The Implementation of RTL Testability Analysis Algorithms through the Discrete Mathematics Concepts

Zdeněk Kotásek, Richard Růžička
Department of Computer Science and Engineering
Brno University of Technology, Božetěchova 2, 612 66 Brno, Czech Republic
e-mail: kotasek,ruzicka@dcse.fee.vutbr.cz

Abstract

In the paper a formal approach to the RT level testability analysis will be presented. It will be shown how the discrete mathematics concepts can be used for the testability analysis. The elements of the circuit are classified into sets according to their function in the design and role during the test application. Then, other sets, relation and mappings are defined to reflect the role of registers and the structure of i path in the circuit. These formal tools are then used to prove the testability of the circuit under analysis.

1. Introduction

Several possibilities how to apply a test of a digital circuit exist. One is through a Partial scan which is a Design for Testability (DFT) technique that scans only a subset of the flip-flops (FFs) of a sequential circuit, attempting to provide a trade-off between the testability of the circuit and the overheads (area and delays) introduced by scan design. Such an approach has some of the advantages of the full scan in the sense that it can considerably increase the controllability and observability of elements in the sequential circuit. On the other hand, it is not as costly as full scan since only a limited set of registers need to be scannable. The first partial scan methods were based upon testability measures [2]. Flip-flops for the scan path were selected using testability measures like controllability and observability. A sequential test pattern generator was still required to handle the remaining circuit. The effectiveness of these approaches highly depends on the quality of the testability measures with respect to the used test pattern generator.

In [5], the PARES system is presented. It is based on the reduced scan shift. For every test vector it is determined, which FFs must be controlled or observed and this is done for a large number of test vectors. FFs that are required to be controlled and observed for many test vectors are selected, and then FFs are arranged in a scan chain according to the difference between the requirements for control and observation. Alternatively, in the second approach, the selection of FFs is not based on test pattern generator results but on structural information, namely the presence of loops. The criterion for partial scan FFs is to break all feedback cycles. It must be noticed that this kind of scan insertion methods does not guarantee a given fault coverage and the SATPG is still required [3], [6].

The last approach is based on selecting at RT level a set of registers from which FFs will be selected for scan path insertion. In [4] a methodology is presented describing datapath and transparency characteristics of modules. Registers that are totally testable are identified and are eliminated from the candidates for scan. Then, registers that are candidates for scan are determined as the result of the structural analysis. In [7], a three-stage process for partial scan is designed. The first two stages focus on cycle-breaking and

This research has been carried out with the financial support of the Grant Agency of the Czech Republic under contract No. 102/98/1463 and the Research Intent of FEI, Brno University of Technology, Czech Republic, grant No. CEZ: J22/98: 262200012 "Research of the Information and Control Systems".

on limiting the maximum length of consecutive self-loops. For the third stage, combinational blocks and their effects on sequential test generation are evaluated using a graph-theoretic representation of the circuit flow graph. The methodology is a combination of the two methodologies discussed above.

The identification of registers through which the test of the unit under analysis (UUA) will be applied is the basic problem which must be solved — no matter how the registers will be modified (full scan, partial scan, BIST). Most of the approaches are based on heuristic methods. Before these methods are applied, the structure of the UUA must be analysed first. During the analysis i paths are searched in the UUA along which diagnostic data are transferred. The concept of i path was described in [1].

2. Definition of the Problem

So far, we have dealt in our research activities with the RTL testability analysis based on applying heuristic approaches. We analysed the structure of UUA and suggested techniques on the basis of which test registers were selected [7],[8]. The possible role of registers during the test application was also taken into account [9].

