CMOS Digital Circuits Test Preparation (September 2007)

Alexandre M. Valério and Prof. Marcelino Dos Santos, *DEEC, IST*

***Abstract*—The test flow that is currently used by AMI Semiconductor (AMIS) is starting to be not enough for providing an acceptable fault detection level. A new test flow is developed in partnership with AMIS, one that is optimized in both fault detection and total number of test patterns, using commercial software such as Tetramax®, from Synopsys®, the Programming Language Interface (PLI), and Verilog-XL®, from Cadence®.**

Two voltage fault models are used: Line Stuck-At (LSA), Bridging (BRI), and one current fault model: Idd quiescent (IDDQ).

With this new test flow it is possible to obtain, in average, a reduction of 36.71% in the number of voltage test vectors, and a bridging fault detection 8.65% superior, for the benchmark circuits that are tested, when compared with the test flow presently used by AMIS.

***Index Terms*—Test flow, Fault models, PLI, Fault simulation, Fault list reduction, ATPG.**

I. Introduction

The test flow currently used by AMIS is starting to be insufficient for the fault detection level that is required for current technologies. AMIS's test flow (ATF) uses one voltage fault model: LSA, and one current fault model: IDDQ.

The LSA fault model is traditionally used in test preparation, but no longer assures the required level of fault detection [1][2][3][4].

With this in mind, the new test flow (NTF) uses several fault models: LSA and BRI voltage fault models, and IDDQ current fault model. This will increase the precision and robustness of the test [5]. ATF uses N-detection LSA test patterns (each fault is detected N times) in order to optimize the fault detection, but the resolution of today's technologies causes the increase of faults of other fault models, such as bridging faults [6][7].

The NTF, besides using several fault models, also uses the concept of fault simulation: before performing Automatic Test Pattern Generation (ATPG) for a certain type of fault model, fault simulation is executed using test patterns obtained for other fault models. This will reduce the number of faults that need test patterns generated for them, and consequently will reduce the total number of generated test patterns, in comparison with ATF. In addition, since the LSA patterns generated with the NTF are simple detection patterns and the ones in the ATF are N-detection patterns, the total number of generated test patterns with the NTF should be smaller in comparison with the ATF.

In order to detect as much faults as possible in circuit designs, Tetramax® uses three ATPG modes:

--Basic scan, an efficient combinational-only mode for full-scan designs.

--Fast-sequential, for limited support of partial-scan designs.

--Full-sequential, for maximum test coverage in partial-scan designs.

With the version of the Tetramax® used for this article (Y-2006.06), Full-sequential BRI ATPG and fault simulation are not supported, which will limit the BRI fault coverage [8].

Although it is not referenced in [8], Tetramax® also does not support IDDQ Full-sequential fault simulation, but PROOFS (a fast, memory efficient sequential circuit fault simulator) is performed so that all LSA test patterns are used. Tetramax® supports IDDQ Full-sequential ATPG.

Whit this in mind, whenever a test is performed in Tetramax®, the best ATPG mode available for each fault model is used.

II. Benchmark circuits synthesis

Politecnico di Torino provides the benchmark circuits used in this paper, and the circuits are available as RTL description, in VHDL format. Since Verilog-XL® needs a Verilog description of the circuit under test, and Tetramax® needs a structural description of the circuit so that it can properly detect and use all nodes and gates, it is necessary to generate a structural Verilog description of the circuit.

Synopsys® Design Vision® is used to do this. Design Vision® is executed with the "-db_mode" option, to use the old database format and allow the use of commands related to scan chain insertion, like "insert_scan".

C35 technology libraries from AMS® are used to synthesize the circuits.

No optimizations are used regarding area used for layout, and mapping effort is used in its lowest setting: medium, to avoid optimizations and loss of nodes and gates present in the

A. M. Valério is a student of Electrical and Computer Engineering in Instituto Superior Técnico, Lisbon, Portugal. (e-mail: alex.evol@gmail.com).

