Politecnico di Torino

# Testing and Fault Tolerance assignment

HW ASSIGNMENT

Luca Dalmasso (s281316)
Alessandro Landra (s284939)

Course: Embedded Systems
Date: 15/01/2022

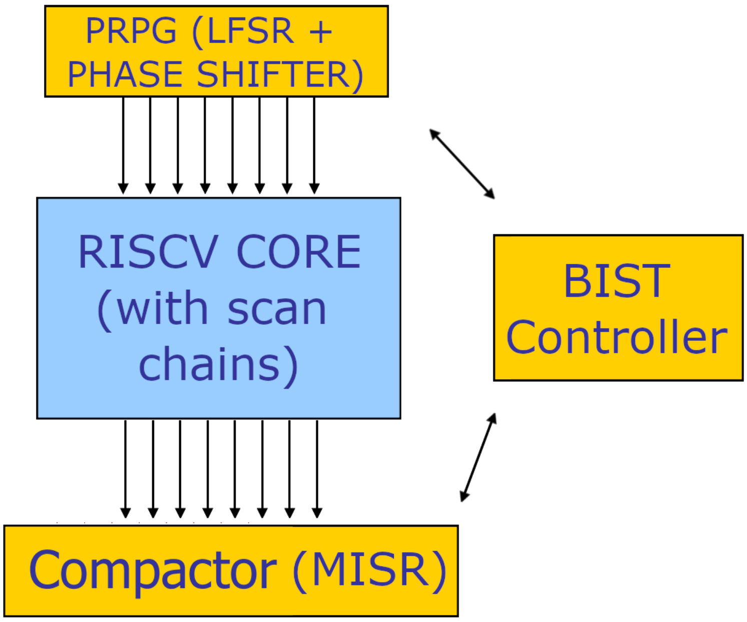
## Goal of the assignment

The purpose of this project is to design a Logic-BIST for testing a RISCV core trough the adoption of some testing techniques presented during the course and usage of industrial tools.
The constraints in terms of test coverage, are at least 70% of coverage for the Stuck-At fault model

## Proposed Solution

The solution proposed is based on the Test-per-scan technique, which is a very popular architecture (Figure1) used to test big sequential circuits with random patterns.

Figure1: Block Diagram of the LBIST



The LBIST designed performs a fixed number of test cycles, in particular our target was 70% of stuck-at coverage and after some tests we found a reasonable number of test patterns necessary to archive that coverage: 40000 patterns.

Each test cycle is a procedure that is organized in 2 steps:

1. Scan chain upload: scan chains are loaded with random patterns, during this step the circuit works in scan mode.
   Duration: #ScanChains * Tclk
2. Capture: this is a clock period where the circuit works in normal mode so that the circuit reaction is captured in the scan chains.
   Duration: Tclk

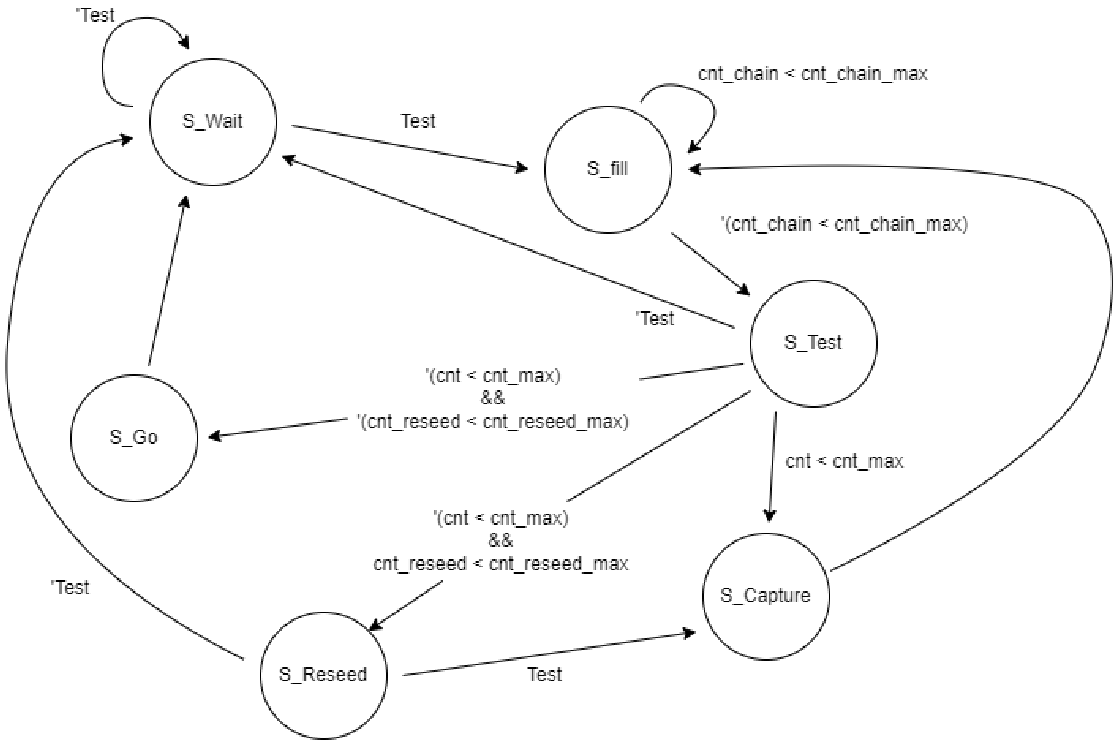
2a. Output compaction: Scan outputs are compacted in a MISR.

