FPGA Design Flow with Automated Test Generation

Karl-Heinz Diener[1], Günter Elst[1], Eero Ivask[2], Jaan Raik[2], Raimund Ubar[2]
[1]FhG EAS IIS Dresden, GERMANY
[2]Tallinn Technical University, ESTONIA

Abstract: *A novel FPGA design flow combined with automated hierarchical test pattern generation was developed and experimented on a real FPGA circuit for telecommunication. A hierarchical test generator for digital systems described in VHDL is presented. Both, register-transfer (RT) and gate level descriptions are used. Decision diagrams are exploited as a uniform model for describing systems at both levels. The method combines bottom-up and top-down approaches to make hierarchical test generation more efficient. It combines RT level deterministic test planning with gate-level local test generation based on deterministic approach at the bottom-up working mode or on random approach at the top-town working mode. Experimental results have shown the advantages of using structural tests generated by ATPG compared to using functional test sequences created by designer.*

1. Introduction

In order to come up with innovative electronic systems in time and with competitive cost, user programmable microelectronic devices (e.g. FPGA) have been used in large scale. Particularly, for small and medium-sized enterprises (SME), FPGA-based solutions seem to be the best way to benefit from the application of microelectronics. That is because FPGA offer the chance to explore the market, i.e. to start with a small volume of project units, and to minimise what can be summed up in the phrase „time-to-market".

To improve the quality of FPGA designs, an Automated Test Pattern Generator (ATPG) is needed. Since traditional gate-level test generation algorithms for complex VLSI systems have lost their importance, hierarchical methods [1-5] are gaining popularity. In this approach, top-down and bottom-up strategies are well-known. In the bottom-up approach, pre-calculated tests for system components generated on low-level will be assembled at a higher abstraction level. Such algorithms typically ignore the incompleteness problem: constraints imposed by other modules and/or the network structure may prevent test vectors from being assembled. Top-down approach has been proposed to solve this problem by deriving environmental constraints for low-level solutions. However, such techniques are of little use when the system is still under development in a bottom-up fashion, or when "canned" local tests have to be applied.

Good results in gate level test generation have been achieved with Binary Decision Diagrams (BDD) [6-8]. Recent research has shown that DDs can be very efficiently used also at higher levels providing a uniform model for both gate and register transfer level test generation [9-10]. In this paper, an ATPG developed at the Tallinn Technical University (TTU) [11] based on using DDs is introduced into the typical FPGA design flow. The method implemented in the ATPG allows to exploit the advantages of both, bottom-up and top-down approaches to create a synergism of the positive features of both strategies, and to adjust easily the ATPG for different technologies. Experimental research has been carried out by using the ATPG for a real telecommunication FPGA chip, a Huffmann encoder developed in Dresden Branch lab for Automation of Circuit and System Design (EAS) of the Fraunhofer Institut fuer Integrierte Schaltungen (IIS).

2. A brief description of the FPGA design flow

Generally, the FPGA development cycle can be interpreted as a sequence of the following design steps (Fig. 1):

- working out the system specification as a behavioral description i.e. a VHDL system model (in most cases on algorithm level or register transfer (RT) level) and simulation patterns (1,2,3,14);
- working out a synthesizeable description (VHDL system model) with using methods and tools for modeling, simulation and analysis (4,5);
- synthesis by using methods and tools - high level synthesis, logic synthesis (8,9,21);

- partitioning, place and route with using tools for layout generation, back annotation and simulation (22);
- programming the FPGA (23);
- testing by a low-cost tester or logic analyser (24).

Each design step starts up from a system model given as representation (fitted to) appropriate to different abstraction levels to be passed through within the design process, and it ends up with a model of an implementation. Based on that, the next design step can be executed, and will lead to a more refined model of implementation, and so on.