Prof. M. dos Santos is an assistant teacher for the Electrical and Computer Engineering Department, Instituto Superior Técnico, Lisbon, Portugal, and a member of INESC-ID Lisbon, Portugal. (e-mail: marcelino.santos@ist.utl.pt).

initial circuit description. Dandling gates and nodes are not removed, in order to allow a higher number of faults for each circuit.

The “industrial worst condition” option is used to synthesize the circuits, and the clock is defined as a 50% duty cycle, square wave and a 100 ns period clock.

Scan chains are inserted automatically, using the “insert_scan” command.

The data relative to clock and scan chains is saved to a STIL format file, with “.spf” extension, that is later used in Tetramax®, in the “Build” part of the test flow.

Finally, the structural Verilog circuit description is saved to file after the synthesis, with Design Vision®’s automatic renaming of nodes to the Verilog standard.

III. TEST PREPARATION METHODOLIGIES

A. *AMIS’s Test Flow*

AMIS uses a test flow that does N-detection for LSA faults, five times each fault, and generates IDDQ test patterns with prior fault simulation with simple or 1-detection LSA test patterns, so that few IDDQ test patterns are generated through ATPG. According to [8], a set of 10 to 20 patterns (each pattern may contain one or more vectors) is considered acceptable, since IDDQ testing is very expensive.

With this test flow only the LSA and IDDQ fault models are considered, since it is commonly considered that LSA test patterns detect the majority of all faults, through a direct way (LSA patterns detecting LSA faults) or an indirect way (LSA patterns detecting other types of faults). This is true only to a certain extent since the LSA fault model does not model all possible defects, and the level of indirect fault detection depends on the way the integrated circuit (IC) is designed, the technology used synthesize it (higher resolution technologies are prone to have more bridging faults than lower resolution ones), just to mention a few.

Studies show that detecting faults with multiple test patterns helps catching defect that cannot be modeled with standard fault models, such as transistor stuck-open or cell-level faults, but the use of a dedicated fault model is more efficient than relying in brute force [8]. The N-detection in Tetramax® is also limited: it is not available for IDDQ and Path Delay faults models, and distributed processing and fault simulation are not supported (version Y-2006.06 of Tetramax®).

The ATF consists in the following steps:

--LSA ATPG with N-detection, five times for each fault.

--BRI fault simulation with multiple detection LSA test patterns.

--LSA ATPG with simple detection or 1-detection, followed by IDDQ fault simulation with simple LSA test patterns, and IDDQ ATPG for the remaining faults.

--BRI fault simulation with IDDQ test patterns.

--BRI fault simulation with simple LSA test patterns.

The BRI fault simulations are only performed in order to compare this test flow’s BRI fault detection level with the NTF BRI fault detection. In a real case scenario, the BRI fault simulations are not performed.

Figure 1 shows a representation of the ATF.

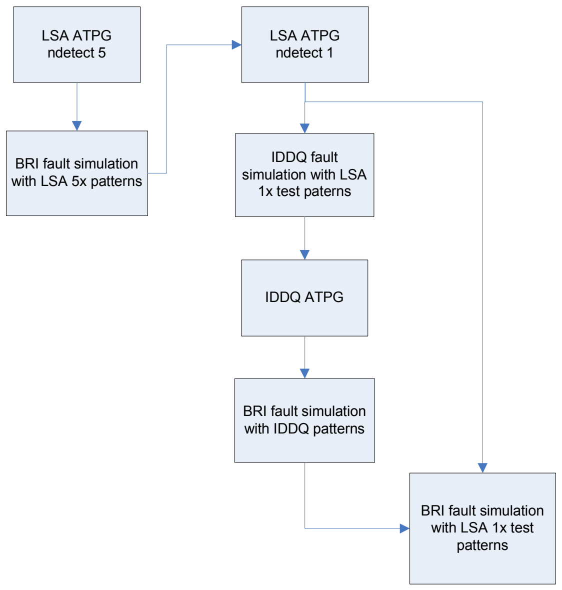
LSA ATPG ndetect 5
BRI fault simulation with LSA 5x patterns
LSA ATPG ndetect 1
IDDQ fault simulation with LSA 1x test paterns
IDDQ ATPG
BRI fault simulation with IDDQ patterns
BRI fault simulation with LSA 1x test patterns

