Test Application Methodology Based on the Identification of Testable Blocks

Tomáš Herrman*
Department of Computer Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
herrman@fit.vutbr.cz

Abstract

The PhD thesis deals with the analysis of digital systems described on RT level. The methodology of data paths analysis is described, the data path controller analysis is not solved in the thesis. The methodology is built on the concept of Testable Block (TB) which allows to divide digital component to such segments which can be tested through their inputs/outputs, border registers and primary inputs/outputs are used for this purpose. As a result, lower number of registers is needed to be included into scan chain - border registers are the only ones which are scanned. The segmentation allows also to reduce the volume of test vectors, tests are generated for segments, not for the complete component. To identify TBs, two evolutionary algorithms are used, they operate on TB formal model which is also defined in the thesis.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance; D.2.5 [**Software Engineering**]: Testing and Debugging—*Diagnostics*

Keywords

digital circuit, testable block, scan chain, optimalization, genetic algorithm, simulated annealing, circuit partitioning

1. Introduction

Many methods to improve testability parameters of digital circuits are known. Most of them are based on the controllability and observability concepts. Controllability is seen as the ability to set values of inputs of any component from the primary inputs of the circuit, while observability is the ability to observe values of outputs of any component at the primary outputs of the circuit. If these two parameters are not high enough then it is necessary to increase them which is the goal of many testability analysis methods. It is important to note that when these methods are implemented (i.e. some additional hardware is included into the Circuit Under Analysis - CUA) then the area of chip can grow or dynamic parameters can become degraded.

In this paper, the concept of Testable Block (TB) and methodology based on this concept are defined. It is also indicated how TB can be used to increase testability parameters in terms of controllability and observability of internal nodes of CUA.

TB can be seen as a segment of a digital circuit which is fully testable through its inputs and outputs - boundary registers or primary inputs/outputs of CUA. Such an approach can be used to reduce the number of registers included in scan chain. Border registers are the only registers which can be used as scan registers. In our methodology, TBs will be identified through evolution algorithm which will operate on a formal model of CUA. The purpose is to subdivide the circuit into number of TBs.

2. Motivation for the Research

During the previous research, a formal model of an RTL structure was developed at our department. The formal model describes structural and diagnostic properties of CUA [6, 4]. The goal of the research presented in this paper could be defined as follows:

1. To develop additional formal definitions and algorithms (needed for TBs) to be included into the existing formal model.
2. To develop a methodology which will enable to identify TBs in CUA.
3. To implement the methodology for the identification of TBs in an RTL structure, with the algorithms operating on the formal model.

3. Existing Formal Model

This model was defined in [4, 6], for the purposes of identification of TBs it had to be further extended. The basic quintuple describes the overall structure of circuit. It is

*Recommended by thesis supervisor: Assoc. Prof. Zdeněk Kotásek. Defended at The Faculty of Information Technology, Brno University of Technology on February 24, 2011.

Herrman, T. Test Application Methodology Based on the Identification of Testable Blocs. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 4, No. 1 (2012) 14-19

based on basic circuit components on RT level such as element, port and connection.

The identification of TBs is based on definitions and rules, they will be described now together with the explanation of the reasons for them.

The model was defined in [1], for the purposes of identification of TBs it had to be further extended. The basic quintuple describes the overall structure of circuit. It is based on basic components of circuit on RT level such as element, port and connection. The identification of TBs is based on definitions and rules, they will follow now together with the explanation of the reasons for them.

Definition 1: Let

- E be the set of circuit elements,
- P be the set of elements ports,
- C be the set of connections,
- PI be the set of primary inputs,
- PO be the set of primary outputs,

then $UUA = (E, P, C, PI, PO)$ is an ordered quintuple reflecting CUA structure model on RTL level.

This definition is based on a traditional view on digital circuit and describes circuit structure using quintuple of sets. The methodology described in this paper is supposed to be used on RTL structure only, therefore elements which can appear in such a structure, must be clearly defined.