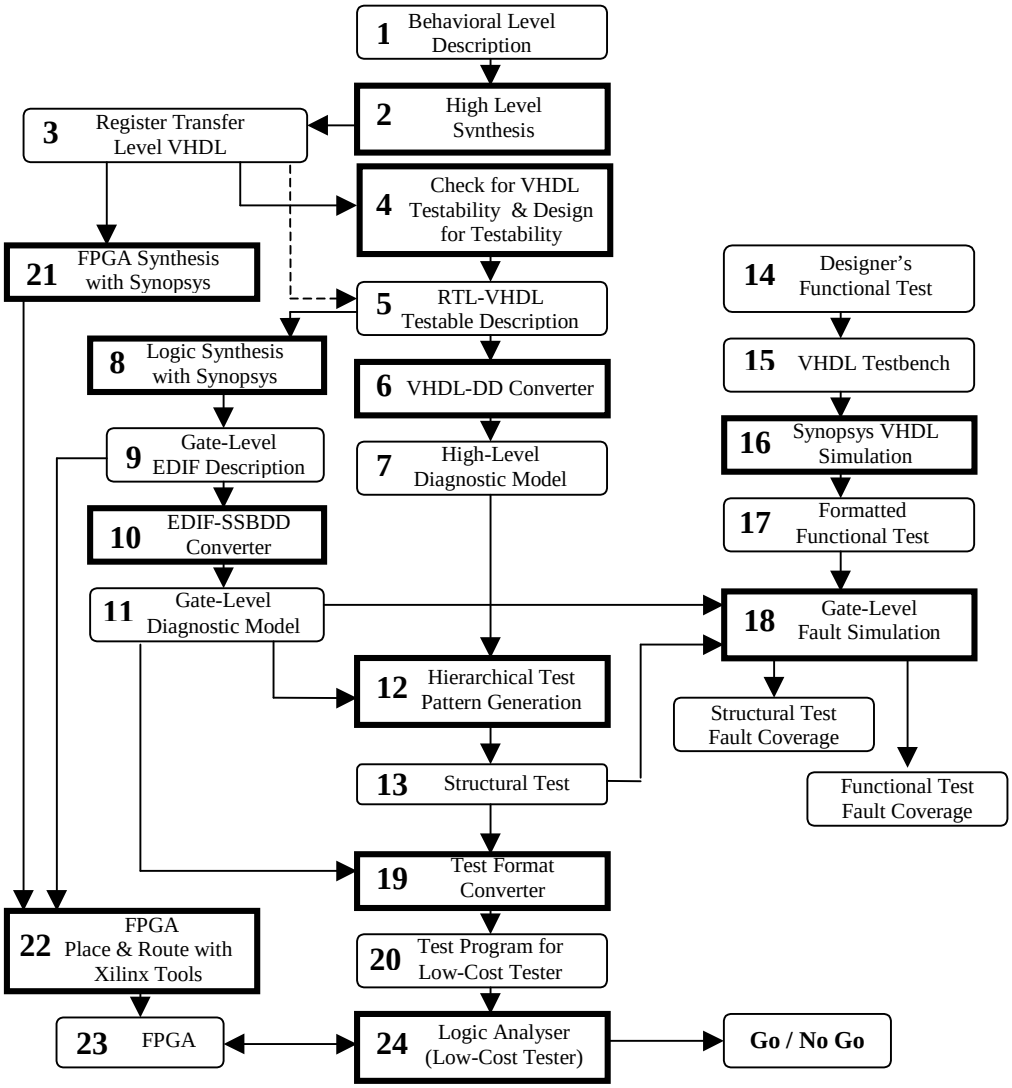

Fig. 1. The combined FPGA Design and Test Flow

A lot of designers' experience is necessary to make the high level system description (e.g. given as VHDL model) ready for synthesis and testability.

The solution of the test problem (i.e. design for testability, test pattern generation) has to be involved in the design cycle from the very beginning. Usually, designers develop simulation patterns in parallel with the implementation to be achieved as result of each design step. These so-called functional test patterns are primarily intended to make sure the IC functionality, however, they have to be elaborated „by hand". On the other hand, the fault coverage of functional test patterns may not be sufficient, as these patterns are not targeted to test the structure, i.e.stuctural faults.