Figure 1 – AMIS’s test flow

B. *Proposed methodology*

The objective for the new test flow is to obtain a test flow that takes in consideration LSA, BRI and IDDQ fault models, and give as result a number of test patterns smaller than the one obtained with the ATF, while achieving higher fault coverage. It was by suggestion from AMIS that the BRI fault model was chosen as the fault model to be added.

So that this test flow results in a reduced number of test patterns as much as possible, it is necessary to perform fault simulation whenever possible. In this way, faults of a certain fault model that are detected by test patterns generated for other fault models are no longer taken in consideration for the ATPG for the fault model in question, reducing the number of the generated test patterns. In addition, LSA ATPG is performed with simple detection of LSA faults.

The NTF is presented in Figure 2, and is composed by the following steps:

--LSA ATPG with simple or 1-detection, IDDQ fault simulation with LSA test patterns, and IDDQ ATPG for the remaining faults. This step is referenced as “S2I” (from Stuck-at to IDDQ).

--Since it is not possible to obtain a real BRI fault list for each circuit, with the available software, a random fault list is generated considering all valid nodes of a circuit. This step is referenced as “VER_RAND” (from Verilog Random).

--BRI fault simulation with LSA test patterns. This step is called “S2B” (from Stuck-at to Bridging).

-- BRI fault simulation with IDDQ test patterns. This step is called “I2B” (from IDDQ to Bridging).

--BRI ATPG for the remaining BRI faults. This step is called “ATPG-BRI” (from BRI ATPG).

All steps are executed with Tetramax®, except for

VER_RAND and I2B that are executed with Verilog-XL® with extra functionalities added with PLI.

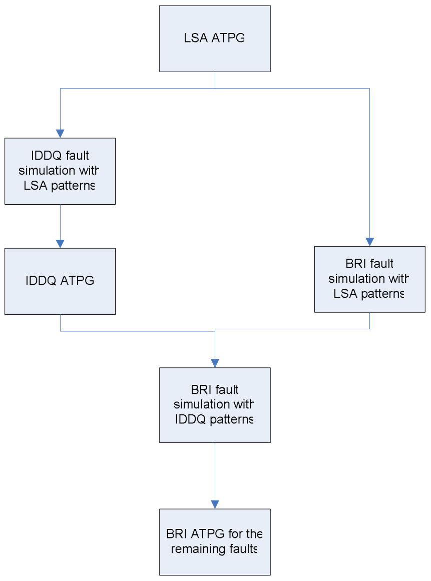
LSA ATPG
IDDQ fault simulation with LSA patterns
IDDQ ATPG
BRI fault simulation with LSA patterns
BRI fault simulation with IDDQ patterns
BRI ATPG for the remaining faults

Figure 2 - New test flow

S2I

Tetramax® is used to perform all actions for this step. Having a Verilog structural description of a circuit and a STIL file with data on the clock and scan chains, a script is used to execute the necessary commands for LSA ATPG, IDDQ fault simulations with LSA patterns and IDDQ ATPG for the remaining IDDQ faults.

The script performs the following steps:

--read the technology libraries and the circuit description.

--perform "Build" for the circuit, in which Tetramax® identifies the parts of the circuit necessary for ATPG, removes their hierarchy and places them in internal memory.

--run the design rules check for the circuit description, based on the clock and scan chain information in the STIL file.

--set options for the ATPG, such as Basic scan, Fast-sequential and Full-sequential generation, and high merge of test patterns.

--add all LSA faults, and run LSA ATPG.

--write the LSA test patterns to file.