For the purposes of the testability analysis, *i paths* and their structure must be identified. The *i paths* are such structures through which diagnostic data (test vectors and responses to them) can be transferred without being modified. Two types of *i paths* are distinguished: *i paths* through an element structure (i. e. from its input to its output) and *i paths* from the output of an element to the input of another element (this type of an *i path* can possibly pass through a structure consisting of more than one element). If diagnostic data can be transferred from an element input to an element output, then we say that the element can operate in the *i mode*. On the basis of *i mode* criterion three types of elements can be identified in a UUA: DT (Data Transporter) type elements which are always transparent (e. g. registers), DP (Data Processors) type elements which are never transparent and DP/DT in which an *i mode* can be set only if certain conditions are fulfilled.

During our research activities we realised that it is reasonable to use mathematical tools and concepts for the UUA structure analysis. The first step of such an approach was the classification of elements in the UUA on the basis of their role during the test application. An important role is assigned to registers, therefore their classification is the finest. Furthermore, set theory is closely related to logic: so much so that many concepts in the two disciplines are almost interchangeable. We also realised that the theory of graphs represents a powerful tool for the UUA structure representation and its analysis. Our approach is based on the matrix representation of the structure and enumerating the transitive closure on such representation. To construct the transitive closure, the Warshall algorithm was used.

It can be stated that theory of sets, logic and theory of graphs combined together complement each others‘s capabilities significantly, thus creating a powerful formalism for concise and precise representation of information and reasoning about it. In the paper it will be presented how we solved the problem of testability analysis by means of discrete mathematic concepts and tools. The paper is organised as follows: Section 3 containst basic definitions of formal tools used in our approach, Section 4 contains an example of DFT algorithm based on definitions from the section 3. There are some experimental results in Section 5.

3. Basic Concepts

3.1 Concepts Dealing with Analysis of the Circuit Structure

For our purposes, i.e. the formal approach to the RTL testability analysis, we describe a digital circuit as a tuple:
$UUA = (E, P, C, PI, PO)$, where:

E is the set of circuit elements and

it can be divided into several disjunctive classes $E = (R \cup MX \cup FU)$, where:

- R is a set of registers,
- MX is a set of multiplexers and
- FU is a set of functional units.

PI is a set of primary input ports of the UUA and

PO is a set of primary output ports of the UUA.

P is the set of ports divided into four disjunctive classes $P = (IN \cup OUT \cup IO \cup CI)$, where:

- IN is a set of input data ports,
- OUT is a set of output data ports,
- IO is a set of data ports, which are both input and output and
- CI is a set of control inputs, clock inputs and enable inputs,

and mappings exist:

$\psi : E \mapsto 2^P$,
which assigns a set of ports to the element:

$\psi(e) = \{p \mid p$ is a port of the element $e\}$

and
$\gamma : (IN \cup OUT \cup IO) \mapsto \prod 2^{CI}$, which
assigns a sequence of sets of control inputs to the port:

$\gamma(p) = (CI_1, \cdots, CI_n)$, where the set $CI_k = \{ci \mid ci$ is the control input, which enables the input $p\}$ and $n \in \{0, \cdots, k, \cdots, n\}$.

The symbol $\prod 2^{CI}$ means n-carthesian product (n-tuples) of power set of the CI set, where n can be any natural number including zero.
C is a set of connections: $C \subset (PI \cup OUT \cup IO) \times (PO \cup IN \cup IO)$, a relation, which contains ordered pairs of ports between which an *i path of length 0* exists.

The principles of the testability analysis require to distinguish between transparent and non-transparent structures. Therefore, the structures which in at least one of their modes can establish an *i path* from their inputs to the outputs will be included into the set $FUI \subset FU$. We said, that these structures have an *i mode*.

The existence of an *i mode* of the element is expressed by a relation:
$M \subset (IN' \cup IO') \times (OUT' \cup IO')$, where

$IN' = \{in' \mid in' \in IN \wedge \exists Q : (in' \in Q \wedge \psi^{-1}(Q) \in (FUI \cup R \cup MX))\}$,

$OUT' = \{out' \mid out' \in OUT \wedge \exists S : (out' \in S \wedge \psi^{-1}(S) \in (FUI \cup R \cup MX))\}$ and

$IO' = \{io' \mid io' \in IO \wedge \exists T : (io' \in T \wedge \psi^{-1}(T) \in FUI)\}$.

