

Quality Estimation of Test Vectors and Functional Validation Procedures Based on Fault and Error Models

T. Riesgo, Y. Torroja, E. de la Torre, J. Uceda

Universidad Politécnica de Madrid

División de Ingeniería Electrónica

c/ José Gutiérrez Abascal, 2; E-28006 Madrid (Spain)

Tel. +34-1-411 75 17; Fax. +34-1-564 59 66

e-mail: teresa@upmdie.upm.es

Abstract. This paper presents a method to estimate the quality of a set of test vectors and the validation procedures from pre-synthesised descriptions in VHDL. The method is based on the definition of fault models, for test features evaluation, and error models, for quality validation estimation.

1. Introduction

In current top-down methodologies [1], there are some tasks that need further development. Test is one of them, as the designer works at the architectural level, where there is no possibility of estimate the fault coverage achieved with a set of test vectors. Only test synthesis techniques [2] can be applied, and there are cases which do not allow this type of approach (for economic reasons or performance losses).

Another important issue to face high complexities is design validation. A lot of work is being done to reduce the amount of simulations made at the logic level by the use of formal verification methods [3]. However, to assure that the initial description is correct a number of functional simulations must be run at the architectural level. With the current complexities it is very difficult to assure that the functionality of the circuit is sufficiently checked.

The final goal of the work is to have a common simulation environment that helps the designer in the earlier stages of the design process, in particular, before the synthesis process. The tasks that will be faced in this work are those related to test vector definition and evaluation, and functional validation, considering:

- * The use of these tools is integrated in a top-down design methodology, as it considers synthesizable descriptions written in VHDL [4] as input.
- * With a common code perturbation scheme and coverage evaluation, both problems can be addressed by changing the fault model and the test vectors into an error model and functional validation stimuli.
- * The fault model has to be defined for achieving a maximum correlation between the VHDL modelled faults and the logic level faults in the synthesised structure.
- * The error model has to be defined in such a way that high error coverage implies a good quality functional validation of the description.

The common point between the test and functional validation issues is the way the VHDL code will be perturbed and the simulation environment for fault and

error simulation, without having confusion between both concepts.

2. Environment for test and design validation

The environment presented for design validation and test is based on perturbing the original VHDL descriptions by means of a fault model, for testing purposes, and an error model, for design validation.

The fault and error models are defined as subsets of the possible perturbations that can be inserted in a synthesizable VHDL description. These perturbations will be inserted in the VHDL original description and will generate a set of perturbed or mutated codes, that will contain the behaviour of the description under fault or under error. Both fault and error models are based on single perturbations with local scope: there is a single element of the description under fault/error, and the effect will be injected in just one code line. The error and fault models depend on the affected language element: data, expressions and statements. The perturbation classes used are: *stuck*, which makes that the affected element takes a constant value, *switch*, that assumes that the affected element can be changed to another one of the same type, and *dead*, that produces that the affected statement does not execute. The detailed descriptions of the models can be found in [5], [6].

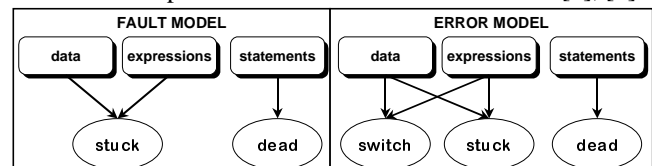


Figure 1. Fault and error models

The fault and error models are defined for behavioural descriptions, although hierarchical elements can be handled considering the stuck perturbations in the interconnections.

The main feature of the fault model consists on the consideration of the synthesis process in its definition. Thus, the array types fail element by element independently and in integer data types its binary implementation is considered. This method has two advantages: the accuracy is better and the number of faults is smaller.

In the case of the error model, only those perturbations that are more likely to occur are considered. Thus, to reduce the size of the error space, singular values of the data elements are considered as target values for the errors.

3. Evaluation of the models. Experimental results

A prototype of a VHDL error and fault simulator has been implemented based on a commercial VHDL simulator. This experimental tool can be used for both applications, test and design validation, just changing the fault and error models and a few minor details.