Definition 2: Let

- R be the set of registers,
- FU be the set of functional units,
- MX be the set of multiplexers,

then $E = (R \cup MX \cup FU)$ is the set of circuit elements in RTL structure.

Definition 2 covers all elements which can appear in an RTL structure, i. e. registers, functional units and multiplexers. This separation is important for testability analysis purposes. The set of registers represents all elements with memory character which cause sequential behavior of CUA. The elements from the MX set control the data flow and the elements from the FU set have a behavior of combinational elements. For the purposes of structural analysis it is also important to define the set of all ports as the points through which diagnostic information will be transported. Each element has ports through which it is connected to ports of other elements or primary inputs/outputs.

Definition 3: Let

- IN be the set of input ports
- OUT be the set of output ports
- CI be the set of control and synchronizing ports

then $P = (IN \cup OUT \cup CI)$ is the set of all ports in an RTL structure.

For the purposes of developing the model, such mechanisms must be created which assign ports from P set to elements which exist in CUA (to registers, FUs and multiplexers).

Definition 4: Let the function: $\psi : E \to 2^P$ be defined, which assigns ports from the set of ports to each circuit element, then:

1. $\psi(e) = \{p | p \in P \land p \text{ is the port of element } e\}$
2. The function ψ is defined for all elements of set E
3. $e_1 \neq e_2 \Leftrightarrow \psi(e_1) \cap \psi(e_2) = \emptyset$

Thus, the function ψ creates the link between the set of elements and the set of ports and assigns the set of ports to elements. The condition 2) provides that the set of ports can be identified for each element. Condition 3 tells that each port belongs to one element only.

All the elements in CUA are interconnected through the connections. The concept of connection is covered by the following definition.

Definition 5: Let C be the set of connections, then $C \subset (PO \cup PI \cup P) \times (PO \cup PI \cup P)$.

The set of connections is a binary relation on the set of ports, the set of primary inputs and the set of primary outputs. The set of connections is the union of topples of ports between which a connection exists. It can be stated that C is reflective, symmetric and transitive.

4. I-path extension

I-path presented in [1] is used to transfer data unchanged from one point in a circuit to another. For the purpose of presented methodology the formal model was extended. We can use i-paths with inverter, but we must take it into account during test patter generation. This path is called ii-path[1]. By connecting two ii-paths we get non inverting i-path.

5. Testable Block

The model of TB will be developed in this chapter. A Testable Block is a segment in CUA separated by registers from other segments. It is guaranteed that the circuitry of TB is testable through its inputs and outputs, transparency properties of CUA are utilized to apply the test. As the consequence, the number of registers included into scan chain is decreased. A TB must satisfy the following features (the features are supported by formal definitions and rules).

It is easy to imagine that if E_{TB} is the set of elements of TB, then it is also a subset of E set, P_{TB} is the set of ports of elements from the set E_{TB} and subset of P, C_{TB} is the set of connections in TB and subset of C. Thus, the structure of a TB can be defined according to the next definitions.