So the elements of the relation M are ordered pairs of ports between which an *i path* exists.

The *i mode* of the circuit element, when there is an *i path* from an element input to an element output can be conditioned by some control signals and/or values on data inputs (*data dependent i path*). In this case, the mapping:

$$\mu : M \mapsto \prod 2^P$$

describes a sequence of control signals (and ports), which are necessary to be generated to establish the *i mode*.

$\mu((i, o)) = (CP_1, \cdots, CP_n)$, where the set $CP_k = \{cp \mid cp$ is the data or control input, which enables the *ipath* inside an element from the input i to output $o\}$ and $n \in \{0, \cdots, k, \cdots, n\}$.

Let a tuple $(i,o) \in M$ and $\exists Q : i \in Q \wedge \psi^{-1}(Q) = fui \in FUI$ and $\exists S : o \in S \wedge \psi^{-1}(S) = fui \in FUI$ and one or more elements from tuple $\mu(i,o)$ belong to the set $(IN \cup OUT \cup IO)$. Then we say that functional unit fui has a *data dependent i mode* and internal *i path* (through the unit) from input i to output o is *data dependent*. Such *i paths* create a subset of the M relation denoted as $M_d \subset M$.

For the purposes of the testability analysis it is necessary to identify all the *i paths* of different lengths in the UUA. This is a relation:

$I = (C \cup M)^*$, where * means the transitive closure of the relation.

The relation I can be understood as a relation, which represents edges in a graph. The set of vertices is a subset of the P set, of course. Now, the predicate reflecting the existence of an *i path* between two ports $p1, p2$ can be introduced as:

$(p1, p2) \in I \Leftrightarrow i(p1, p2)$.

3.2 Concepts Dealing with Testability Analysis

Let us have two sets:

$IP \subset P$, and $IP = \{ip \mid ip \in (IN \cup IO) \wedge \exists X : (ip \in X \wedge \psi^{-1}(X) \in FU)\}$ is the set of all input ports of all fus and

$OP \subset P$, and $OP = \{op \mid op \in (OUT \cup IO) \wedge \exists Y : (op \in Y \wedge \psi^{-1}(Y) \in FU)\}$ is the set of all output ports of all fus.

During the analysis it is necessary to identify registers in the UUA, judge their possible role during the test application and include them into sets accordingly. Thus, the following sets are distinguished:

$TDR \subset R$ is a set of test drivers, this set is distinguished for all input ports $ip \in IP$, so $TDR_{ip} = \{tdr \mid tdr \in R \wedge \exists q : q \in \psi(tdr) \wedge (q, ip) \in C\}$. So the set TDR_{ip} is the set of all registers, which are connected immediately (by an *i path of length 0* to the input port ip and

$TRV \subset R$ is a set of test receivers, this set is distinguished for all output ports $op \in OP$, so $TRV_{op} = \{trv \mid trv \in R \wedge \exists d : d \in \psi(trv) \wedge (op, d) \in C\}$. So the set TRV_{op} is the set of all registers, which are connected immediately (by an *i path of length 0* to the output port op.

The sets TDR are sets of registers, from which all input ports of all functional units in the UUA are controllable and the sets TRV are sets of registers, on which all output ports of all functional units in the UUA are observable. Such classification of registers can be sufficient for some DFT techniques (e.g. BIST), but for another DFT techniques (e.g. Partial Scan), it can be useful to create some other sets:

$TIR \subset R$ is a set of test input registers. This set is distinguished for each $tdr \in \bigcup TDR_{ip}$, so $TIR_{tdr} = \{tir \mid tir \in R \wedge (\exists d_1, q_1 : d_1 \in \psi(tdr) \wedge q_1 \in \psi(tir) \wedge i(q_1, d_1)) \wedge tir \in CTRL)\}$. In other words, the set TIR_{tdr} is the set of registers, that are controllable and it is possible to reach the input of an appropriate tdr from their outputs using *i paths* existing in the UUA and

