Behavioural and Adaptable Design for Testability

Tudor NICULIU[*] Chouki AKTOUF[+] Chantal ROBACH[+]
[*]: Universitatea POLITEHNICA din Bucuresti, Facultatea de Electronica si Telecomunicatii,
Bucarest, ROUMANIA
[+]: LCIS-INPG, 50 Rue Barthelemy de Laffemas, B.P. 54, 26902, Cedex 9,
Valence, FRANCE

Abstract

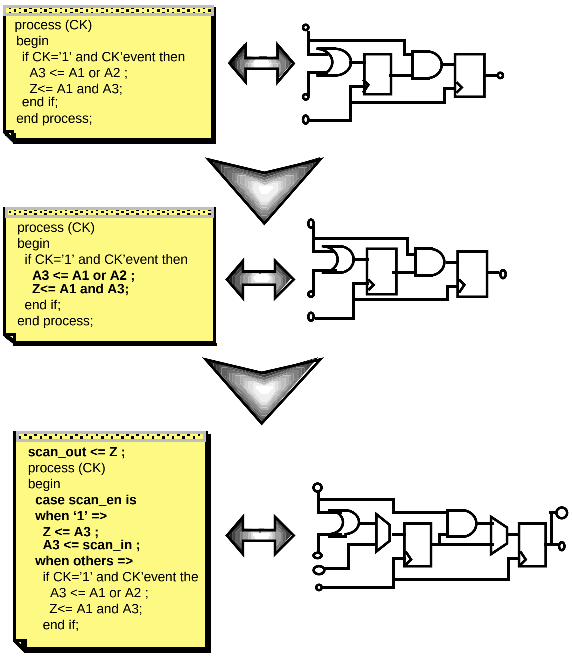
This work proposes a design-for-testability (DFT) approach at the behavioural level. The approach uses testability measures in order to select necessary test points. Given a behavioural description of a complex design, the proposed measures detect signals and variables that cause a low test quality. For that, different aspects such as a syntactical description and an analysis of a FSM-based model are combined. The proposed approach is first used to select memory elements which are considered in an internal scan DFT approach.

Summary

DFT must suit the behavioural specification of today's complex system design. Referring to high-level synthesis, DFT can operate before, while or after it. The first choice necessitates the use of a suitable DFT technique, model or method to the particular design. We call it behavioural adaptable DFT: it improves the testability, measured with adequate methods, direct on the behavioural specification or aided by special representations, that have to permit returning to the behavioural description after improving the testability of the system to be designed. The second choice is synthesis for testability, while the third could be called register-transfer level DFT. In the case of a behavioural adaptable DFT, this work focuses on the use of scan.

Given a behavioural description (e.g. written in VHDL), memory elements are first detected (see Figure 1).

Once memory elements are detected, the selection of those to include in the scan register, i.e. partial scan (PS) is considered. Most of PS methods are applied by now on register-transfer level and lower levels of the design hierarchy.



```
process (CK)
 begin
  if CK='1' and CK'event then
    A3 <= A1 or A2 ;
    Z<= A1 and A3;
  end if;
 end process;
```

```
process (CK)
begin
  if CK='1' and CK'event then
    A3 <= A1 or A2 ;
    Z<= A1 and A3;
  end if;
end process;
```

```
scan_out <= Z ;
process (CK)
begin
  case scan_en is
  when '1' =>
    Z <= A3 ;
    A3 <= scan_in ;
  when others =>
    if CK='1' and CK'event the
      A3 <= A1 or A2 ;
      Z<= A1 and A3;
    end if;
```

Figure 1 : Scan-based DFT approach at the behavioural level

The proposed selection method considers different selection criteria based on combination of the following models:

