Politecnico di Torino

Testing and fault tolerance Hardware Assignment

Project Report

Master degree in Electronics Engineering

Professor: Matteo Sonza Reorda

Authors:

Reza Khoshzaban -s288966@studenti.polito.it
Seyedeh Neda Hariri -s270052@studenti.polito.it

Academic Year 2021/2022

Contents

CHAPTER 1

Project Spesifications

1.1 Goal of the project

Developing a synthesizable LBIST for the RI5CY processor is the goal of this project. Then the fault coverage of the LBIST execution and its test application time is evaluated.

1.2 Specifications and components

The main specifications and components of the design are:

- Number of scan chains of the scan-inserted RISCV: 30
- Number of longest scan chain : 102
- A 286-bit LFSR
- A 286-bit phase shifter
- A 28-bit MISR
- A FSM-implemented LBIST controller
- A 256-bit multiplexer of the input of the PIs and a 109-bit multiplexer for the PIs.

1.3 Design Flow

The following steps were followed for the development of this project:

- Implementing LBIST controller, MISR, LFSR and other components
- Implementing a Top Module and a wrapper
- Developing a test bench that invokes the LBIST and then executing a given firmware
- Obtaining patterns generated by the LFSR and the phase shifter, testing the RISCV using Tmax to gain a reasonable coverage
- Implementing upgrades to the design in order to improve the fault coverage
- Synthesising the design and comparing the area overhead with the original RISCV

CHAPTER 2

LBIST implementation and improvement applications

2.1 Implementing the LBIST

2.1.1 Scan-chain insertion

First step that had to be done was to insert a reasonable amount of scan-chains to the processor. For this project, 30 scan chains were decided to be inserted to the processor which would affect the number of shift-in and capture cycles in the testing stage. Analyzing the .spf file of the newly scan-inserted processor, it was understood that the number of cells in the longest scan-chain is 102 deciding the number of cycles needed for the shift-in and capture process.

2.1.2 LFSR design

The LFSR was implemented according to the fact that it is supposed to feed both PIs and the scan-chains, so since the processor had 261 bits of PIs and 30 scan-chains were also inserted, the LFSR would have 291 bits of input, but since 5 of the PIs were critical bits including [clk,rst-ni,clock-en,test-en,fregfile-disable] and could not be fed by the LFSR, the LFSR is implemented by 256 bits and the 5 critical bits were directly connected to the wrapper.

2.1.3 MISR design

In this design, MISR is set to only capture the values of the scan chains, so it would be 30-bit MISR, but since the Test Point Insertion was implemented later on, it was modified to a 28-bit MISR which could capture the outputs of the scan-chains and produce a signature to be compared with the golden value at the end of the testing cycle.

2.1.4 Phase shifter

The phase shifter is implemented in order to increase the fault coverage of the LBIST. This is a 286-bit design which takes the values produced by the LFSR as the input and by performing XORs among the bits, produces fully randomized values in its output.

2.1.5 LBIST Controller

The controller is designed as a FSM with 5 stages including [reset, initial-shift-in, PI-PO, shift-in-capture, comparison] which is activated upon receiving the value 1 as its Normal-Test signal.
The reset state is the idle state in which all the components are off and the counters are all set to value 0.
The initial-shift-in state invokes the LFSR and enable the test-mode and scan-enable in order to start the initial feeding of the scan chains and this stage is set to loop for 102 cycles which is equal to the length of the longest scan chain.
The PI-PO state is the state where for one cycle, the scan-en is disabled , the multiplexer in the input of the RISC is set to 0 in order to feed the PIs and the MISR to capture the POs. This stage takes only one cycle since it is a combinational part of the processor.
The shift-in-capture state re-enables the scan-en and starts capturing the values from the output of the scan-chains in parallel with shifting-in again the values from the LFSR to the scan-chains. This state is repeated for the number of the intended patterns.
The comparison state compares the final signature produced by the MISR with the golden value which is obtained simulating the fault-free processor. In case of match, the GO-NOGO signal is raised to 1 and otherwise it is set to 0 which indicates occurrence of a fault in the circuit.

