ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2007. No. 2(74)*
ELEKTRONIKA IR ELEKTROTECHNIKA

T121 *SIGNAL TECHNOLOGY*
SIGNALŲ TECHNOLOGIJA

Testability Analysis of the VHDL Structure for Fault Coverage Improving

V. I. Hahanov, M. A. Kaminska, O. Lavrova
DA Department, Kharkov National University of Radio Electronics,
Lenin Avenue 14, Kharkov, Ukraine, 61166, tel.: (380) 57-70-21-326, e-mail: maryna4329@kture.kharkov.ua

Introduction

Testability is one of the most important factors that are considered during digital devices design along with reliability, speed and the cost. The low level of device testability leads to increase of number of non-tested faults and verification time at design, production and operations stages. Therefore, the cost of diagnostic (a degree of faults concentration) decreases essentially during techniques of testability design. Testable design methods: 1) structural analysis of the device, calculation of the controllability, observability, testability values which could be used on the design phase; 2) ways of testable circuit structural design, which used BIST logic (for example, boundary scan cells [1], [2]) to provide access to internal connections in circuit. Hence analysis of testability needs to be done at earlier level of device description. This is the main reason of development of the method of testability analysis at the system (algorithmic) level.

The *purpose of the work* is the essential time decreasing of verification, tests synthesis and/or increasing of a degree of faults covering by using testability values inside circuit under test. *Object under test* – digital device, which described on hardware description language and presented as oriented graph. For main goal achievement it is necessary to solve following *tasks:* 1. Digital circuit model description in VHDL (or with other HDLs) and represent it as oriented graph. 2. Calculation testability values (controllability and observability) 3. Separating of the test and functional procedures. 4. Verification and testing of proposed method.

Methods of digital systems design

Analysis on the high-level circuit's description increases possibilities of the designers, efficiency of test procedure is increasing, and information transmission process between logic blocks is improving. Important moment here is development of the new technologies, based on specifications, and new test standards elaboration for open new possibilities on the world market. Providing of the functional correctness on the register transfer level is one of the most difficulties for SoC and ASICs designers. Main goal for designers is decreasing of the circuit verification time, which needs in new design and verification technologies. It is recognized two types of verification – dynamic (fault simulation) and static or formal verification (using of the assertions) [3]. Static verification allows to make obvious of many properties by structural analysis of the device on register transfer level. Such tools are used for exhaustive model verification. By efficient using of the verification technologies is possible to check practically all inaccuracies in project on early stage of designing.

Traditionally, engineers execute verification procedure using Black-box verification, in other words, engineer creates project model written on hardware description language, for example, Verilog, VHDL. After that, engineer creates a testbench, which allows generating test vectors. Such approach includes test generation, checking of output reaction, and fault coverage analysis.

Alternatively, White-box testing with using of assertions (or additional scan logic) in code or circuit structure is more popular design technology. As a result, internal lines in the device under verification will be more observable and controllable. In addition to assertions, it is possible to use other AD-HOC technologies (additional synthesizable scan logic in circuit's bottlenecks, which allows separating scan mode and normal functioning mode and having to be transparent in normal functioning mode). Such approach allows significantly improve fault coverage, as well as product quality and accelerate market entry, but it requires additional area overhead. To understand of necessity of adding such additional scan logic, controllability, observability and testability definitions have to be considered.

Observability – ability to observe output response on the input stimulus (tests) which have to be generated by testbench, in other words it is possibility to observe the result of influences on input or internal lines of code or structure. In this case testbench can be considered as tool for "restricted" observability, i.e. observability only for external ports of device or model.

Controllability – ability to stimulate (activate) internal lines in code or in project structure. In this context following parameters could be considered: line coverage, branch coverage, path coverage, and expression coverage [3]. *Line coverage* measures the number of times a particular line of code was executed (or not) during a simulation. *Branch coverage* measures the number of times a section of code diverges (IF, CASE operators, etc.) into a unique flow. *Path coverage* measures the number of times a unique path through the code (including both statement and branches) is executed during a simulation. *Expression coverage* measures controllability of the individual variables, which contribute to the expression's output value.

In this paper controllability and observability of external and internal nodes in program code, and this code on VHDL is represented as oriented graph will be considered. Based on the obtained values, check points have to be selected and additional logic have to be added to these points. Bottlenecks (internal lines of oriented graph with worst values of controllability and observability), which have been selected, have to be additionally verified. Such procedure could be executed by every engineer or developers, even developer, who starts his work in project later than project was started, because testability analysis and bottlenecks selecting executes on the internal device model. In this case, engineer or developer executes only testability analysis and prepare easy procedure of bottlenecks selecting for add scan logic. Instead of additional scan logic, using of assertions doesn't give exact algorithm of bottlenecks selecting, and only programmer knows, where assertions have to be added. If this programmer will leave project, other developer will have to understand developed code and "feel deeply" how this project works and where assertions have to be added. That pushed to the development of a new method of the testability analysis and purposeful selecting of bottlenecks in device for adding scan logic.

