Probabilistic and Simulation-Based Masked-BIST Implementation

F. Guerreiro, J.M. Fernandes, M.B. Santos, A.L. Oliveira, I.M. Teixeira, J.P. Teixeira
IST / INESC-ID, R. Alves Redol, 9, 1000-029 Lisboa, Portugal
jct@inesc-id.pt

Abstract

Among the high-quality BIST solutions for digital systems (reseeding, bit flipping, etc) the masked-based (m-BIST) solution has been proposed, leading to the identification (at RTL) of hard functionality, accessibility problems and functional test generation which may be reused for production test. Mask generation is not a trivial problem for sequential modules. In this paper, a novel, hybrid approach for m-BIST is proposed, combining simulation and probabilistic (S-based and P-based) techniques for automatic mask generation. Both controllability and observability metrics are computed. Two proprietary tools, VeriDOS and Ascopa, are introduced for m-BIST implementation. A limited subset of feedback registers is identified at RTL for partial scan BIST solutions. A ITC'99 benchmark circuit (b13) is used as test vehicle. Results are ascertained by fault simulation finally at structural level, leading to high single line stuck-at fault coverage results.

1. Introduction

BIST (Built-In Self Test) is a key DfT technique for digital systems in emerging nanometer technologies [1]. However, its usefulness strongly depends on its ability to detect (and, eventually, diagnose) physical defects associated with new node technologies. High-quality BIST takes into account test effectiveness, overhead, time and power [2]. In order to uncover random pattern-resistant faults, selected determinism needs to be injected in built-in test patterns to be applied to the MUT (Module Under Test) [3]. Among the proposed approaches (e.g., LFSR-reseeding, bit flipping [4], bit fixing [5]) the m-BIST (or masked-based BIST) approach [6] [7] has several advantages regarding test quality: test preparation at RT-level, high defects coverage at structural (logic) level (allowing test reuse) [8], short test time, and test power typically lower than the test power associated with PR (pseudo-random) test stimuli application [9]. The key added-value of m-BIST is the possibility of defining, at RTL, functionally-dependent partially specified input vectors (referred as *mask*s) which can significantly enhance the accessibility of hard to observe functionality (referred as *dark corner*s). The m-BIST approach was initially proposed as a simulation based (S-based) technique, for which a low cost RTL fault simulation enables the identification of *dark corners* [8].
However, mask generation needed to be performed manually, which jeopardizes any practical application of the methodology. More recently, a probabilistic-based (P-based) technique to m-BIST has been introduced [10]. The purpose of this paper is to propose a more general implementation of the m-BIST approach, selectively using the S and P-based techniques to allow automatic mask generation in a cost-effective way. Mask generation was, so far, completely based on controllability data. Now, both controllability and observability metrics are considered. Two proprietary tools, VeriDOS (S-based) [11] and ASCOPA (P-based) [10] are used. An ITC'99 sequential benchmark circuit, the b013 Torino benchmark [12][13] is used as a test vehicle to discuss implementation details and to identify directions for future research. Hardware implementation of the MMIC block – the Mask-based Multiplexer Interface Circuit (such as the ones presented in [2]) are out of the scope of this paper, and will be presented elsewhere. Experiments show that m-BIST can effectively be implemented with a partial scan technique. S-based results converge to P–based results, as the number of PR test vectors increase. The combination of S-based and P-based techniques allow us to first deal with controllability
problems, and then with observability problems. The paper is organized as follows: in section 2 previous work on simulation based testability metrics is reviewed. In section 3 the new probabilistic testability analysis and automatic mask generator tool is presented. In section 4 a case study is presented, both types of testability analysis (S-based and P-based) are compared, BIST preparation is carried out and simulation results presented. Finally, section 5 summarizes the work and describes the conclusions.