[1] inverted i-path

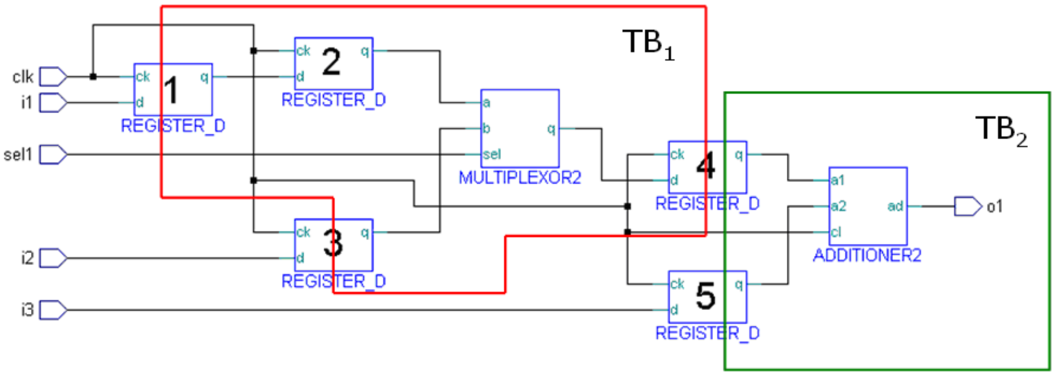

Figure 1: TB example

Definition 6: Let E be the set of circuit elements, then E_{TB} is the set of elements belonging to TB and it must hold $E_{TB} \subseteq E$.

Definition 7: Let E_{TB} be the set of elements of TB, P be the set of circuit elements ports and $\psi(e)$ be the function which assigns a port from the set of ports to element e, then $P_{TB} = \bigcup\limits_{e \in E_{TB}} \psi(e) \subseteq P$ is the set of TB ports.

Definition 8: Let PO be the set of primary outputs of CUA, P be the set of circuit elements ports and C be the set of connections, then $PO_{TB} = \{po | po \in PO \wedge (\exists p \in P_{TB} : ((p, po) \in C))\} \subseteq PO$ is the set of primary outputs which are directly connected to TB.

Let PI be the set of primary outputs of CUA, P be the set of circuit elements ports and C be the set of connections, then $PI_{TB} = \{pi | pi \in PI \wedge (\exists p \in P_{TB} : ((p, pi) \in C))\} \subseteq PI$ is the set of primary inputs which are directly connected to TB.

Definition 9: Let E_{TB} be the set of elements of TB and R be the set of circuit registers, then $R_{TB} = \{r | r \in R \wedge r \in E_{TB}\}$ is the set of TB registers.

Every connection from a port of an element inside TB which does not start or does not end in the register inside TB must end or start in the element inside TB or on a primary input or a primary output. This is to say that FUs and MXs inside TB must not have their ports connected to ports outside TB, the connection is possible through registers only. This condition is reflected by the following two definitions.

Definition 10: Let R_{TB} be the set of TB registers, C be the set of connections, P be the set of elements ports, $\psi(e)$ be the function which assigns a port from the set of ports to an element e and P_{TB} be the set of ports of elements from the set E_{TB}, then $BRO_{TB} = \{r | r \in R_{TB} \wedge \exists(p_1, p_2) \in C : (p_1 \in \psi(r) \wedge p_2 \in (P \backslash P_{TB}))\}$ is the set of output border registers of TB.

Definition 11: Let R_{TB} be the set of TB registers, C be the set of connections, P be the set of elements ports, $\psi(e)$ be the function which assigns a port from the set of ports to an element e and P_{TB} be the set of ports of elements from the set E_{TB}, then $BRI_{TB} = \{r | r \in R_{TB} \wedge \exists(p_1, p_2) \in C : (p_1 \in (P \backslash P_{TB})) \wedge p_2 \in \psi(r)\}$ is the set of input border registers of TB.

Definition 12: Let C be the set of connections, P_{TB} be the set of ports of elements from the set E_{TB}, PO_{TB} be the set of primary outputs which are directly connected to TB, P be the set of elements ports, $\psi(e)$ be the function which assigns a port from the set of ports to an element e, BRO_{TB} be the set of output border registers of TB and BRI_{TB} be the set of input border registers of TB, then $C_{TB} = C \cap [(P_{TB} \times (PO_{TB} \cup P_{TB})) \cup ((PO_{TB} \cup P_{TB}) \times P_{TB}) \cup ((\bigcup\limits_{e \in BRO_{TB}} \psi(e)) \times P) \cup (P \times (\bigcup\limits_{e \in BRI_{TB}} \psi(e)))]$ is the set of TB connections.