Testability analysis methods and structural analysis of the circuit

Methods of testability analysis were developed for different levels of abstraction, depending on trends in designing tools. Most famous methods on the gate level are CAMELOT, Peterson method, SCOAP, TADATPG [4], [5], [6], [7]. These methods are oriented on structural analysis of the device and using of the deterministic test. Thus, in Peterson method testability values are calculated on the circuit structure, represented as oriented graph. Later, Parker-McCluskey method was developed [8]. This method and further approaches, based on this method are used on the gate level and RTL. Now, actual issues for EDA market is testability analysis on system level or register transfer level.

Thus, in [9] was offered technology of the testability improving on the behaviour level. Analysis algorithm based on the initial program code modification. In [10] proposed method of the testability analysis, which executed during synthesis procedure. In [11] presented method of the testability analysis on the system level, which based on the interaction of interaction VHDL constructions and their implementation into the circuit structure. Such approach could be used for post synthesis structure. In this paper method of testability analysis is proposed, which could be used mainly on the system level and also, on the register transfer level and gate level. Device, which described on VHDL, should be presented as oriented graph.

Method allows executing easy analysis of graph structure and providing it modification for improving of it testability before synthesis. Testability analysis has to be provided on all design stages. At that, most adequate analysis is conforming to gate level structure as more detailed circuit representation. Nevertheless, analysis on the high level abstraction, where project model represented as structure of interconnected components allows to execute testability analysis with minimal computational efforts. Obtained testability values and using of the boundary scan technologies can influence on the costs of the diagnostic assurance and assistance (timing and costs efforts for test synthesis, fault simulation, defect diagnosis for each design stages).

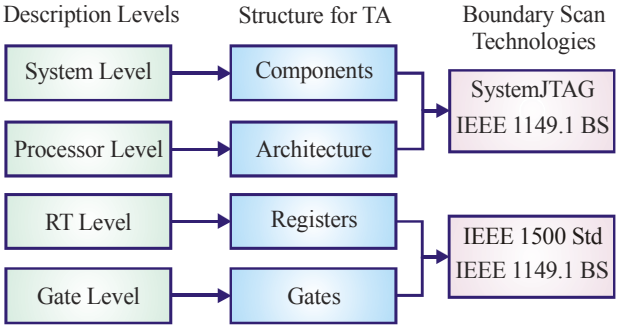

Fig. 1. Design stages of digital devices

Each level of hierarchy has own name and base structural elements. Logic elements And, Or, Not, Xor and others are use for gate level. Registers, counters, multiplexers, ALU, decoders are use for RT level. System level is represented by group of interconnected components. Here project model on system level is described as oriented graph. Main difficulty of test procedure is lying in necessary to test functionality of all functional blocks (such blocks can be described incorrect). Unreachable states, deadlock conditions, operators if, case, loop also have to be checked. Proposed method consists of controllability, observability and testability values calculation. Method could be used on the device structure before synthesis (circuit, described on hardware description languages), on register transfer level (post synthesis structure), gate level. Method based on probabilistic approach of testability calculation of each node in circuit. Equipotential lines in circuit are used for identify node for gate level; interconnections between logic blocks for RT level; and signal lines between logic operators in VHDL code.

Controllability calculation.

Values of controllability on the primary inputs are equal to 1 (because primary inputs of device could be directly setting in each logic value, so controllability of such nodes is total-lot). Values of controllability are calculated from primary inputs to primary outputs.

Practically, most values of the controllability are situated between the limits of range [0; 1]. Value of the controllability of the line on rank r is depend of the values of controllability on the rank r-1 and coefficient of the controllability transfer (K(0), K(1)), that defined by the logic function of the logic block.

Coefficient of the controllability transfer, that defined by the logic function (cubic coverage) of the gate (*K(1)* – for setting of logic one on the output of the gate, *K(1)* – for setting of logic zero on the output of the gate).

But when device was described by cubic coverage, it is necessary to redefine value 'X' to avoid problem of simulation. Values of controllability are calculated by following formulas:

$$C^1(X_{i+1}) = 1 - K(1) \cdot \prod_{i=1}^{m} C^1(X_i); \qquad (1)$$

$$C^0(X_{i+1}) = 1 - K(0) \cdot \prod_{i=1}^{m} C^0(X_i); \qquad (2)$$

where n(1), (n(0)) – number of vectors, which allows to set output of logic block in logic one (logic zero); m – number of inputs in logic block. In case of reconvergent fun-outs (fig. 2), formulas of controllability calculation are following:

$$C^0(X_{i+1}) = 1 - K(0) \cdot \prod_{i=1}^{m} C^0(X_i) \cdot 2^{k+1}; \qquad (3)$$

$$C^1(X_{i+1}) = 1 - K(1) \cdot \prod_{i=1}^{m} C^1(X_i) \cdot 2^{k+1}; \qquad (4)$$

where k – number of reconvergent fun-outs.

