PathFinder

Final Project (ECE 300)

Nisar Ahmed and Vamsi Krishna
Dept. of Electrical & Computer Engineering
University of Connecticut
{nisar,vamsi.kundeti}@engr.uconn.edu

I. Introduction

The deep-submicron (DSM) effects are becoming more prominent with shrinking technology, thereby increasing the probability of timing-related defects. The transition fault and path delay fault testing together provide a relatively good coverage for delay-induced defects. Path delay model targets the cumulative delay through the entire list of gates in a pre-defined path while the transition fault model targets each gate output in the design for a slow-to-rise and slow-to-fall delay fault. However, transition fault model is widely used due to reasonable fault list (equal to twice the number of gates) as the number of paths exponential increase with primary inputs (including scan flip-flops).

To perform a transition fault test, a pattern pair *(V1,V2)* is applied to the circuit-under-test (CUT). Pattern *V1* is termed as the initialization pattern and V2 as the launch pattern. *V2* launches the signal transition ($0 \rightarrow 1$ or $1 \rightarrow 0$) at the desired node. The response of the CUT to the pattern *V2* must be captured at functional speed (rated clock period). Depending on how the transition is launched and captured, there are two transition fault pattern generation methods. In the first method, referred to as launch-off-shift (LOS), the transition at the gate output is launched in the last shift cycle during the shift operation. The transition is launched from the scan-in pin (SD) of any flip-flop in the scan chain. This activates the required transition at the target gate terminal which is propagated and captured through the functional path at an observable point (D) of any flip-flop in the scan chain. The second transition fault test method is referred to as launch-off-capture (LOC) method. In this method, the transition is launched and captured through the functional pin (D) of any flip-flop in the scan chain. The launch pattern *V2* depends on the functional response of the initialization vector *V1*.

Transition fault test patterns generated using commercial ATPG tools provide only the fault coverage information. They do not provide any information regarding the path exercised to test a particular target fault site. Therefore, in this project, we develop a tool referred to as the *PathFinder* which traces all the paths exercised by each transition fault test pattern. Some of the many possible applications of this tool are listed as follows: 1) To trace the least slack path exercised by a pattern 2) to perform analysis of detectable delay defect size based on least slack path affected through a fault site and 3) pattern fail diagnosis based on the paths terminating at the failed scan chain element.

II. Problem Definition

Transition fault test patterns (LOS or LOC) can be generated using any ATPG tool. Each pattern only contains the following information: 1) scan chain load pattern (*V1*) and 2) scan chain unload pattern (expected results after final capture). Therefore, to trace the paths exercised inside each pattern, a simulation engine is required to determine what gates in the design are actually switching during pattern application. However, pattern simulation is a slow process and our objective is to use an alternative technique and avoid simulation. We, breakdown the path tracing problem into two sub-problems:

1) **Problem 1:** Identification of all transition fault sites detected by each pattern. This provides information of all gates switching inside the circuit for each respective pattern.
2) **Problem 2:** Trace all paths through the above fault sites using the design connectivity information for each pattern.

III. Framework

In order to identify all the transition fault sites detected by each pattern (*Problem 1*), we utilize the fault simulation engine of the ATPG tool. Figure 1 shows the flow of detected fault site per pattern. Initially, ATPG is performed for

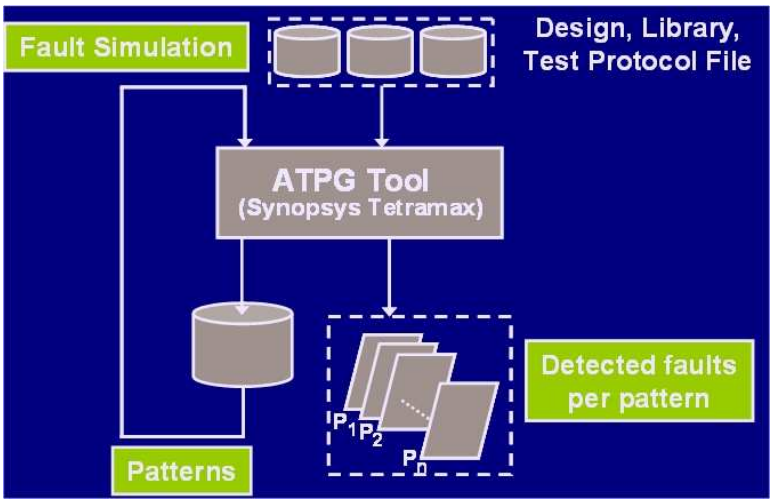

Fig. 1. Identification of faults detected by each pattern.

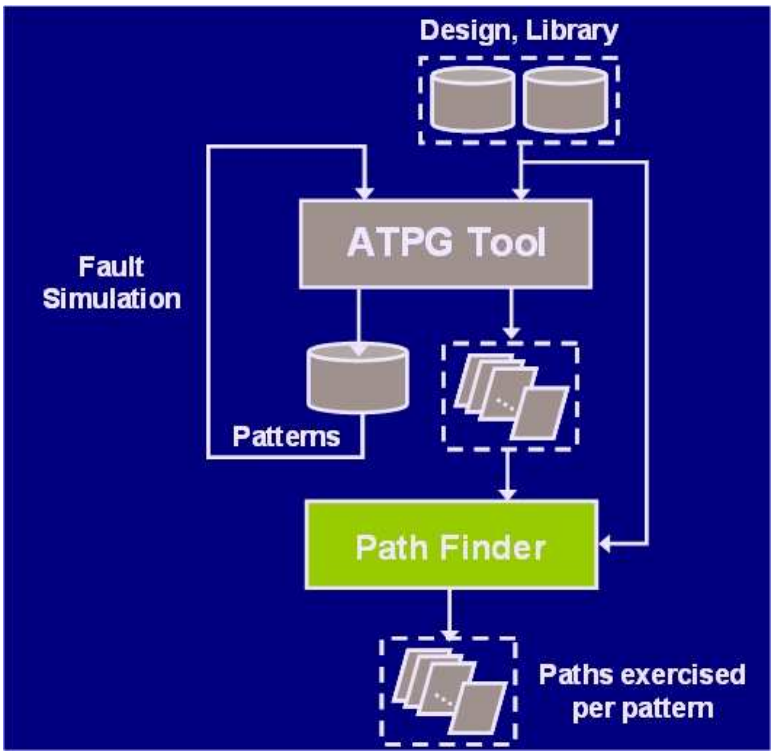

Fig. 2. Framework using PathFinder tool to trace all the paths exercised in each transition fault test pattern.

all the fault sites and all the generated patterns are stored. Later, each pattern is fault simulated individually without fault dropping (for the entire fault list without removing faults detected by previous patterns) and the detected faults by each pattern are stored in separate files. Figure 2 shows the entire framework. After identifying the transition faults detected by each pattern, the *PathFinder* tool uses this information to trace all the paths (through these detected faults) exercised in the design.