Finally, TB formal model definition:
Definition 13: Let E_{TB} be the set of TB elements, P_{TB} be the set of TB ports, PO_{TB} be the set of primary outputs which are directly connected to TB, C_{TB} be the set of TB connections, PI_{TB} be the set of primary inputs which are directly connected to TB and it must hold $\forall e \in E_{TB}, \forall p \in \psi(e), \exists(p_1, p_2) \in I : (((p_1 \in PI_{TB}) \vee (p_1 \in BR_{TB})) \wedge p_2 = p) \vee (p_1 = p \wedge (p_2 \in PO_{TB}) \vee (p_2 \in BR_{TB}))$, then $TB = (E_{TB}, P_{TB}, C_{TB}, PI_{TB}, PO_{TB})$ is formal model of TB.

A test to a digital circuit is always applied through registers. The identification of the registers is the goal of testability analysis methodologies, our methodology is based on the identification of border registers.

Rule 1: All registers which are inside a TB must not be included into scan chain.
$\forall r \in (R_{TB} \backslash BR_{TB}) : r \notin SCAN$

Rule 2: Any TB must not overlap another TB (any two TBs must be disjunctive).
For each TB_x and TB_y must hold $\forall e \in (E_{TB_x} \backslash BR_{TB_x}) : e \notin E_{TB_y}$.

Figure 1 shows an example of circuit divided into 2 TB. TB$_1$ has one internal register R2, three border registers and one internal element (multiplexor2).

6. TB Identification Methodology

6.1 Formal model construction

The goal of this step is to convert the design described in HDL (Hardware Description Language) into formal representation that was described in the previous sections. The procedure is as follows. First, the design is synthesized into the RTL by the third party commercial tool. The result is a structural VHDL file based on the RTL primitives. The VHDL file is converted to our formal model by means of tools (vhdl2zb, zb2ruz) developed for this purpose. The vhdl2zb is VHDL parser, it converts the design into intermediate binary format that describes the structure. Then, it is analyzed by zb2ruz and i-paths in the circuit are identified. We recognize two i-paths categories - the classical transparent and the inverting ones (can be modeled as an i-path with inverter). For the i-paths identification, the library of i modes is utilized. The output of this step is textual file that contains formal representation of the design.

6.2 Testable blocks partitioning

The purpose of this step is to divide the circuit into independently testable partitions of logic – TBs. For this purpose, the developed partitioning tool is used. It operates over model created in previous step. Each CUA register can be classified as input, output or internal register of the TB. This means that we have 3^n options how to assign function (input, output, internal register) to registers available in CUA, where n is the number of registers in the CUA. Thus the size of the search space grows exponentially. Because of the exponential complexity of the problem and because it is possible to utilize a simple representation of the problem in bit-string (chromosome), genetic algorithm [5] and simulated annealing [2, 3] were chosen. Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. Genetic algorithms are known to be quite good at finding generally good solutions in acceptable time even for big search spaces.

Simulated annealing is a generic probabilistic metaheuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global optimum of a given function in a large search space. The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends both on the difference between the corresponding function values and also on a global parameter T (called the temperature), that is gradually decreased during the process.

The fact whether a particular register is selected as a border register is encoded into the chromosome. All the identified TBs candidates are then recursively checked to satisfy all the previously given definitions and rules. For the TBs that satisfy the definitions, a fitness value is calculated. For the experiments described in this paper, the fitness value depends on the number of logic outside TBs and sequentional depth of the identified TBs (blocks with the lower value of depth are preferred). The mutual independence of TBs (in terms of testing) is suitable for various optimizations over TB test vectors set without influencing other TBs.