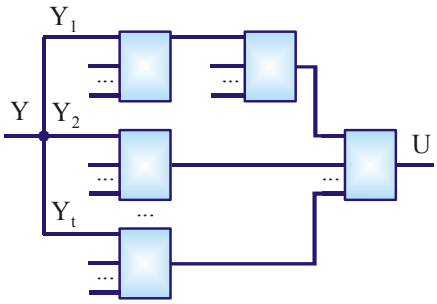

Fig. 2. Case of reconvergent fun-outs

Values of observability and testability it is proposed to calculate as in [7], and also use method of circuit modification as in [12].

Example.

Let's consider the circuit, described on VHDL:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
entity karpati is
    port(
    a,b,c,d,e,f: in signed (1 downto 1);
    x,y: out  signed(2 downto 0));
end karpati;
architecture karpati of karpati is
begin
x <= (b*d)+(a*e);
y <= (a*f)+(c*d)/(a*f)+(b*d);
end;
```

Oriented graph is following:

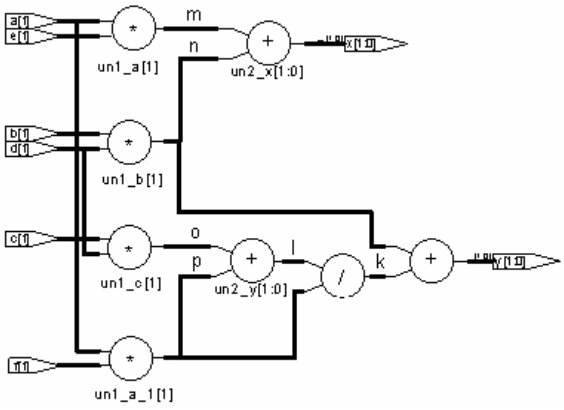

Fig. 3. Circuit under test

Values of the controllability, observability and testability are shown in Table1.

Table 1. Testability values

Line	$C^0(x)$	$C^1(x)$	O(x)	$T^0(x)$	$T^1(x)$
a	1	1	0,465	0,465	0,465
b	1	1	0,45	0,45	0,45
c	1	1	0,86	0,86	0,86
d	1	1	0,66	0,66	0,66
e	1	1	0,25	0,25	0,25
f	1	1	0,68	0,68	0,68
m	0,25	0,75	0,25	0,187	0,563
n	0,25	0,75	0,46	0,135	0,405
o	0,25	0,75	***0,86***	0,035	0,105
p	0,25	0,75	0,68	0,08	0,24
l	0,67	0,333	0,813	***0,939***	0,875
k	***0,835***	***0,945***	0,25	0,876	***0,959***
x	0,67	0,333	1	0,67	0,333
y	0,953	0,911	1	0,959	0,991

Analysis on the post synthesis structure will be complicated due to appearance of additional signal lines (due to increasing of operand's capacity).

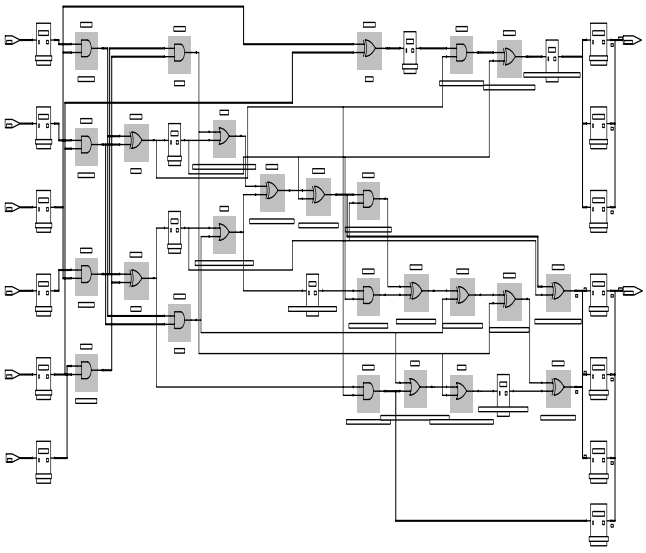

Fig. 4. Structure after synthesis (gate level)

Conclusions

The method of testability index calculation is developed. It is also supposed the way of code (or circuit structure, if testability analysis will be executed on the gate level or RTL) modification with aim of testability improving and annihilation of non-tested paths and nodes. This is scientific novelty of the current research.

Practical significance and advantages are: 1) Arising of quality of faults covering (10-30 %) of existing test with small hardware-controlled expenditures (up to 20 %). 2) Time decreasing of tests synthesis by separation of testing and functional procedures. 3) Spending of device analysis on the earliest design stages and increasing of Yield Ration.

This method could be used on the gate level analysis, which gives more exact indexes for the further scheme modification than at higher levels of device description.

References

1. **IEEE Std 1149.1-2001,** IEEE Standard Test Access Port and Boundary-Scan Architecture. – New York, 2001. – 208 p.
2. **IEEE P1500/D11,** Draft Standard Testability Method for Embedded Core-based Integrated Circuits. – New York, 2005. – 138 p.
3. **Foster H., Krolnik A., Lacey D.,** Assertions-based Design.– Kluwer Academic Publishers, 2003. – 392 p.
4. **Goldstein L. M. and Thigen E. L.** SCOAP: Sandia Controllability/Observability Analysis Program // Proc. 17th Design Automation Conf, 1980. – P. 190–196.
5. **Bennetts R. G., Maunder C. M. and Robinson G. D.** CAMELOT: A Computer-Aided Measure for Logic Testability // IEEE Proc., 1981. – Vol. 128, Part E, No. 5. – P. 177–189.
6. **Spillman R., Glaser N., Peterson D.** Development of a general testability figure-of-merit // IEEE International conference of Computer-Aided Design, 1983. – P. 34–35.