2. S-based m-BIST / VeriDOS

An additional functionality can easily be added to a commercial RTL simulator, using the programming language interface (PLI), in order to allow the simulator to give precious information regarding RTL bit fault activation and detection [11]. Applying a random test

pattern to a given MUT, the percentage of simulation time that each bit (of a given RTL variable) remains at each logic level, '0' or '1', can be viewed as a natural ***controllability*** measure. If the bit is forced to a certain logic value, the number of detections, at observable outputs, can be viewed as a ***detectability*** measure, in reality a testability measure that combines both controllability and observability. These testability values, obtained by simulation, strongly depend on the input vectors applied to the MUT. Hence, they can only be analyzed as testability metrics if a large set of random vectors is applied. This is the case when a BIST solution is envisaged, fro which a pseudo-random (PR) test pattern generator (TPG) is built-in, and run at speed.

More important than quantifying controllability values for individual RTL bits is to evaluate the exercise of the conditions in the code. In order to be able to count the number of times a condition is satisfied, the VeriDOS tool requires that each condition must be a Boolean variable. Each auxiliary variable have a continuous value assigned that correspond to the corresponding condition under test. This procedure allows monitoring the condition activation. In order to be able to evaluate the condition testability the argument of the condition is replace with the Boolean variable. Doing this replacement it becomes possible to disable the condition or to force it to true. VeriDOS, for conditional constructs inhibits and forces the execution of each CASE possibility and forces each IF/ELSE condition to both possibilities. Two testability metrics are defined for each branch:

Branch Controllability:

$$COi = COi(bi) = \begin{cases} n_{ai}/n & , n_{ai} < n \\ 1 & , n_{ai} \geq n \end{cases} \quad (5)$$

where n_{ai} is the number of activations of branch b_i

Branch Detectability:

$$DOi = DOi(bi) = \begin{cases} n_{di}/n & , n_{di} < n \\ 1 & , n_{di} \geq n \end{cases} \quad (6)$$

where n_{di} is the number the non execution of branch b_i was detected. These metrics, in the interval [0, 1], reflect the fraction of success in achieving *n* fault activations and *n* fault detections, respectively. Multiple activation / detection is important to finally achieve high test effectiveness, when the physical structure (and not the RTL behavioral description) is tested.

These S-based testability metrics can thus rewardingly be used to increase structural fault coverage [8]. The proposed methodology for increasing testability in the *dark corner*s, where low COi or DOi is identified, was based on the application of *mask*s: partially defined input vectors that only force the primary inputs required for the corresponding *dark corners* activation or detection. However, the proposed *mask* generation process was done manually and the approach targeted only combinational circuits. When sequential MUT are under consideration, both primary and secondary inputs (PI, SI, respectively) need to be taken into account. In particular, state variables (the SI) are expected to bring testability problems, thus leading to the use of DfT (Design for Testability) techniques, like test-per-scan. However, not *all* SI are expected to be difficult to activate and to detect. The more the masks (forcing few PI and SI bits) are loosely deterministic, the less controllability of SI is required. This can save significant resources in terms of test overhead, test time and test power.

3. P-based m-BIST / ASCOPA

3.1 Testability metrics

The recently proposed probabilistic analysis tool [9], named *Ascopa* – **A**utomatic Static **C**ontrolability / **O**bservability **P**robabilistic **A**nalysis tool, accepts as entry the Verilog subset based on the IEEE 1364 1995 and IEEE 1364.1 1999 norms [14][15]. After the Verilog parsing, each node of the Control Flow Graph (CFG) is associated with a Data Flow Graph (DFG). Each node of the CFG is associated with a Verilog statement. Two types of nodes are considered: arithmetic expression nodes and Boolean expression nodes. Each node is associated to a particular Verilog construct in the code, such as an IF, CASE or CASETAG. A variable dependence graph (at the bit level) is created concurrently while traversing the Verilog description. As this dependence graph is built, the condition paths where each bit variable is assigned are identified without loosing the hierarchical information, keeping a direct connection to the lines of the RTL description.