- *Structural*: A weighted directed graph (S-graph), models the flip-flops as nodes and the combinational paths between them as arcs; testability is represented by weights, related to total length of feedback cycles and to sequential depth (longest in-to-out path) [1].
- *FSM-based*: A Finite-State-Machine (FSM) network, models the behavioural specification; the operation of this model is simplified by implicit techniques for state enumeration, efficiently generating the state transition graph and storing it as characteristic functions in reduced ordered Binary Decision Diagrams, allowing to estimate testability of memory elements by reachability analysis [3].
- *Textual*: A Hardware Description Language (HDL) behavioural specification guides directly the testability measurement and improvement,

determining the contributions to low testability of the hardware corresponding to signals/ variables in the behavioural specification of the system/ component [4]. Syntactical analysis on the behavioural description is used to measure the testability, determining different aspects that reduce it for the specified module; hard-to-test parts, represented by variables/signals, are indicated. Experimented textual testability measures are: restricted value range of variable/signal in a description statement, non-uniform distribution caused by an operation on the values of variables/signals, the reduced accessibility of instructions caused by conditions. These measures are compatibilized by algebraic operations and combined to reflect the testability of a variable/signal.

The overall *testability measure* is a combination of different aspects that contribute to the low testability of the memory elements corresponding to objects in the behavioural description (signals/variables) of the system/ component. These aspects can be modeled by weights in the behavioural S-graph, in addition to the structural testability measures based on feedback cycles in the graph and sequential depth: the relationship degree between nodes modeling memory elements, represented by arc weights, can be reduced by conditions in the HDL description. Arc weights determine node weights through topology, finally reflecting the hardness to test the behavioural object corresponding to the node. Node weights are further influenced by characteristics that can be determined by an analysis of the description text: restricted value range of an object in a description statement and non-uniform distribution caused by an operation on the object's values. For each method, a parameter p_i which reflects the importance given to the method is considered.

To illustrate the proposed approach, a B-VHDL description of a 16-bit counter is considered. The related S-graph is shown in Figure 2. Please see [2] for the VHDL description.

The following objects are detected as memory elements: COUNT, IS_LEGAL, COUNT_READY, BITS_SEEN, SEEN_ZERO, SEEN_TRAILING. Using the proposed testability measures by using p_i = 1, the variable SEEN_ZERO gives the lowest testability value. Consequently, it is considered as the best candidate to be included in the scan register. For the rest of the objects, the following order is deduced: SEEN_TRAILING, COUNT, BITS_SEEN, COUNT_READY and IS_LEGAL.

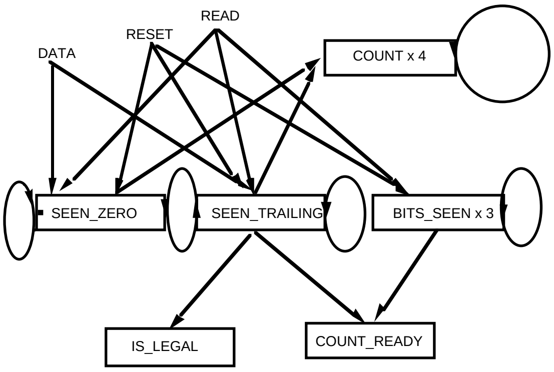


Figure 2: S-graph for a 16-bit Counter

In conclusion, a new Design for Testability approach was presented. The approach operates at the behavioural specification of today's complex system design. The approaches takes fully advantage of known methods that detect weak testable descriptions. The approach is under validation on several examples of complex designs.

References

1. J. Carletta, C. Papachristou: *Behavioral Testability Insertion for Datapath/Controller Circuits*, Journal of Electronic Testing, Vol.11, pp.9, 1997.
2. C. Aktouf, H. Fleury, C. Robach: *High Level Scan Insertion Using VHDL-based ITC Benchmarks*. To appear in Design and Test magazine, July 2000.
3. P. Kalla, M. Ciesjelski: *A Comprehensive Approach to the Partial Scan Problem using Implicit State Enumeration*, Proc. of International Test Conference, 1998.
4. E. Larsson: *High-Level Testability Analysis and Enhancement Techniques*, Ph.D., Linkôping Univ., 1998.
5. T. Orenstein, Z .Kohavi, I. Pomeranz: *Optimal Algorithm for Cycle Breaking*, JETTA, Vol.7, No.1/2, pp.71, 1995.