The fault model has been checked with a set of examples, to verify that the fault coverage achieved with a set of test vectors on the synthesizable description is correlated with the logic level fault coverage with the same test vectors, on the synthesised structure. The results shown in table 1 are expressed as the average error and the correlation coefficient, which indicates the relation between data series, the VHDL and the logic fault coverages.

name	VHDL faults	logic faults	description	average error	correl. coeff.
alu	302	1481	8-b ALU/8 operat.	12.0 %	0.956
		2160		12.9 %	0.944
		2530		14.3 %	0.935
alu16	493	6662	16-b ALU/8 operat.	11.9 %	0.933
		7164		10.9 %	0.929
		7174		10.5 %	0.932
multis	503	3244	seq. multiplier (8 b)	7.7 %	0.985
shreg	66	480	shift register (8 b)	3.1 %	0.907
subst	586	2621	ALU + FSM	10.3 %	0.974
tcon16	208	1146	16-bit counter with test logic	8.4 %	0.961
trarec	223	1468	reception-transmission unit	20.3 %	0.998

Table 1. Fault model evaluation

From this table, it can be observed that the correlation coefficients are, for all the examples, greater than 90%. This implies that the incremental fault coverage follows the same tendency at both levels. In some cases, there are important average errors although the most important errors are at the lower levels of fault coverage, while at higher fault coverages (>80%) the error decreases. Another important issue is that we have observed that those parts of the description with lower fault coverages are common in the pre-synthesised code and in the logic level structure. Therefore, this method can also be used for detecting low testability areas at the early stages of the design process.

For evaluating the quality of the functional validation estimation based on the defined error model, a set of simulations have been run on VHDL examples whose functionality is well-known. The goal was to check that a “good” set of functional stimuli produces high error coverage, and that an incomplete set of functional stimuli produces lower error coverages and the non-detected errors have information about these functions that have not been sufficiently checked. Table 2 shows the results obtained for the examples. It can be observed that the number of modelled errors is low and the error coverage high because the functional stimuli were appropriate selected. The non-detected errors contain information about the functions that have not been checked or about undetectable errors.

name	# code lines	# errors	detected	error coverage
alu	32	120	118	98.3%
alu16	32	120	118	98.3%
cont16	11	46	46	100.0%
fsm4	30	92	92	100.0%
multis	30	156	151	96.8%
shreg	9	46	45	97.8%
tcon16	57	260	252	96.9%
trarec	45	156	152	97.4%

Table 2. Results for the error model evaluation

4. Conclusions

This paper has presented a common simulation environment for estimating the quality of test vectors and functional validation procedures, for synthesizable VHDL descriptions. The method is based in the definition of fault and error models that perturb the VHDL code.

The most important advantage in the fault model definition is that the synthesis process is taken into account to give the test system more accuracy when estimating the achieved fault coverage. The fault model has been evaluated with a set of examples, showing a high correlation between the low and the high-level fault coverages.

For the error model, the main goal is to have a manageable model that allows the designer knows how good are the input stimuli applied to the VHDL code and if they are enough to verify the code functionality. Only singular values will be considered as errors to make the evaluation system more efficient. The error model has been evaluated with a set of examples whose functionality was well known. The results show that a “good” set of input stimuli achieve a high error coverage and “incomplete” sets of functional stimuli show lower error coverage and the list of non detected errors have information about those parts of the code that have been incompletely checked.

These two quality estimation approaches can run in a single implementation of an error/fault simulator just changing the error and fault model.

5. References

- [1] PRENDA, *ASIC Design Methodologies*, edited by UPM/DIE (Spain), November 1996.
- [2] B. Bennetts, “Test Synthesis: Towards Higher Levels of Abstraction”, *Proc. 2nd Archimedes Workshop on Synthesis of Testable Circuits*, Barcelona, February 1995.
- [3] P. Camurati, P. Prinetto, “Formal Verification of Hardware Correctness: Introduction and Survey of Current Research”, *IEEE Computer*, pp. 8-19, July 1988.
- [4] IEEE, *IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-1993*, IEEE, 1993.
- [5] T. Riesgo, J. Uceda, “A Fault Model for VHDL Descriptions at the Register Transfer Level”, *Proc. of EURO-DAC with EURO-VHDL’96*, Geneva (Switzerland), September 1996.
- [6] T. Riesgo, Y. Torroja, C. López, J. Uceda, “Estimation of the Quality of Design Validation Experiments Based on Error Models”, *Proc. of VHDL Users’ Forum in Europe*, Toledo (Spain), April 1997.