After creating the variable dependence graph, *Ascopa* solves the Chapman-Kolmogorov equations that describe the steady state behaviour of the circuit, and calculates the probability P_i of logic "1", associated with each state of the Markov chain, for each bit (*i*) of the RTL description. When performing this calculation, *Ascopa* makes the assumption that bit probabilities are independent and computes the probability of each state as the product of the probabilities of the bits of the state register. This assumption can be a source of errors, dependent on the states codification. In order to minimize these errors, probabilistic testability can be computed using pairs of bits[16]. After convergence, P_i measure the controllability of each node to "1" directly and $1\text{-}P_i$ is a metric of the controllability to "0".

Having obtained the probabilistic controllability results, observability is computed using an implementation of the principle presented in [1]. The computation of observability is performed in reverse topological order, using a backward propagation process through the dependency variables graph, starting from every output

bit. A bit variable x_i is said to be observed if assigning a value of q to x_i instead of p will cause a change in some observable bit variable y_j from value k to value l. The probability of this observation is denoted as $O_{y_j:k,l}(x_i : p, q)$ and is computed differently if x_i is a PI or an internal variable.

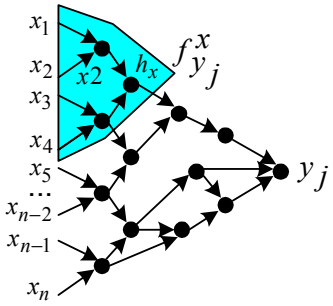

Figure 1- Scheme for a generic output y_j.

Case 1: the observed variable, x_i, is a PI

In this case the value for the observability is computed as:

$$O_{y_j:k,l}(x_i : p, q) = \sum_{k \neq l} P(f_{y_j=k}(x_i = p) \wedge f_{y_j=l}(x_i = q)) \quad (1)$$

with $k, l, p, q \in [0,1]$ and $p \neq q$.

Every x_i transition can potentially be observed by every output bit variable. Thus $O_{y_j:k,l}(x_i : p, q)$ is computed for each PO. The observability values are estimated conservatively as:

$$O(x : p, q) = \max_{j \in \{1,\ldots,m\}} O_{y_j}(x : p, q)$$

By symmetry, $O(x : p, q) = O(x : q, p)$.

Case 2: the observed variable, x_i, is internal

Let x be the internal node, and h_x a function that represents the dependence of x in terms of the PI's and registers. Considering an output y_j to observe x, with function f_{y_j}, let $f^x_{y_j}$ be the function at node y_j in terms of PI's, registers and internal node x, as represented in Figure 1:

$$f^x_{y_j} = f_{y_j}(x_1, x_{i-1}, x, x_{i+1}, \ldots, x_n)$$

In this case the value for the observability is computed as:

$$O_{y_j:k,l}(x : p, q) = \sum_{k \neq l} \frac{P(f^x_{y_j=k}(x = p) \wedge f^x_{y_j=l}(x = q) \wedge h_{x=p})}{P(h_{x=p})}$$

with $k, l, p, q \in [0,1]$ and $p \neq q$.

The term $h_{x=p}$ is included in order to calculate the observability conditioned to the fact that node x has the value p (Bayes formula). There is no symmetry for this case and $O(x : p, q) = O(x : q, p)$ may be no longer true.

3.2 m-BIST with *Ascopa*

Besides probabilistic controllability and observability metrics evaluation, *Ascopa* automatically generates the masks [10] required for each condition activation. This mask generation process is carried out early when the dependence graph is built. A mask can be generated for each condition identified in the RTL code. At present, the selection of the masks to use is made manually, based on the testability metrics values generated by *Ascopa*. Other criteria are under evaluation. *Ascopa* mask generation an be used to enhance the functional test quality, to identify design flaws and to solve controllability and observability problems. If low controllability of one bit is identified in the testability analysis step, the masks that activate conditions where the bit is assigned must be considered (taking into account the required value). If low observability is diagnosed in the testability analysis process, he masks that activate conditions where the low observability bit figures in the right-hand-side of the assignments must be considered. However, in some situations, a sequence of masks is required, to activate a condition and observe the activated functionality. This functionality is intended to be included in *Ascopa* in the future. For combinational MUT, each mask is applied *n* times, in order to fully scrutinize the synthesized physical structure. For sequential MUT, where bits of PI and SI inputs are forced, masks will also be applied *n* times. Emphasis is put on differentiating controllability from observability problems.