3. Description of the automatic test pattern generator

In the ATPG, a method which allows to implement both, bottom-up and top-down approaches at the uniform basis is used. In the bottom-up approach, the transparency features will be exploited to build

up transparent I(F)-paths for assembling library patterns for components. For different technologies, dedicated test patterns for components can be used. If this approach will fail (no transparent path can be activated), we switch over to the top-down approach for extracting high-level constraints with the goal to be considered when deriving tests for components at the lower level.

Decision diagrams (DD) [9-10] are used in the paper for a joint description of structural properties, functions, faults and fault propagation modes of digital systems. Each node in the model has a strong relation to the structure of the control and datapaths: nonterminal nodes represent the control logic in modules (subnetworks), the nodes labelled by data variables represent buses, and the nodes labelled by expressions represent data manipulation logic. In such a way, to generate a test for a given node means to create a test for a related structural part of a module or subnetwork. A technique is used for inserting fault propagation modes into the DD-model as an additional information to the description of functions.

To generate a test for a module in a network, we have to propagate the fault effects from the output of the module up to the primary outputs of the network as well to propagate the test stimuli from the primary inputs of the network up to the inputs of the module. All these procedures are carried out on DDs extended by transparency modes.

For the top-down approach, a novel method to combine random and deterministic techniques is used. First, high-level symbolic test frame will be created by a deterministic search. This frame will be used in the role of a filter between the random TPG and the component under test. If the filter does not allow to find a random test with 100% fault coverage for the component, another test frame will be created. In such a way, the following main parts in the ATPG are used alternatively: deterministic high-level test frame generator, random low-level test generator, high-level simulator for transporting random patterns to the component under test and low-level fault simulator for estimating the quality of random patterns.

4. Additional interfaces needed for including the ATPG into the design flow

The combined FPGA Design and Test Flow is based on using the tools which are shown in Fig.1 by bold blocks. The ATPG (12) is based on hierarchical algorithms which use Structurally Synthesized Binary Decision Diagrams (SSBDD) at the lower level, and RTL Decision Diagrams (DD) at the higher level [10].

To embed the ATPG into the existing design flow, the interfaces between design tools and test generation/fault simulation tools with two coresponding converters have been created: the EDIF-DD converter (10) from EDIF netlist (9) of gate-level circuits to the low-level SSBDDs (11), and the VHDL-DD converter (6) from RTL level VHDL descriptions (5) to the high-level DDs (7). The converter 19, which makes possible testing with low level test equipment, has been developed in a cooperation between TTU and EAS IIS Dresden. In the present moment the VHDL-DD converter is oriented to a VHDL subset used as output format of a High Level Synthesis System [12]. The gate level fault coverage for both structural and functional tests (14,15,16,17) was graded by gate-level fault simualtor (18).

5. Experimental research on proving the motivation of using the ATPG

As design examples to check the efficiency of using the ATPG, the Huffmann encoder circuit developed in EAS IIS Dresden, a complex multiplier circuit *mult8x8*, and two well-known international high-level synthesis benchmark circuits *diffeq* and *gcd* were chosen. The Huffmann encoder is a part of the sender side of a video signal transmission system. The register-transfer level VHDL description of the encoder was synthesized by a high-level synthesis tool from Univ. Tuebingen, C-LAB Paderborn and Univ. GH Paderborn. The description was further applied to logic synthesis with SYNOPSYS Design Compiler and FPGA placement and routing with XILINX software. The resulting FPGA had a complexity of about 1300 gates.

To show the real need of developing a dedicated compact structural test for a FPGA design, the quality and fault coverage of the designer's functional test for the Huffman Decoder was investigated for comparison against the automatically synthesized structural tests.

For this experiment, at first, the procedure of logic synthesis by Synopsys CAD tools (8) for the design (3) created at the EAS Dresden, was carried out. Then, a gate-level diagnostic model (11) of the logic design in the form of SSBDDs was generated by the converter (10). Finally, the fault simulation experiment (blocks 14-18) was carried out. A Test Format Converter (19) was needed for transforming

the test sequence produced by the ATPG into the test program for the logic analyser to carry out the test experiments on a low-cost tester (logic analyser),

Experimental results on structural test generation are available in the present moment for three highly sequential circuits (with global feedback loops embracing the control and data paths) shown in Table 1. The experiments showed that the DD based test generation algorithm runs (on the example of *gcd*) an order of magnitude faster than previously published approaches [13]. The functional test of the Huffmann encoder with a length of 3,5 millions of patterns led to a fault coverage of 61,5% only. The results of automated test pattern generation for the Huffmann encoder will be available in few weeks.