7. **Кулак Э. Н., Каминская М. А.** Модификация цифровых схем с использованием метода анализа тестопригодности TADATPG // Радиоэлектроника и информатика, 2005. – № 3. – С. 113–119.
8. **Parker K. P. and McCluskey E. J.** Probabilistic Treatment of General Combinational Networks // IEEE Trans. on Computers, 1975. – Vol. C-24, No. 6. – P. 668–670.
9. **Larsson E., Peng Z.** A Behavioral-Level Testability Enhancement Technique // IEEE European Test Workshop.– Constance, Germany, 1999.
10. **Flottes M. L., Pires R., Rouzeyre B.** Analyzing Testability from Behavioral to RT Level // Proc. European Design&Test Conf., 1997. – P. 158–165.
11. **Zdenek Kotasek, Richard Ruzicka, Josef Strnadel, Jan Hlavicka.** Interactive Tool for Behavioral Level Testability Analysis // Proceedings of the IEEE ETW2001.– Stockholm, Sweden, 2001. – P. 117–119
12. **Кулак Э. Н., Каминская М. А.** Модификация цифровых схем с использованием метода анализа тестопригодности TADATPG (часть 2) // Радио-электроника и информатика, 2005. – № 4.– P. 60–68.

V. I. Hahanov, M. A. Kaminska, O. Lavrova. Testability Analysis of the VHDL Structure for Fault Coverage Improving // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – № 2(74). – P. 29–32.

Method of digital device testability analysis, which is represented on the system level (VHDL description) for verification and test synthesis tasks simplification for fault coverage improvement on the given test patterns is offered. Method is based on the topological analysis of circuit, which is represented as RTL blocks and circuit's further modification by separation of testing and functional procedures for testability improving and testing procedure simplification. Ill. 4, bibl. 12 (in English; summaries in English, Russian and Lithuanian).

В. И. Хаханов, М. А. Каминская, О. Лаврова. Повышение качества теста на основе анализа тестопригодности VHDL кода // Электроника и электротехника. – Каунас: Технология, 2007. – № 2(74). – С. 29–32.

Предлагается метод анализа тестопригодности цифрового устройства, представленного на системном уровне (VHDL описание) для выполнения процедуры верификации и синтеза тестов. Метод основан на топологическом анализе схемы, представленной в виде соединенных между собой RTL блоков и дальнейшей модификации схемы с целью повышения общей тестопригодности устройств и повышения качества теста. Ил. 4, библ. 12 (на английском языке; рефераты на английском, русском и литовском яз.).

V. I. Hahanov, M. A. Kaminska, O. Lavrova. VHDL struktūrų testuojamumo analizė pagerinti klaidų aptinkamumą // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2007. – № 2(74). – P. 29–32.

Pristatomas skaitmeninių įtaisų testuojamumo analizės metodas, paremtas sisteminio lygmens aprašu (VHDL aprašu), skirtas atlikti patikrai ir testų sintezės užduotims supaprastinti, klaidų aptinkamumui pagerinti, naudojant testavimo šablonus. Metodas pagrįstas topologine grandyno, atvaizduojamo RTL blokais, analize ir tolimesne jo modifikacija, atskiriant testavimo ir funkcines procedūras. Taip pagerinamas testuojamumas ir supaprastinama testavimo procedūra. Il. 4, bibl. 12 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).