4. Case Study. S and P Results

The b013 (originally an interface circuit to meteo sensors [13]) as one of ITC'99 benchmark circuits [12], described at RTL and at logic level (synthesized with a commercial synthesis tool to a Verilog logic description). Its top level description is depicted in Figure 2. Ten of the 53 registers (identified at RT-level and at logic level) are synchronization registers for the 10 PO lines. The b13 description encompasses 4 FSMs (Finite State Machines).

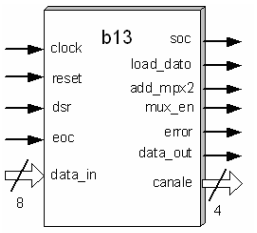

Figure 2 – b13 benchmark MUT.

After the probabilistic based testability analysis and mask generation with *Ascop*a, RTL fault simulation with *VeriDOS* is carried out, using a PR test pattern for the PI (except the Reset and Clock lines), with 5k, 25k and 50k input vectors, in order to evaluate the eventual need of additional masks. Despite the good correlation between probabilistic and simulation testability metrics (as shown for this example in Tables 1 and 2), significant differences may occur for some bits. This may be due to the fact that RTL fault simulation is performed with a limited set of input vectors; thus, the experiment may, e.g., lead to zero controllability for a given RTL condition (CASETAG in line 220 of the Verilog description) and 13% controllability metric value, computed by *Ascopa.* RTL simulation allows the identification of testability problems uncovered by probabilistic analysis. Additionally, these differences can also be due to the assumption, referred in section 3, of independence amount bit probabilities. The errors caused by this assumption can be significantly reduced when probabilistic testability is computed using pairs of bits [16]. Table 1 compares the conditions controllability evaluation performed by *Ascopa* and by *VeriDO*S. Only the three highlighted conditions were found difficult to satisfy by the simulation vectors and were not identified as difficult by *Ascop*a. Table 2 compares the b13 bits controllability (excluding primary inputs) values computed with *Ascopa* and with *VeriDO*S. A threshold of 0.05 and 0.95 was considered for difficult to control bits (to "1" and to "0" values, respectively). The bits with low simulation controllability that are not identified by *Ascopa* are highlighted. The table shows a good correlation between simulation and probabilistic controllability evaluation especially for simulations with large number of vectors. Table 3 shows the automatically generated, masks (each column is a mask) for the conditions identified by *Ascopa* and the additional three identified by *VeriDOS* as required for these vectors. Each row in Table 1 corresponds to a state variable or a PO. The 20 masks, listed in Table 3, are the only ones required for the 24 difficult conditions listed in Table 1, since 4 pairs of masks were compatible and thus could be collapsed.

Type of condition	verilog line nbr	Ascopa	veridos 5k
IF	245	0,00	0,00
CASETAG	243	0,00	0,01
CASETAG	386	0,00	0,08
CASETAG	381	0,00	0,08
IF	301	0,00	0,01
CASETAG	376	0,00	0,08
CASETAG	366	0,00	0,08
CASETAG	371	0,00	0,08
CASETAG	361	0,00	0,08
CASETAG	356	0,00	0,08
IF	270	0,00	0,00
IF	299	0,00	0,00
CASETAG	391	0,00	0,03
CASETAG	351	0,00	0,08
CASETAG	346	0,00	0,29
IF	312	0,00	0,00
CASETAG	257	0,00	0,00
CASETAG	263	0,00	0,00
IF	132	0,00	0,10
IF	343	0,01	0,01
IF	150	0,03	0,01
IF	206	0,06	1,00
IF	116	0,07	0,51
CASETAG	127	0,10	0,11
CASETAG	144	0,11	0,11
CASETAG	109	0,11	0,11
CASETAG	139	0,12	0,11
CASETAG	100	0,13	0,11
CASETAG	220	0,13	0,00
CASETAG	114	0,14	0,22
CASETAG	148	0,14	0,11
IF	199	0,15	1,00
CASETAG	105	0,16	0,11
IF	314	0,19	0,51
CASETAG	192	0,20	0,00
IF	182	0,21	0,22
CASETAG	197	0,27	0,99
CASETAG	180	0,40	0,01
IF	341	0,98	0,79
CASETAG	268	0,99	0,99

Table 1 – Controllability of RTL *conditions*.