--change the fault model to IDDQ, and add all IDDQ faults.

--perform IDDQ fault simulation with LSA patterns is performed, removing from the fault list the IDDQ faults detected by LSA patterns.

--run IDDQ ATPG, with the same generation and merging options as the LSA ATPG.

--write the IDDQ test patterns to file.

VER_RAND

This step is executed with Verilog-XL® with extra functionalities added through PLI.

Tetramax® cannot add bridging faults automatically, and Verilog-XL® by itself cannot generate random bridging faults, so C user functions and PLI functions are used to add this functionality to the Verilog simulator.

While generating random bridging faults, all nodes of all modules are considered, but only simulated nets are used (the net formed by the output node of a gate, a wire connected to it and to an input node of another gate) and unconnected nodes are not considered for bridging faults. In addition, if the fault between generic A and B nodes already exists, the fault between B and A is not accepted, since they are equivalent.

The total number of random bridging faults to generate depends directly on the number of LSA faults, and is given by equation (1).

$$\text{No.BRI faults} = \text{No.LSA faults} \times 3.7 \,. \tag{1}$$

The number "3.7" is used because for relatively large circuits (like circuit b15) the number of LSA patterns is similar to the number of BRI patterns.

There is a limit for the number of bridging faults that can be generated for a circuit, and is given by equation (2). Through deduction, this limit corresponds to the number of combinations of N circuit nodes, 2 by 2, multiplied by 2 since a pair of nodes can be of two types of faults: ba0 or ba1. In case the requested number of bridging faults is larger than the limit of faults, only a number of faults equal to the limit are generated.

$$\text{Max No.BRI faults} = {}^{N}C_{2} \times 2 \,. \tag{2}$$

The bridging fault list that is generated is saved to file in a format that can be read back by Verilog-XL® through C and PLI functions, and to a file in the form of Tetramax® commands:

```
add faults -bridge_location nodeA nodeB -bridge [1|0] -agressor_node Second.
```

S2B

Using Tetramax®, the bridging fault list is read and bridging fault simulation with LSA test patterns is performed. Again, a script is used to do this. The main actions are the following:

--read the technology libraries and the circuit description.

--perform "Build" for the circuit, in which Tetramax® identifies the parts of the circuit necessary for ATPG, removes their hierarchy and places them in internal memory.

--run the design rules check of the circuit description, based on the clock and scan chain information in the STIL file.

--fault model is set as Bridging, the random faults list are read and the patterns to use for simulation are set as the LSA test patterns.

--BRI fault simulation is performed with the LSA patterns.

--the faults that are not detected are written to file, in the Tetramax® fault summary format, which will be read by Verilog-XL® with the help of C and PLI functions:

Type of fault | State of detection code | Victim node | Aggressor Node.

The bridging faults that are not detected by LSA patterns are used in the I2B step.

I2B

This step is executed with Verilog-XL®, with C and PLI functions that allow this simulator to read a bridging fault list from Tetramax® and verify which bridging faults are detected by IDDQ test patterns.

The fault list is read into user structures that hold the name, present logic value and the list of faults for each node of the circuit.

Three PLI user tasks are added to the simulator: "$iddq_init", "$iddq_bri" and "$iddq_close", that initialize the necessary structures for holding information, that check which bridging faults are detected at the end of a pattern, and that save the fault list to file and cleans the user structures, respectively.

These user tasks are placed in the Verilog IDDQ pattern testbench generated in the end of S2I. "$iddq_init" is placed inside the "initial" block, "$iddq_bri" is placed inside the "event IDDQ" description and "$iddq_close" is placed before the "$finish" system call at the end of the testbench.

The Tetramax® generated Verilog testbench already has IDDQ test definitions, but they are not used since they are for Synopsys® PowerFault®, and this software is not used in this test flow.

A trigger (PLI function) is added to every node directly associated to a BRI fault, so that whenever there is a change on the logic value of that node that change is saved in the user structures.