Table 1. Test Generation Results of the ATPG

Circuit	Gates	Faults	PIs	POs	FFs	Control states	Fault cover %	Test length	Time s
Functional test									
Huff.enc.	~1300	5336	31	30	118	27	61,5	9658*	-
Structural tests									
Huff.enc.	~1300	5336	31	30	118	27			
diffeq	4195	15836	81	48	115	6	95.4	3277	20.4
mult8x8	1058	3975	17	16	95	8	79.5	2846	14.8
gcd	227	844	9	4	15	8	91.0	924	5.6

* Fault simulated part of the total 3,5 millions patterns (the last 5000 patterns gave only 1%)

6. Conclusions

Experiments with a FPGA chip for telecommunication have shown that functional tests developed based on designers' experience by hand are tainted with two essential drawbacks: they are much longer than the automatically synthesised structural tests, and generally, functional tests do not offer a sufficient fault coverage. The new approach tends to a drastic reducing of test cost and brings out a remarkable progress to control (master) the test problem.

Acknowledgements

The research has been supported by German-Estonian bilateral project EST 008-96 funded by BMFT Germany and by a grant No 1850 from Estonian Science Foundation.

References:

[1] D. Bhattacharya, J.P. Hayes, A hierarchical test generation methodology for digital circuits, JETTA: Theory and Application, vol. 1, pp. 103-123, 1990.

[2] M.Karam, R.Leveugle, G.Saucier, Hierarchical test generation based on delayed propagation, IEEE 1991 International Test Conference, pp.739-747.

[3] J.Lee, J.H.Patel, Hierarchical test generation under intensive global functional constraints, 29th ACM/IEEE Design Automation Conf., June 1992, pp. 261-266.

[4] H.-D. Hümmer, W. Geisselhardt. Hierarchical Test Generation with FunTest IC and DUST. Proc. 1st IEEE Workshop on Hierarchical test Generation, Blacksburg, 1993.

[5] U. Gläser, H.T. Vierhaus. "FOGBUSTER: An Efficient Algorithm for Sequential Test generation". IEEE EURO-DAC'95, Brighton, 1995.

[6] R. Ubar, Beschreibung digitaler Einrichtungen mit Alternative Graphs für die Fehlerdiagnose, Nachrichtentechnik/Elektronik, 30, H.3, 1980, S. 96-102.

[7] R.E.Bryant, Graph-based algorithms for Boolean function manipulation. IEEE Trans. on Computers, Vol.C-35, No8, 1986, pp.667-690.

[8] B.Becker, R.Drechsler, How many Decomposition Types do we need? European Design and Test Conf., Paris, 1995, pp. 438-443.

[9] R.Ubar, Test Synthesis with Alternative Graphs. IEEE Design & Test of Computers, Spring 1996, pp.48-57.

[10] R.Ubar. Combining Functional and Structural Approaches in Test Generation for Digital Systems. Micro-electronics and Reliability. Elesevier Science Ltd. Vol. 38:3, pp.317-329, 1998.

[11] G.Jervan, A.Markus, J.Raik, R.Ubar. DECIDER: A DD-Based Hierarchical Test Generation System. Proc. 2nd Int. Workshop on Design and Diagnostics of Electr. Circuits. Szczyrk, Poland, Sept. 2-4, 1998, pp.269-273.

[12] G.Jervan, A.Markus, J.Raik, R.Ubar. VHDL Based Test Generation System. *Proceedings of the 5th Electronic Devices and Systems Conference*, Brno, June 11-12, 1998, pp.145-148.

[13] E.M.Rudnick, R.Vietti, A.Ellis, F.Corno, P.Prinetto, M. Sonza Reorda. Fast Sequential Circuit Test Generation Using High-Level and Gate-Level Techniques., Proc. of the DATE Conf, 1998