Signal	Ascopa	veridos 5k	veridos 25k	veridos 50k
tx_end[0]	0,00	0,00	0,00	0,00
load[0]	0,00	0,00	0,00	0,00
tx_conta[6]	0,03	0,27	0,17	0,20
out_reg[0]	0,15	0,52	0,52	0,48
out_reg[1]	0,15	0,04	0,38	0,52
out_reg[2]	0,15	0,52	0,71	0,61
out_reg[3]	0,15	0,48	0,42	0,33
out_reg[4]	0,15	0,99	0,48	0,37
out_reg[5]	0,15	0,99	0,67	0,67
out_reg[6]	0,15	0,00	0,57	0,48
out_reg[7]	0,15	0,00	0,38	0,52
send[0]	0,28	0,00	0,00	0,00
canale[3]	0,30	0,11	0,11	0,11
conta_tmp[3]	0,30	0,11	0,11	0,11
tx_conta[5]	0,34	0,27	0,17	0,20
next_bit[3]	0,22	0,13	0,08	0,09
S2[0]	0,33	0,00	0,00	0,00
S2[1]	0,41	0,99	1,00	1,00
load_dato[0]	0,41	0,11	0,11	0,11
next_bit[2]	0,36	0,25	0,17	0,19
next_bit[0]	0,40	0,33	0,21	0,24
tx_conta[4]	0,45	0,31	0,20	0,23
S1[2]	0,46	0,55	0,55	0,55
S1[1]	0,47	0,45	0,45	0,45
canale[0]	0,47	0,44	0,45	0,45
tx_conta[3]	0,48	0,32	0,21	0,24
conta_tmp[0]	0,48	0,44	0,45	0,45
tx_conta[2]	0,49	0,34	0,22	0,25
canale[2]	0,48	0,44	0,44	0,44
next_bit[1]	0,49	0,28	0,18	0,21
tx_conta[1]	0,50	0,34	0,22	0,25
conta_tmp[2]	0,49	0,44	0,44	0,44
canale[1]	0,49	0,45	0,45	0,44
tx_conta[0]	0,50	0,61	0,39	0,41
conta_tmp[1]	0,49	0,45	0,45	0,44
mpx[0]	0,50	0,48	0,29	0,32
soc[0]	0,53	0,33	0,33	0,33
send_data[0]	0,53	0,22	0,22	0,22
S1[0]	0,55	0,55	0,55	0,55
error[0]	0,65	0,30	0,55	0,49
confirm[0]	0,45	0,00	0,00	0,00
mux_en[0]	0,65	0,44	0,44	0,44
shot[0]	0,57	0,00	0,00	0,00
rdy[0]	0,77	1,00	1,00	1,00
tre[0]	0,69	0,99	1,00	1,00
send_en[0]	0,98	0,69	0,44	0,50
add_mpx2[0]	0,98	0,53	0,32	0,36
itfc_state[0]	1,00	0,99	0,99	0,99
itfc_state[1]	1,00	0,99	0,99	0,99
data_out[0]	1,00	1,00	1,00	1,00

Table 2 – Controllability of RTL *variables*.

As summarized in the last column of Table 1, in order to force these remaining 20 masks, only 9 register bits (of a total of 43) are required to form a partial scan chain (referred as **switc**h, as their values need to be forced, for different masks, either to '0' or '1' value). Additionally, 6 register bits must have a common preset (that does not affect the rest of the registers) and 5 signals must have a reset possibility. Note that the generated masks lead to a loosely deterministic BIST solution, as few PI (only the reset line, in this case study) and few bit of the state variables need to be forced to '0' or '1' values. As an example, for the **tx_conta** state variable, only its most significant bit needs to be masked.