When the test patterns are being simulated, and whenever an IDDQ measurement is made, a user task ($iddq_bri) is called to check every node and its faults to see which BRI are detected. The test to see if a fault is detectable is the following: if the aggressor node has the logic value "0" ("1"), the victim node has "1" ("0") and the fault type is "ba0" ("ba1"), then the fault is detectable.

The BRI faults that are detectable are removed from the fault list.

When the simulation ends, the fault list is saved to file as Tetramax® commands, the same way as it is done in VER_RAND step.

ATPG-BRI

In this final step, the bridging fault list written at the end of I2B is used on Tetramax® so that BRI ATPG is performed for the bridging faults that are undetected by LSA patterns and by IDDQ patterns.

Just like with other steps that use Tetramax®, a script is used to execute this step. The following actions are necessary for this step:

--read the technology libraries and the circuit description.

--perform "Build" for the circuit, in which Tetramax® identifies the parts of the circuit necessary for ATPG, removes their hierarchy and places them in internal memory.

--run the design rules check of the circuit description, based on the clock and scan chain information in the STIL file.

--the fault model is set as Bridging and the random faults list are read.

--since Full-sequential ATPG is not available for the bridging fault model, only Basic scan and Fast-sequential ATPG is performed.

--run the ATPG, with high merge options.

--write the bridging test patterns to file.

IV. Results

To test and compare the new test flow with AMIS's test flow, 10 benchmark circuits from Politecnico di Torino are used.

For each circuit, the NTF is used firstly and then the ATF is used. The obtained results are the described below. Intermediate results are initially presented, that allow a comparative study of the intermediate steps of each test flow. The global results concerning the number of test vectors, the level of fault detection and the execution time for each circuit are presented afterwards. Note that the following results are relative only to detectable faults.

The number of LSA and BRI faults for each circuit is presented in Table 1. In the case of circuit b12 it is not possible to generate the required number of random faults, only 6006 are possible since there are not enough nodes for generating the 34032 faults. With just 78 nodes, only (78C2) x 2 = 6006 faults are possible.

Column "IDDQ<-LSA" in Tables 2 and 3 represent the percentage of IDDQ faults detected by LSA test patterns. This applies to all other columns of Tables 2 and 3. Note that column "BRI<-IDDQ" of Table 2 shows the percentage of BRI faults that are not detected by LSA patterns (column BRI<-LSA) and are detected by IDDQ patterns. This reasoning also applies to columns "BRI<-IDDQ" and "BRI<-LSA" of Table 3.

In Tables 2 and 3, with both test flows, the average detection of IDDQ faults by LSA patterns is 96.60%, being this way possible to conclude that LSA patterns detect the majority of IDDQ faults but it is still necessary to generate IDDQ patterns for the remaining IDDQ faults. Both test flows give the same average detection because the IDDQ fault simulation and ATPG are performed with the same conditions in both cases.

Since there is only a relatively small number of IDDQ faults that are undetected, the number of IDDQ patterns that are obtained by ATPG is also small, as can be seen on Table 4, which is as advantage since IDDQ testing is expensive.

In addition, in Tables 2 and 3, it is possible to see that the BRI fault detection with multiple detection LSA test patterns is in average 82.90%. This fault coverage is 8% greater than the obtained with simple detection LSA test patterns in the NTF.

This is because multiple detection LSA patterns are larger in number in comparison with simple detection LSA patterns, and since there are more test patterns, more BRI faults are detected.

In Tables 2 and 3, it can be seen that the BRI fault detection by LSA and IDDQ test patterns is in average 91.18% with the NTF and 94.77% with the ATF. In other words, a large percentage of BRI faults are detected by LSA and IDDQ test patterns, but that is not enough since there still is a significant percentage remaining. This justifies the use of BRI ATPG executed in the NTF.

Regarding the "BRI<-IDDQ" partial detection, it is not possible to compare both test flows since the initial number of BRI faults in each case is different.