2.1.6 Top Module

A top module is implemented in order to port-map the components with pins connecting the RISCV to the wrapper and the memory and a Normal-Test and a Go-NoGo pin to be connected to the controller and the test bench.

2.1.7 Wrapper

A wrapper was implemented in order to connect the top module to the memory and also the test bench. The wrapper already provided in the LAB sessions was used with further modifications done in order to include the top level in it.

2.1.8 Multiplexers

The first multiplexer implemented in the PIs is responsible to feed the PIs with the values from wrapper in the normal mode or the values produced by the LFSR in the test mode.
The memory multiplexer is designed to connect the POs to the wrapper in the normal mode and leave the POs which derive the memory floating in the test mode.

2.2 Initial Simulation and Fault Testing

Our testing procedure is based on the Division section of the processor in order to avoid long simulation periods.
Starting with the initial design of the LBIST which did not include the phase shifter a very low coverage of 11 % was obtained. As suggested, random pattern generation method was used and using 30 patterns which was the number of patterns intended in our design, the simulation resulted in 46 % of coverage.

2.3 Improvement applications

2.3.1 Phase shifter

The first step taken towards the improvement of the coverage was to implement a phase shifter. This phase shifter applies a fully random procedure of XORs among the bits which are fed to the scan chains and another loop implemented to further randomize the values fed to the PIs during the testing phase. It is understood that also fully randomizing the values fed to the PIs with the same procedure done for the scan chains could improve the fault coverage even further, but due to the lack of time and high number of the bits, it was not implemented.

2.3.2 Test Point Insertion

The test point insertion was done to improve the coverage.
After running the provided scripts, the new circuit was analyzed and it was figured that the pins [test-mode,test-so13,test-so15] were removed and a new pin called [test-mode-tp] was added which was supposed to stay high during the testing phase. This pin was then connected to a newly inserted pin in the controller and was enabled in the initial-shift-in state until the end of the testing procedure. The top level entity was also modified according to the modifications done in the test point insertion step.

2.3.3 Seeding

In order to obtain higher coverage, some different seeds were implemented in the LFSR and it was observed that the seed can have a considerable effect on the fault coverage.
Re-seeding could be implemented in the LFSR which would help improve the fault coverage, but due to the high number of bits in the LFSR and lack of time, it was not implemented.

2.3.4 Further improvements

At last, the approach of using one LFSR for both scan chains and PIs and high number of PIs resulted in some restrictions due to the lack of time. Using another LFSR for only the scan-chains would make re-seeding possible.

2.4 Final Fault Simulation

After the improvements were implemented mentioned above and considering 30 patterns, a fault coverage of 69% was obtained with a CPU time of 35.74.
Using random patterns generation and 100,000 patterns for all the faults, about 84% of fault coverage in CPU time of 113.78 was obtained and about 86% was obtained in the faults inserted in the Division section.
It is believed that with the further improvements of the phase shifter, better choice of the initial seed and re-seeding, a better coverage is obtainable with taking into account the fact that re-seeding would require an internal memory for the LBIST which would increase the area overhead. Also increasing the number of patterns can improve the fault coverage while simulating all the faults in the processor. Comparing the results obtained by our LBIST and the random pattern generation method, it is believed that with higher number of patterns and the improvements mentioned above, the LBIST is capable of achieving a fault coverage close to the results in the random patterns generation method.

CHAPTER 3

Synthesis

3.1 Original RISCV synthesis

The results regarding the area of the original RISCV:

Total Cell Area	63552.986668

Table 3.1: RI5CY area report

3.2 LBIST Synthesis

The result regarding the area of the synthesized LBIST with the scan inserted and test point inserted RISCV is as following:

Total Cell Area	66141.166396

Table 3.2: LBIST area report

3.3 Conclusion

With the obtained results from the synthesis section, it is observed that the LBIST has the overhead of 2,588.179728 in the total cell area.