For this particular case study, test effectiveness (as evaluated by the RTL fault coverage, see [8]) strongly depends on the Reset input variable. Figure 3 shows this dependence, not only for the b13 benchmark, but also for

other Torino benchmarks. As a consequence, all simulation results reported in this paper concerning b13 have been obtained with a reset signal applied every 1000 test vectors.

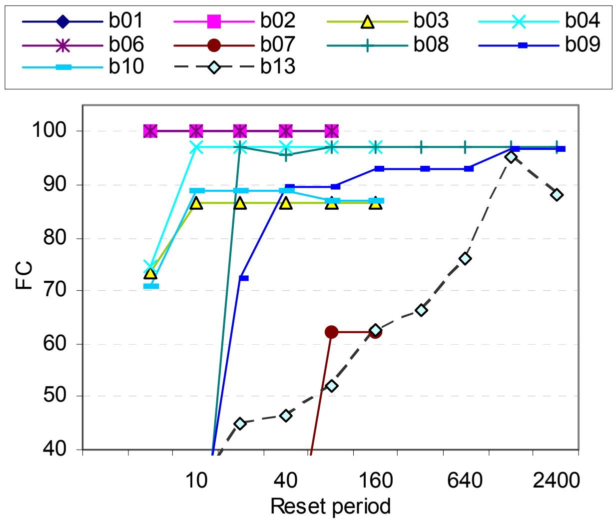

Figure 3 – RTL fault coverage results for the b13 benchmark and other Torino benchmarks, for variable reset application.

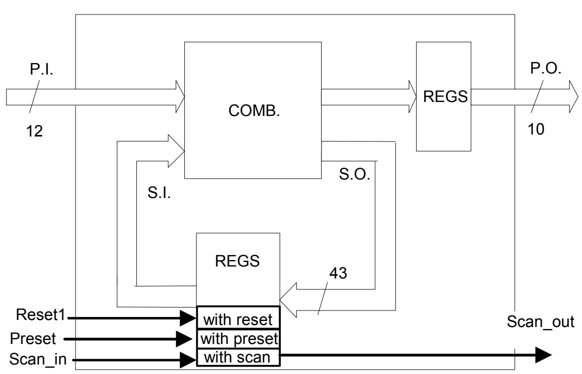

Figure 4 – Proposed m-BIST solution for the b13 benchmark.

Consequently, the S-based and P-based techniques used with m-BIST lead, for this case study, to a cost-effective BIST solution, illustrated in Figure 4. The DfT solution is obtained as follows:

1. Exclude all the register bits never forced by any mask (always "x");
2. Identify the bits that are never forced to "1". Those bits just require a common reset control (Reset1 in Figure 4, reset in Table 3).
3. Identify the bits that are never forced to "0". Those bits just require a common preset control (Preset in Figure 4, preset in Table 3).
4. Include in the partial scan chain the remaining bits, i.e., the ones that require both logic values for different masks.

	line of code:																				
	245	386	381	301	376	366	371	361	356	270	391	351	346	257	263	132	150	220	192	180	
mux_en	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
send_en	x	1	1	x	1	1	1	1	1	x	1	1	1	x	x	x	x	x	x	x	preset
canale	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
canale	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
canale	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
canale	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
load	x	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	preset
soc	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tre	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	reset
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
out_reg	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
conta_tmp	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	preset
conta_tmp	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	reset
conta_tmp	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	reset
conta_tmp	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	reset
add_mpx2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
mpx	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
next_bit	x	1	1	x	0	0	0	0	0	x	0	0	0	x	x	x	x	x	x	x	**mask**
next_bit	x	0	0	x	1	1	1	1	0	x	0	0	0	x	x	x	x	x	x	x	**mask**
next_bit	x	0	0	x	1	0	1	0	1	x	0	1	0	x	x	x	x	x	x	x	**mask**
next_bit	x	1	0	x	1	1	0	0	1	x	1	0	0	x	x	x	x	x	x	x	**mask**
tx_conta	x	1	1	x	1	1	1	1	1	x	1	1	1	x	x	x	x	x	x	x	preset
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_conta	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
send	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
confirm	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
tx_end	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x	x	x	x	x	preset
send_data	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
rdy	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	reset
itfc_state	0	x	x	x	x	x	x	x	x	1	x	x	x	0	1	x	x	x	x	x	
itfc_state	0	x	x	x	x	x	x	x	x	1	x	x	x	1	0	x	x	x	x	x	
load_dato	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
error	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
S2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	**mask**
S2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	0	**mask**
shot	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	preset
data_out	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
S1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	x	x	x	**mask**
S1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	x	x	x	**mask**
S1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	x	x	**mask**