$TOR \subset R$ is a set of test output registers. This set is distinguished for each $trv \in \bigcup TRV_{op}$, so $TOR_{trv} = \{tor \mid tor \in R \wedge (\exists d_2, q_2 : d_2 \in \psi(tor) \wedge q_2 \in \psi(trv) \wedge$

$i(q_2, d_2)) \wedge tor \in OBSR)\}$. In other words, the set TOR_{trv} is the set of registers, that are observable and it is possible to reach their inputs from the output of an appropriate trv using *i paths* existing in the UUA.

It is evident, that registers from the the set $\bigcup_{ip} TDR_{ip}$ can be controlled by registers that form some subset of $\bigcup_{tdr} TIR_{tdr}$. So the problem of coverage of the set $\bigcup_{ip} TDR_{ip}$ by minimal subset of $\bigcup_{tdr} TIR_{tdr}$ appears and some algorithm to "prune" this set is needed. Such algorithm will be described later. The problem with the $\bigcup_{trv} TOR_{trv}$ set is similar.

As described above, registers in TIR (TOR) sets must be controllable (observable) from (on) primary inputs (outputs) of the UUA. This situation is described in formal definition by expression $tir \in CTRL$ ($tor \in OBSR$). Sets $CTRL$ and $OBSR$ are defined here:

$CTRL = \{cr \mid cr \in R \wedge ((\exists d_3, pi : d_3 \in \psi(cr) \wedge pi \in PI \wedge (pi, d_3) \in C) \vee cr \in SCAN)\}$ and

$OBSR = \{or \mid or \in R \wedge ((\exists q_3, po : q_3 \in \psi(or) \wedge po \in PO \wedge (q_3, po) \in C) \vee or \in SCAN)\}$.

In other words, the register is controllable (observable), if it is included into scan chain — the register is included into $SCAN \subset R$ set — or their input (output) is connected directly (*i path of length 0* exist) to the primary input (output) of the UUA.

3.3 Testability features of the UUA

Let us have sets:

$CIP \subseteq IP$ and $CIP = \{cip \mid cip \in IP \wedge \exists tir, tdr : (tir \in TIR_{tdr} \wedge tdr \in TDR_{cip})\}$ is a set of input ports that are controllable from a tir and

$OOP \subseteq OP$ and $OOP = \{oop \mid oop \in OP \wedge \exists tor, trv : (tor \in TOR_{trv} \wedge trv \in TRV_{oop})\}$ is a set of output ports that are observable on a tor.

Then it is possible to calculate a percentage of controllability of element input ports:

$poc = \frac{|CIP|}{|IP|} \cdot 100$ and

a percentage of observability of element output ports:

$poo = \frac{|OOP|}{|OP|} \cdot 100.$

A percentage of testability of the UUA now can be defined in this way:

$pot = \frac{poc+poo}{2}$

4. An Example of the DFT Algorithm

We have utilised the proposed formal tools to develop an algorithm that selects a set of registers, which are needed to be included into the Partial Scan chain. This set is called $SCAN$.

The proposed algorithm is as follows:

1. The identification of all sets in the UUA tuple. All sets are derived from VHDL description of the circuit. The UUA is described by its structure, elements which create the circuit are described behaviourally. The ψ mapping is also derived.

2. The creation of sets TDR_{ip} and TRV_{op} for all input and output ports of all $fu \in FU$.

3. The creation of sets TIR_{tdr} for all $tdr \in \bigcup_{ip} TDR_{ip}$ and TOR_{trv} for all $trv \in \bigcup_{op} TRV_{op}$.