In Table 3, it is possible to see that the BRI fault simulation with the simple detection LSA patterns detect, in average, more 7.04% BRI faults. This occurs due to the fructuous conjugation of the necessary conditions for the activation of the bridgings, and the random attributions that occur at the end of the ATPG process.

In Tables 5 and 7, it is possible to observe that in both test flows the average detection of LSA faults is about 98.69% and 100% for IDDQ faults. As for BRI faults, with the NTF an average of 98.52% of all BRI are detected, while with the ATF only 89.87% are detected. Although with the ATF we have better average detection of BRI faults by LSA and IDDQ patterns, no BRI ATPG is performed, unlike with the NTF, so the average BRI fault detection with the ATF is always lower than the one obtained with the NTF.

In Table 8, it can be seen that the total number of voltage test vectors generated with the NTF is always smaller than the number generated by the ATF. In four cases, the NTF number of voltage test vectors is 2.6 (b07, b15) and 3 (b11, b12) times smaller than the ATF voltage test vectors. It can also be seen that circuit b14 has 32 current (IDDQ) test vectors, which looks like a large number of vectors, but that is the number of vectors and each pattern can be composed by one or more vectors. In the S2I log it is can be seen that only 18 IDDQ test patterns are generated for circuit b14, which is quite acceptable.

Concerning the required processing time of each test flow for each circuit, it can be seen on Tables 9 and 10 that the required CPU processing times for each test flow are in the same order of time. Note that these values do not consider the time that is necessary to read the BRI fault list by Tetramax® and Verilog-XL® in steps S2B, I2B and ATPG-BRI, which adds a considerable amount of time to the necessary processing time for the NTF.

With this in mind, it is possible to say that the NTF will take longer to prepare the test for a given circuit, in comparison with the ATF. However, since the NTF is optimized in fault detection and number of test patterns, the extra time that is necessary to generate the test for a given circuit is acceptable, since the test preparation is performed only once and the test is performed thousands of times for thousands of IC's.

V. Conclusion

After analysis of the results, it is possible to say that the NTF is in fact more efficient than the ATF.

Although in both test flows the LSA and IDDQ fault detections are about the same, the BRI fault detection with the NTF is better, about 8,65% more in average, since BRI ATPG is performed in it.

About the total number of test patterns for each circuit, with the NTF the number of test patterns is always smaller than the number of patterns generated with the ATF. In some cases, the NTF patterns are two or three times smaller in number than the ATF ones.

This is greatly due to the fact that with the ATF N-detection LSA ATPG is performed, which results in a larger number of patterns than the number that is obtained with the NTF. However, the BRI ATPG used in the NTF adds a significant number of test patterns, so in the end the total number of voltage test patterns is usually similar between the two test flows.

As for the time needed to finish a test flow for a given circuit, it is seen that the NTF will take longer to finish than the ATF. However, since the NTF is optimized in fault detection and number of test patterns, the extra time that is needed is acceptable, since the test preparation is only performed once and the test itself is performed thousand of times for thousands of circuits.

In this way, it is possible to say that the objectives initially set for the NTF are in fact reached: it is a usable test flow for complex circuits since it uses commercial software, it is optimized in fault detection and number of test patterns, being in this way adapted to current technologies.

As future work, there are a few areas with interest:

--use real BRI fault lists with the NTF.

--add more fault models to the test flow, such as Path delay and Transition models.

--study why there are circuits in which the number of patterns with the NTF is 3 times smaller than the one obtained with the ATF, and take advantage of this information to better the test flow.

--test the NTF with newer versions of Tetramax®, where Full-sequential ATPG or fault simulation for BRI and IDDQ faults is probably already permitted by the software.

--study another new test flow to see if it's better than the NTF described here, maybe one where LSA ATPG, BRI ATPG and then IDDQ ATPG is performed, with this sequence.

--find a way to make the BRI fault list insertion in Tetramax® faster.