Table 3- Generated masks (20 columns) and associated register bit values required to activate the conditions with controllability < 0.05 (as identified by *Ascopa*)

Using the 20 masks obtained as described in the previous section, controllability of the relevant state variable bits is guaranteed. However, observability may be limited, as only the PO are assumed as *observable* outputs. However, for BIST solutions, *some SO may be selected for observation* (e.g., included in a MISR for signature analysis). The P-based technique, implemented with *Ascop*a, can assist by providing the set of SI (or SO) made controllable through the mask application, but with low observability metric value. The S-based technique, implemented with *VeriDOS*, can help, by providing fault simulation results, assuming that (1) only PO, (2) *all*

outputs (PO+SO) and finally (3) *selected outputs* (PO+ the selected set of SO) are observable. In order to evaluate the test effectiveness obtained by the application of the masks, automatically generated by *Ascop*a, as measured by *structural Fault Coverage (FC) of single line stuck-at(LSA) faults,* logic synthesis has been performed with a commercial tool, using AMS 0.35µm CMOS technology. As *VeriDOS* is a mixed-level fault simulator, it has been used for FC computation. Results are shown in Figure 5.

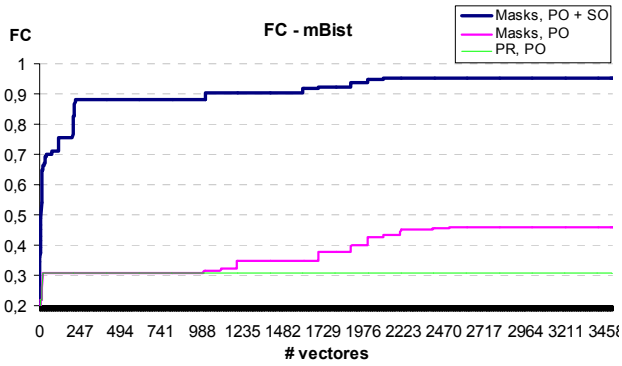

Figure 5 – b13 fault coverage results

Figure 5 shows that PR test leads to por FC results. Using the generated masks but observing only PO, insufficient FC values are still obtained. However, by observing PO+SO, almost 100% FC is reached. Using observability data generated by *Ascop*a, several SO can be removed and the same FC results hold.

5. Conclusions

In this paper, a novel, hybrid approach for the implementation of m-BIST for sequential circuits has been proposed. Both simulation and probabilistic based techniques (S-based and P-based) are used for dark corner identification, and controllability and/or observability problems identification. S-based is used for fast dark corner identification and the rough estimation of controllability and observability metrics. These estimates converge to the values obtained via probabilistic computation, when the number of test vectors increases. The P-based technique is used to identified controllability or observability problems missed by the low-cost RTL fault simulation, due to the limited test length.

Automatic mask generation, based on low values (less than 5%) controllability metrics computation using the Ascopa tool, leads to a set of masks, which allow some collapsing. A limited set of state variable bits (or registers) can thus be identified for controlling purposes, e.g., through a partial scan solution. Observability problems are also identified and, selectively, a limited subset of secondary outputs (normally not observable, but which may become observable in a BIST solution) is identified, allowing a smart, low-cost TPI (Test Point Insertion). The proposed approach is now being validated through more benchmark examples, in order to identify criteria for generalizing the results obtained with the case study, and to automate the BIST preparation process.