During the entire test, the scan chains are fed with random patterns generated by a PRPG that is periodically “reset” with a new seed, this is done in order to produce new random vectors with a new starting point and thus, have a major probability to increase the coverage.

During the entire test, the scan chains are connected to a MISR that is always enabled. This means that all the time the MISR will use the outputs for computing the signature, not only after the capture phases.
We chose to use 64 scan chains, so the MISR is on 64 bit, this guarantees a very low aliasing probability ($2^{-64}$).

The BIST controller (Figure2) is the unit that controls the test logic during the test phase, it is a very simple 6 state FSM with 3 counters.

Figure2: BIST Controller FSM



The test procedure starts when the Test signal is raised (Test = 1 → TEST_MODE, Test = 0 → NORMAL_MODE) the FSM will move to S_fill state.
The FSM will alternate Fill (S_fill), Capture (S_capture) and Reseed (S_reseed) phases discussed before.

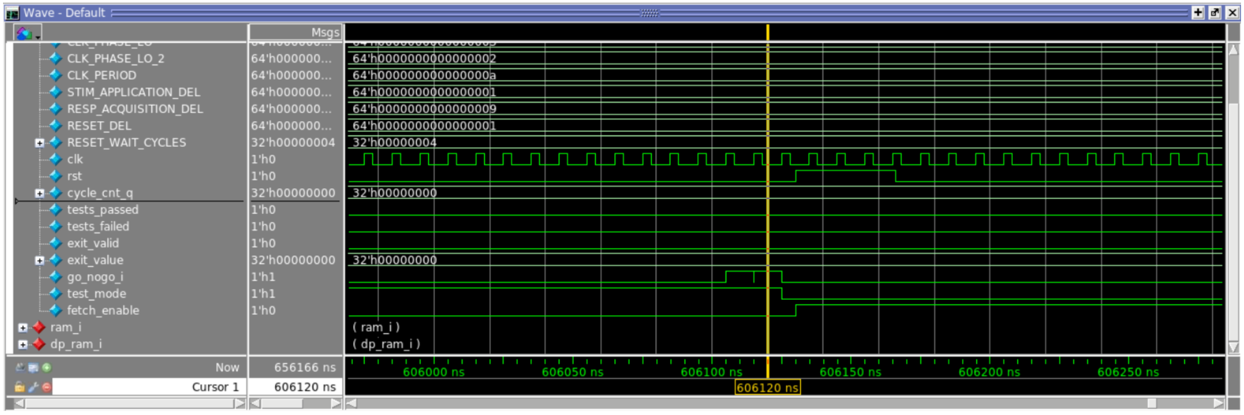
Fill phase will last for 48 clk cycles (48 is the maximum length of a scan chain) so that all scan cells are correctly fed in parallel with random patterns, after this the processor is ready to do a capture, so the Capture state will be reached (only 1 clk cycle is used here).

The flow, described by the state chart (Figure2), shows Fill and Capture states that are interleaved by an intermediate state (S_test) that every 101 capture phases, perform a reseeding on the PRPG by moving to the S_reseed state.

After a certain number of reseeds done, the test phase will move to the S_go state where the signature produced by the MISR is compared with the golden one.
This is the last step of the test phase, the GO-NOGO signal will be raised to one for two clock cycles if  the circuit is good, only for 1 clock cycle if instead the circuit is faulty.
In all cases when go_nogo is raised by the LBIST means that the test procedure ended (All patterns have been produced and compacted in the MISR).

In the following picture (Figure3) is possible to see the end of a test procedure, the go_nogo is raised to 1 for two clock cycles, the test ended good, the circuit is not faulty, so the firmware can start his execution (fetch_enable signal is raised).