Basic circuit data:

1	elements	146	8	control inputs	75
2	connections	6131	9	all ports	496
3	registers	75	10	f. coverage (no scan)	66%
4	primary inputs	36	11	i-paths	7688
5	primary outputs	10	12	ρ size	8452
6	input ports	225	13	conver. time to mod.	6m 27s
7	output ports	150	14	Mentor suggestion	44; 9

Genetic algorithm:

15	average TB division time	13,2s
16	one generation computation time	36ms
17	registers in scan	30
18	border registers	68
19	TBs	9

Simulated annealing:

20	average TB division time	15,2s
21	one generation computation time	35ms
22	registers in scan	8
23	border registers	33
24	TBs	3

Direct division method:

25	average TB division time	2,4s
26	registers in scan	30
27	border registers	70
28	TBs	10

Table 1: Experimental results with circuit *m10*

6.3 Experimental results

The following experiments were made on the set of the test circuit which were developed for this purpose and on other test circuit.

Table 1 shows experimental results for one of the test circuit. The first row informing about number of the circuit elements (size of E) that includes registers, combinational elements and multiplexers. On the second row there is the number of connections between two ports(correspond to size of C). The numbers on rows 3 to 8 indicate the size of R, PI, PO, IN, OUT, CI sets. Total number of port (sum of rows 4 to 8) is on the row 9. The fault coverage on row 10 was determined by *Flextest* generator without scan chain usage. The row 11 shows number of i-paths in the circuit. On the next row 12 there is the size of ρ set in which the path between two ports between which i-path exist is stored. All the times in next rows are average times form 10 measuring. Rows 18,23 and 27 show the number of border registers. The registers which are accessible from PI/PO are eliminated and the final number of the registers in scan chain are on rows 17,22 and 26. The number of the registers for scan chain suggested by DFTAdvizor - Mentor Graphics® (row 14) is subdivided into two parts. The first part uses *ATPG-Based Partial Scan* method and the second uses *AUTOMATIC-Based Partial Scan* method.

The *direct division method* does not use evolutionary technique but select all possible registers as the border registers and divide circuit into maximal number of TB.

The following tables demonstrate:

1. Time complexity of used algorithms and its relation to the number of connections (cons) and i-paths - see Table 2,

Name	FUs	FFs	i-paths	Cons	i-paths search time	TB partit. time	Num of TB	Units outside TBs
	[1]	[1]	[1]	[1]	[s]	[s]	[1]	[%]
COM	45	21	1874	1217	242	3.2	5	17
ISA	75	29	8831	2988	2375	15.2	2	4
DIFFEQ	11	6	2041	213	2.2	0.81	1	0
DEC	29	7	1283	529	22.8	4.3	2	20.6
FIFO2	226	144	33339	11297	41200	233	6	14
S298	47	19	10204	1367	349	14.7	2	3

Table 2: Results of TB partitioning algorithm

Name	BRs	FFs	FUs	nodes
[-]	[1]	[1]	[1]	[1]
TB1	2	4	12	15
TB2	4	1	6	3
TB3	2	4	12	15
TB4	2	4	12	15
TB5	2	4	12	15

Table 3: Details about TBs identified in COM component

Name	BRs	FFs	FUs	nodes
[-]	[1]	[1]	[1]	[1]
TB1	28	86	222	241
TB2	3	3	8	7
TB3	3	3	8	7
TB4	3	3	8	7
TB5	5	1	6	3
TB6	5	1	6	3

Table 4: Details about TBs identified FIFO2 component

2. effectiveness of algorithms for partitioning UUA into TBs - see Table 2,

3. properties of TBs - see Tables 3 and 4.

Legend for table 2:

Name	Circuit identification
FUs	The number of functional units in the circuit (before mapping to AMI library)
FFs	The number of flip/flops in circuit
i-paths	The number of i-paths identified in circuit
Cons	The number of connections in circuit
i-paths search time	Time needed for identification of all i-paths in circuit
TB partit. time	Time needed for partitioning circuit into TBs
Num of TBs	The number of TBs identified in circuit
Units outside TBs	The number elements (registers and functional units) which were not included into TBs

Legend for tables 3 and 4:

Name	Circuit identification
BRs	The number of border registers in TB
FFs	The number of flip/flops in TB
FUs	The number of functional units (from the AMI library) the TB is built from
Nodes	The number of nodes in TB (without interface)

In tables 3 and 4, blocks that have equal values in columns 2-7 seem to be identical (can be freely swapped in the circuit without any function modification)

It can be stated that the time complexity is strongly dependent on the number of connections in circuit which was an expected result. Our motivation was to gain absolute values of computational times. We also verified which circuit portion was kept outside TBs and thus not tested (9th column in Table). For example for circuit COM, it resulted in 17% circuitry kept outside TBs.