Acknowledgment

I would like to thank Prof. Marcelino dos Santos for all the help and motivation provided, and INESC-ID for providing the necessary software to perform the tests.

REFERENCES

[1] Kohei Miyase, Kenta Terashima, Seiji Kajihara, Xiaoqing Wen, Sudhakar M. Reddy, "On Improving Defect Coverage of Stuck-at Fault Tests," ats, pp. 216-223, 14th Asian Test Symposium (ATS'05), 2005.
[2] Arumi, Rodriguez-Montanes, Figueras, Eichenberger, Hora, Kruseman, Lousberg, Majhi, "Diagnosis of Bridging Defects Based on Current Signatures at Low Power Supply Voltages," vts, pp. 145-150, 25th IEEE VLSI Test Symmposium (VTS'07), 2007.
[3] Peter C. Maxwell, Robert C. Aitken, Vic Johansen, Inshen Chiang, "The Effectiveness of IDDQ, Functional and Scan Tests: How Many Fault Coverages Do We Need?", International Test Conference, IEEE, Paper 6.3, pp.168 a 177, 1992.
[4] Vijay R. Sar-Dessai, D. M. H. Walker, "Resistive Bridge Fault Modeling, Simulation and Test Generation", International Test Conference, 1999.
[5] David B. Lavo, "A Comprehensive Look at VLSI Fault Diagnosis", 19 de Julho de 2002.
[6] R. Rodriguez-Montanes, E. M. J. G. Bruls, J. Figueras, "Bridging Defect Resistance Measurements in a CMOS Process," International Test Conference, pp.892 a 899, 1992.
[7] E. Isern, J. Figueras, "IDDQ Test and Diagnosis of CMOS Circuits", IEEE Design & Test of Computers, IDDQ Series, pp.60-67, 1995.
[8] Tetramax® ATPG User Guide, Synopsys®, Instalation manuals of Synopsys® tools, Version Y-2006.06, June 2006.

Table 1- Number of LSA and BRI faults

Circuit	# LSA faults	# BRI faults
b04	3994	14778
b05	10044	37163
b07	2046	7570
b11	3352	12402
b12	9198	6006
b14	42598	157613
b15	38712	143234
b17	121178	448359
b18	386966	1431774
b22	158854	587760

Table 2 – Partial detection results with the NTF

Circuit	IDDQ <- LSA	BRI <- LSA	BRI <- IDDQ
b04	94,61%	63,15%	65,60%
b05	96,33%	85,20%	73,60%
b07	93,40%	76,12%	54,59%
b11	97,55%	75,70%	56,96%
b12	95,63%	85,95%	58,53%
b14	98,48%	73,46%	61,33%
b15	97,34%	74,18%	71,10%
b17	96,05%	72,50%	66,67%
b18	97,87%	73,69%	74,67%
b22	98,76%	68,73%	66,11%
Avg	96,60%	74,87%	64,92%

Table 3 – Partial detection results with the ATF

Circuit	IDDQ <- LSA	BRI <- LSA5x	BRI <- IDDQ	BRI <- LSA
b04	94,61%	68,64%	64,78%	7,86%
b05	96,33%	94,09%	74,13%	21,20%
b07	93,40%	91,28%	66,82%	8,62%
b11	97,55%	92,35%	66,71%	7,79%
b12	95,63%	84,54%	57,48%	6,37%
b14	98,48%	84,14%	64,28%	5,95%
b15	97,34%	78,83%	70,16%	1,72%
b17	96,05%	77,62%	65,87%	2,57%
b18	97,87%	79,66%	73,98%	3,86%
b22	98,76%	77,82%	66,54%	4,44%
Avg	96,60%	82,90%	67,08%	7,04%

Table 4 – Total number of test vectors with the NTF

Circuit	LSA	IDDQ	BRI
b04	51	9	121
b05	195	9	91
b07	63	18	41
b11	108	11	57
b12	175	12	31
b14	829	32	1804
b15	540	18	530
b17	549	16	1778
b18	1224	22	4205
b22	1241	11	5180