4. The identification of the set $SCAN \subseteq (\bigcup_{tdr} TIR_{tdr} \cup \bigcup_{trv} TOR_{trv})$. This is done by the following steps (described for $tirs$, the algorithms are similar for $tors$):

   (a) The unavoidable registers are identified. If the set $TIR_{tdr1} = \{tir\}$, where $tdr1 \in TDR_{ip}$ and $\nexists tdr2 :$

$tdr2 \in TDR_{ip}$, then this tir is denoted as unavoidable and must be included into the $SCAN$ set. All input ports controlled by this tir are unimportant for the next analysis (they are covered by the tir and can be excluded from the next analysis).

(b) For the reduction of the set of tir elements, the following rules are applied onto the set:

  i. If $TIR_{tdr1} = TIR_{tdr2}$, then the $tir \in TIR_{tdr1}$ from which the tdr elements are more easily controllable (γ mapping) is retained.

  ii. Let $TIR_{tdr1} \subset TIR_{tdr2}$, then the registers from TIR_{tdr2} are retained.

5. Experimental Results

To improve testability features of several digital circuits, the proposed algorithm was used. Together with circuits, which were developed to evaluate the effectivenes of the methodology, benchmark circuits was used. These benchmark circuits are described in VHDL structurally at RT level. One of them, the *Diffeq* circuit [8] contains 6 registers. Moreover, no register is controllable from/observable on primary ports. The proposed methodology selects three registers to be included into the Scan Path. With these registers it is possible to cover all *fu*s. Another circuit used to verify efficiency of the algorithm is *Tseng* benchmark circuit. Two registers of five were recommended to include them into Scan Path.

6. Conclusion

Using of formal tools to describe the structure of the circuit and its testability features bring several advantages. One of them is the *provability*. Now it is possible to decide exactly, if the circuit is testable and which *fu*s are not fully controllable or observable. Moreover, the use of the transitive closure brings acceleration of the testability analysis procedure. When e.g. the Warshall algorithm is for *i path* analysis used, the complexity of the problem is changed to polynomial.

References

[1] Abadir, M. S. - Breuer, M. A.: A Knowledge Based System for Designing Testable VLSI chips, IEEE Design&Test, August 1985, pp. 56–68

[2] Agrawal, V. D.- Cheng, K. - Johnson, D. D. - Lin, T..: A Complete Solution to the Partial Scan Problem, Proc. of the 1987 International Test Conference, September 1–3, 1987, Washington, pp. 44–51

[3] Dey, S. - Potkonjak, M. - Roy, R.: Synthesizing Designs with Low-Cardinality Minimum Feedback Vertex Set for Partial Scan Application, Proc. of 12th IEEE VLSI Test Symposium, April 25–28, 1994, Cherry Hill, USA, pp. 2–7

[4] Flottes, M. L. - Pires, R. - Rouzeyre, B. - Volpe, L.: A Fast and Effective Technique for Partial Scan Selection at RT Level, Proc. of IEEE ETW 1997, May 28–30, 1997, Cagliary, Italy, pp. 36–42

[5] Higami, Y. - Kajihara, S. - Kinoshita, K.: Partial Scan Design and Test Sequence Generation Based on Reduced Scan Shift Method, Journal of Electronic Testing: Theory and Applications, 7, Kluwer Academic Publishers, 1995, pp. 115–123

[6] Chakradhar, S. T. - Balakrishnan, A. - Agrawal, v. D.: An Exact Algorithm for Selecting Partial Scan Flip-Flops, Journal of Electronic Testing: Theory and Applications, 7, Kluwer Academic Publishers, 1995, pp. 83–93

[7] Tai, S. - Bhattacharya, D.: A Three-Stage Partial Scan Design Method to Ease ATPG, Journal of Electronic Testing: Theory and Applications, 7, Kluwer Academic Publishers, 1995, pp. 94–104

[8] Paulin P. - G., Knight J. - P., Girczyc E. F.: HAL: A multi-paradigm approach to automatic data path synthesis, Proc. of 23rd Design Automation Conference, 1986, pp. 263–270