From tables 2 and 3, the partition of COM and FIFO2 circuit into TBs is evident. FIFO2 circuit is partitioned into TBs unequally, so one block is much bigger while COM2 is partitioned more equally. 3 and 4 represent detailed information on partitioning of TBs.

We also verified that that the formal model can be used for the development of testability analysis algorithms. We also revealed that data needed for the construction of the model can be gained from data available in the professional design systems, e. g. Mentor Graphics.

7. Conclusions

The software we have developed is able to:

1. analyze Verilog/VHDL description and transform it into formal model,
2. partition circuit into TBs,
3. identify border registers and include them into partial scan chain,
4. identify and break loops.

It can be concluded that a formal model (representation) of RTL structure was developed on which testability algorithms can be implemented. It was proven that a diagnostic problem can be thus converted into a mathematical problem (theory of graphs, discrete mathematics and theory of sets). We have shown how the formal model can be utilized for the identification of TBs together with the principles of integrating the methodology into professional design tools. The formal model can be further extended in the future if a need to develop and verify testability analysis methodology appears.

Acknowledgements. This research was supported by the Grant Agency of the Czech Republic under grant No. GA102/09/1668 SoC circuits reliability and availability improvement.

References

[1] M. S. Abadir and M. A. Breuer. A knowledge based system for designing testable vlsi chips. *IEEE Design & Test of Computers*, pages 56–68, 1985.

[2] V. Cerny. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, pages 41–51, 1985.

[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science. New Series*, 220:671–680, 2009.

[4] Z. Kotásek. Uplatnění principů řiditelnosti/pozorovatelnosti při návrhu číslicových obvodů, habilitační práce. page 80. Brno University of Technology, 2000.

[5] M. Obitko. Introduction to genetic algorithms with java applets. http://www.obitko.com/tutorials/genetic-algorithms, 1998.

[6] R. Růžička. *Formální přístup k analýze testovatelnosti číslicových obvodů na úrovni RT*. PhD thesis, Fakulta informačních technologií VUT v Brně, 2002.

Selected Papers by the Author

J. Škarvada, Z. Kotásek, T. Herrman. Power Conscious RTL Test Scheduling,. *Proceedings of 11th Euromicro Conference on Digital Systems Design Architectures, Methods and Tools*, Los Alamitos, US, IEEE CS, 2008, p. 721–728

J. Škarvada, Z. Kotásek, T. Herrman. Testability Analysis Based on the Identification of Testable Blocks with Predefined Properties. *Microprocessors and Microsystems*, Vol. 32, No. 5, 2008, Amsterdam, NL, p. 296–302

J. Škarvada, T. Herrman, Z. Kotásek. RTL Testability Analysis Based on Circuit Partitioning and Its Link with Professional Tool. In *IEEE 8th Workshop on RTL and High Level Testing*, Beijing, CN, ICTAC, 2007, p. 175–181

J. Škarvada, T. Herrman, Z. Kotásek. Testability Analysis Based on the Identification of Testable Blocks with Predefined Properties. In *10th EUROMICRO CONFERENCE ON DIGITAL SYSTEM DESIGN Architectures, Methods and Tools (DSD 2007)*, Lübeck, DE, IEEE CS, 2007, p. 611–618.

T. Herrman. Formal Model of Testable Block. *Proceedings of 12th Conference Student EEICT 2006*, Volume 4, Brno, CZ, FEKT VUT, 2006, p. 451–455.