Table 5 – Total fault detection with the NTF

Circuit	LSA	IDDQ	BRI
b04	89,09%	100,00%	90,22%
b05	99,82%	100,00%	99,83%
b07	100,00%	100,00%	99,95%
b11	100,00%	100,00%	100,00%
b12	100,00%	100,00%	99,20%
b14	99,14%	100,00%	99,43%
b15	99,88%	100,00%	99,37%
b17	99,90%	100,00%	99,03%
b18	99,75%	100,00%	99,22%
b22	99,32%	100,00%	98,96%
Avg	98,69%	100,00%	98,52%

Table 6 – Total number of test vectors with the ATF

Circuit	LSA 5x	LSA	IDDQ
b04	132	51	9
b05	195	195	9
b07	208	63	18
b11	406	108	11
b12	551	175	12
b14	3605	829	32
b15	2340	540	18
b17	2205	549	16
b18	4917	1224	22
b22	5357	1241	11

Table 7 – Total fault detection with the ATF

Circuit	LSA 5x	LSA	IDDQ	BRI
b04	89,09%	89,09%	100,00%	81,28%
b05	99,82%	99,82%	100,00%	98,05%
b07	100,00%	100,00%	100,00%	94,68%
b11	100,00%	100,00%	100,00%	96,47%
b12	100,00%	100,00%	100,00%	94,37%
b14	99,15%	99,14%	100,00%	90,42%
b15	99,90%	99,88%	100,00%	85,42%
b17	99,91%	99,90%	100,00%	85,68%
b18	99,74%	99,75%	100,00%	86,41%
b22	99,31%	99,32%	100,00%	85,96%
Avg	98,69%	98,69%	100,00%	89,87%

Table 8 – Total number of voltage and current test vectors

	NTF		ATF		
Circuit	# voltage vectors (LSA + BRI)	# current vectors (IDDQ)	# voltage vectors (LSA5x + LSA1x)	# current vectors (IDDQ)	ATF/NTF voltage test patterns
b04	172	9	183	9	1,06
b05	286	9	390	9	1,36
b07	104	18	271	18	2,61
b11	165	11	514	11	3,12
b12	206	12	726	12	3,52
b14	2633	32	4434	32	1,68
b15	1070	18	2880	18	2,69
b17	2327	16	2754	16	1,18
b18	5429	22	6141	22	1,13
b22	6421	11	6598	11	1,03

Table 9 – NTF processing time

	NTF				
Circuit \ CPU processing time (s)	LSA ATPG, IDDQ fault simulation and ATPG	BRI fault simulation with LSA patterns	BRI fault simulation with IDDQ patterns	BRI ATPG	Total time
b04	1146,65	0,21	1,20	1,15	1149,21
b05	48,52	0,22	1,00	1,00	50,74
b07	581,29	0,16	0,70	0,21	582,36
b11	2,71	0,14	0,80	0,22	3,87
b12	7,71	0,30	0,50	0,31	8,82
b14	2156,67	1,60	35,70	14,79	2208,76
b15	94,19	1,87	25,50	9,13	130,69
b17	301,66	6,20	1412,20	116,42	1836,48
b18	5477,95	24,36	7484,80	458,25	13445,36
b22	7087,95	20,35	2977,20	823,76	10909,26

Table 10 – ATF processing time

	ATF		
Circuit \ CPU processing time (s)	Multiple detection LSA ATPG (5x)	LSA ATPG, IDDQ fault simulation and IDDQ ATPG	Total time
b04	1094,25	1146,65	2240,90
b05	46,14	48,52	94,66
b07	0,40	581,29	581,69
b11	0,56	2,71	3,27
b12	1,05	7,71	8,76
b14	2153,09	2156,67	4309,76
b15	94,72	94,19	188,91
b17	314,06	301,66	615,72
b18	4634,57	5477,95	10112,52
b22	6542,30	7087,95	13630,25