References

[1] M.L. Bushnel, V.D. Agrawal, "Essentials of Electronic Testing for Digital Memory and Mixed-Signal VLSI Circuits", Kluwer, 2000.

[2] J. Braga, P. Coimbrão, M. B. Santos, I.C. Teixeira and J. P. Teixeira, "On the Practical Implementation of Masked-based BIST", Proc. of the 18th. Conf. On Design of Circuits and Integrated Systems (DCIS), pp. 139-144, November, 2003.

[3] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, J. Rajski, "Logic BIST for Large Industrial Designs: Real Issues and Case Studies", Proc. IEEE Int. Test Conf. (ITC), pp. 358-367, 1999.

[4] H. Vranken, H.-J. Wunderlich, "Bit-Flipping BIST", Proc. IEEE Int. Conf. On CAD (ICCAD), pp. 337-343, 1996.

[5] N.A Touba, E.J. McCluskey, "Altering a Pseudo-random Bit Sequence for Scan-based BIST", Proc. IEEE Int. Test Conf. (ITC), pp. 167-175, 1996.

[6] M.B. Santos, F.M. Gonçalves, I.C. Teixeira and J.P. Teixeira, "RTL-Based Functional Test Generation for High Defects Coverage in Digital Systems", Journal of Electronic Testing, Theory and Application (JETTA), vol. 17, Nº 3/4, pp. 311-319, Kluwer, June/August 2001.

[7] Santos M.B., Fernandes J.M., Teixeira, I.C., Teixeira J.P., "RTL Test Pattern Generation for High Quality Loosely Deterministic BIST", Proc. of the Design Automation and Test in Europe Conference (DATE), pp. 994-999, March 2003.

[8] M.B. Santos, F.M. Gonçalves, I.C. Teixeira and J.P. Teixeira, "Implicit Functionality and Multiple Branch Coverage (IFMB): a Testability Metric for RT-Level", Proc. of the International Test Conf. (ITC), pp. 377-385, Baltimore, October, 2001.

[9] M. B. Santos, I.C. Teixeira, J. P. Teixeira, S. Manich, R. Rodriguez and J. Figueras, "RT-Level Preparation of High-Quality/Low Energy/Low-Power BIST", Proc. of the International Test Conf. (ITC), pp.814-823, Baltimore, October, 2002.

[10] J.M. Fernandes, M.B. Santos, A. Oliveira, J.P. Teixeira, "A Probabilistic Method for the Computation of Testability of RTL Constructs", Proc. of the Design Automation and Test in Europe (DATE) Conf., pp. 176-181, 2004.

[11] M. B. Santos, F.M. Gonçalves, I.C. Teixeira and J. P. Teixeira, "Defect-Oriented Verilog Fault Simulation of SoC Macros using a Stratified Fault Sampling Technique", Proc. of the IEEE VLSI Test Symp. (VTS), pp. 326-332, 1999.

[12] Scott Davidson, "ITC'99 Benchmark Circuits – Preliminary Results", Proc. IEEE International Test Conf (ITC), pp. 1125, 1999; benchmarks available at http://www.cerc.utexas.edu/itc99-benchmarks/bench.html .

[13] F. Corno, M. Sonza-Reorda, G. Squillero, "RT-level ITC'99 Benchmarks and First ATPG Results", IEEE Design & Test of Computers, pp. 44-53, July-August 2000.

[14] IEEE Standard 1364-1995, "IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language", IEEE, Inc., New York, NY, USA, October 14, 1996.

[15] IEEE Standard P1364.1/D1.4, "Draft Standard for Verilog Register Transfer Level Synthesis", IEEE, Inc., New York, NY, USA, April 26, 1999.

[16] Radu Marculescu, Diana Marculescu, and Massoud Pedram, "Probabilistic Modeling of Dependencies During Switching Activity Analysis", IEEE Trans. on Computer Aided Design of Int.Circs. and Syst., Vol. 17, Nº. 2, 1998.