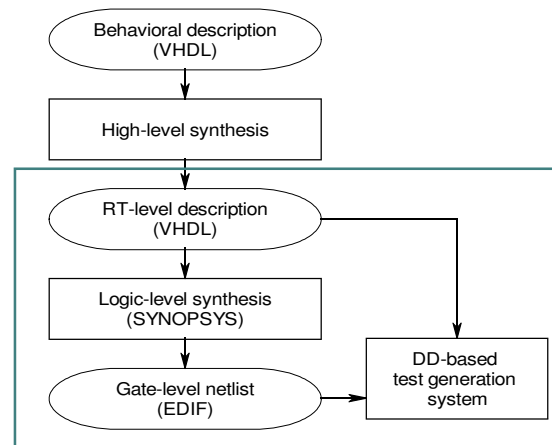


Figure 1: DECIDER : hierarchical test flow

3. High-Level Decision Diagram Model

DECIDER uses an internal system model of High-Level Decision Diagrams (HLDD). Decision Diagrams (DD) have been used in verification for about two decades. Reduced Ordered Binary Decision Diagrams (BDD) as canonical forms of Boolean functions have their application in equivalence checking and in symbolic model checking. Recently, a higher abstraction level DD representation, called Assignment Decision Diagrams (ADD, introduced by D. Gajski [3], have been successfully applied to, both, register-transfer level (RTL) test and verification.

The main issue with the BDDs and assignment decision diagrams are that they allow logic or RTL modeling, respectively. TUT has developed a different decision diagram representation, High-Level Decision Diagrams (HLDD) that, unlike ADDs can be viewed as a generalization of BDD. HLDDs can be used for representing different abstraction levels from RTL to TLM (Transaction Level Modeling) and behavioral. HLDDs have proven to be an efficient model for simulation and fault modeling since they provide for a fast evaluation by graph traversal and for easy identification of cause-effect relationships.

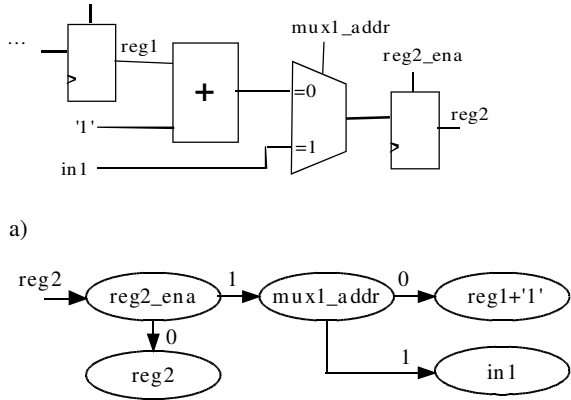


Figure 2: A datapath fragment a) and its HLDD b)

In HLDD models representing digital systems, the non-terminal nodes correspond to conditions or to control signals, and the terminal nodes represent operations (functional units). Register transfers and constant assignments are treated as special cases of operations. When representing systems by decision diagram models, in general case, a network of HLDDs rather than a single HLDD is required. During the simulation in HLDD systems, the values of some variables labeling the nodes of a HLDD are calculated by other HLDDs of the system. Figure 2 presents an example of an HLDD for an RTL datapath fragment.

3. DECIDER Test Generation Concept

The DECIDER algorithm runs in two phases. During the first phase, constraints required to activate test paths in the system are extracted using HLDD models. At the second stage, the constraints are solved relying on traditional solvers. The test generation constraints considered in current paper can be divided into two categories: *path activation constraints* and *transformation constraints*. Path activation constraints correspond to the logic conditions in the control flow graph that have to be satisfied in order to perform propagation and value justification through the circuit.

Transformation constraints, in turn, reflect the value changes along the paths from the inputs of the high-level module under test to the primary inputs of the whole circuit. These constraints are needed in order to derive the local test patterns for the module under test. Both types of constraints can be represented by common data structures and manipulated by common procedures for creation, update, modeling and simulation.

Figure 3 explains the role of these constraints in test generation for a circuit module. In the Figure there are two path activation constraints: $true = f(x_1, x_2)$ and $false = g(x_2, x_3)$. The first one is necessary to propagate the value from the output of the module to the primary output y_3 of the circuit. The latter is required for justification of the first input (D_1) of the module under test. Both these constraints

are extracted from the conditional nodes traversed in the HLDD of the system during high-level path activation. In addition, the Figure presents two transformation constraints. These constraints represent the function for computing the value of the corresponding module input depending based on the values of primary inputs of the circuit.

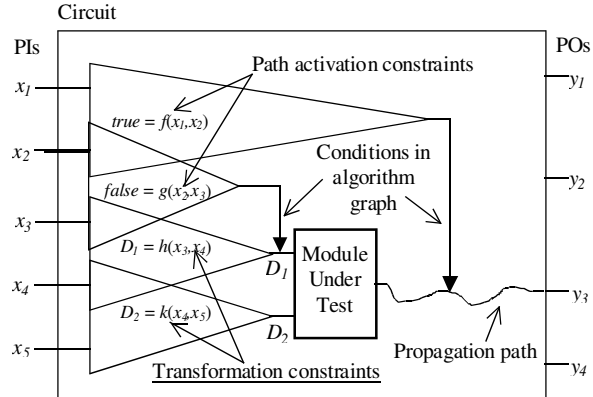


Figure 3: The concept of path activation constraints

Table 1 show comparison to available sequential ATPGs GATEST and HITEC on five HLSynth benchmarks. The results show that DECIDER is very efficient in testing sequential designs. It achieves in average 2.5 % higher fault coverage than GATEST on the given benchmark set.

Table 1: Comparison of stuck-at fault coverage

circuit	faults	HITEC		GATEST		DECIDER	
		F.C., %	time, s	F.C., %	time, s	F.C., %	time, s
gcd16	1754	59.11	365	86.13	190.7	90.95	677.4
mult8x8	2036	65.9	1243	69.2	821.6	74.7	93.7
ellipf	5388	87.9	2090	94.7	6229	95.04	1258.9
risc	6434	52.8	49,020	96.0	2459	96.5	150.5
diffeq	10,008	96.2	13,320	96.40	3000	97.09	453.7
average F.C.:		72.4		88.4		90.9	

4. Conclusion

We presented a HLDD based RTL ATPG DECIDER that can be applied to high-level test and design verification. Experimental results show the efficiency of this tool.

5. References

[1] J. Raik, R. Ubar. Sequential Circuit Test Generation Using Decision Diagram Models, *Proceedings of the DATE Conference*, pp. 736-740, March 9-12, 1999.
 [2] URL: <http://www.synopsys.com>
 [3] V. Chayakul, D. D. Gajski, L. Ramachandran, "High-Level Transformations for Minimizing Syntactic Variances", *Proc. of DAC*, pp. 413-418, June 1993.

Acknowledgement

The work has been supported by EC STREP project VERTIGO (IST 033709) and Development Center ELIKO.