Figure3: End of test procedure, firmware starts the execution



Wave - Default

| | Msgs | |
|---|---|---|
| CLK_PHASE_LO_2 | 64'h000000... | 64'h0000000000000002 |
| CLK_PERIOD | 64'h000000... | 64'h000000000000000a |
| STIM_APPLICATION_DEL | 64'h000000... | 64'h0000000000000001 |
| RESP_ACQUISITION_DEL | 64'h000000... | 64'h0000000000000009 |
| RESET_DEL | 64'h000000... | 64'h0000000000000001 |
| RESET_WAIT_CYCLES | 32'h00000004 | 32'h00000004 |
| clk | 1'h0 | |
| rst | 1'h0 | |
| cycle_cnt_q | 32'h00000000 | 32'h00000000 |
| tests_passed | 1'h0 | |
| tests_failed | 1'h0 | |
| exit_valid | 1'h0 | |
| exit_value | 32'h00000000 | 32'h00000000 |
| go_nogo_i | 1'h1 | |
| test_mode | 1'h1 | |
| fetch_enable | 1'h0 | |
| ram_i | | ( ram_i ) |
| dp_ram_i | | ( dp_ram_i ) |
| Now | 656166 ns | |
| Cursor 1 | 606120 ns | 606120 ns |

## TEST DURATION

Considering the following configurations (Good for 70+% coverage):

- Fill: 48 Tclk
- Capture: 1 Tclk
- Test: 1 Tclk
- Repeat Fill Capture and Test for 101 times with the same seed: (48 + 1 + 1) * 101 = 5050 Tclk → 5050 random patterns for each test cycle.
- Reseed
- Repeat for other 11 times (12 seeds in total)

Total overhead: 5050 * 12 * Tclk → Tclk=10 ns → 5050 * 12 * 10 = 606 us.

Roughly speaking, the test phase needs at least 0.6 ms during which 60 K patterns are generated and compacted.

## PERFORMED OPTIMIZATIONS

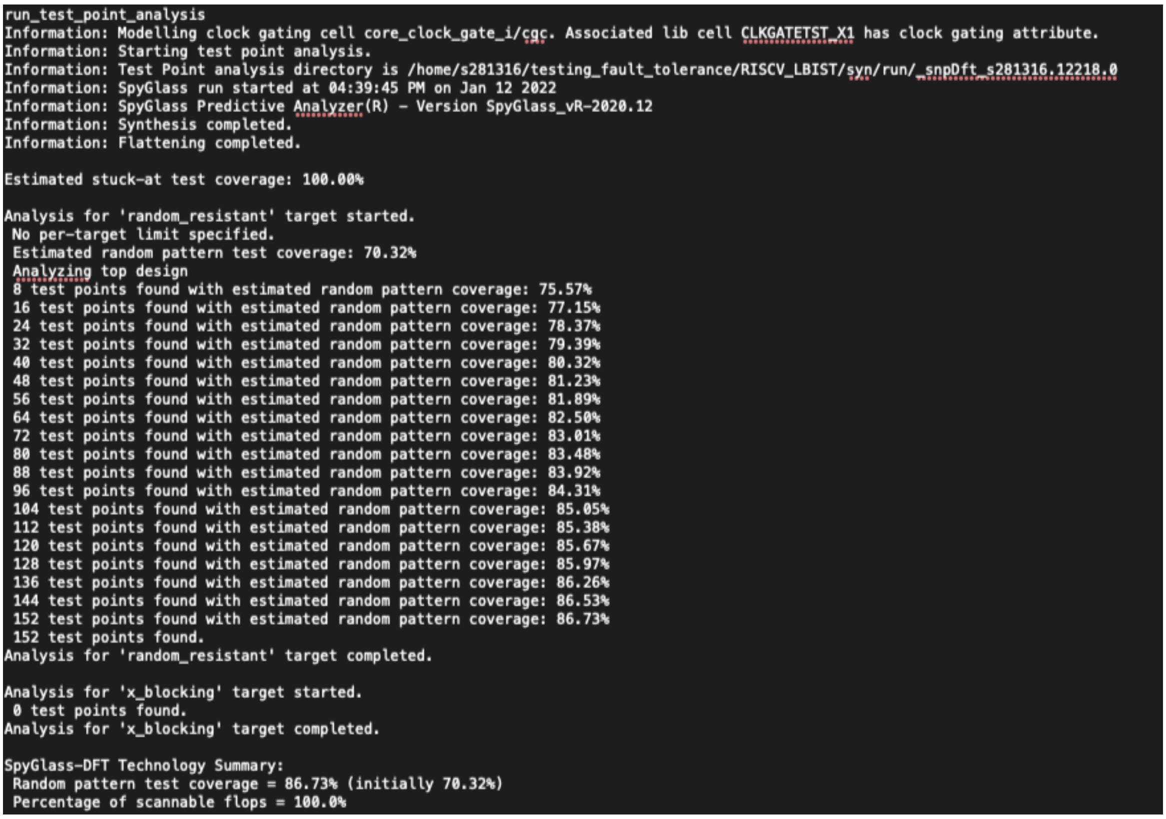
1. Scan chains number
   As introduced before, we decided to use 64 scan chains, this choice is a trade off in terms of Accuracy, Cost and Test duration.
   This number was selected for 3 reasons:

   a. Low MISR aliasing probability.
   b. Higher number would lead to an higher hardware cost (Bigger PRPG and MISR) without an effective advantage in terms of aliasing (2^(-64) is already a negligible probability).
   c. Lower number would lead to a lower hardware cost but the price to pay is a higher aliasing probability and also a longer test time (less scan chains = longer scan cells into a single chain → more Tclk used by the Fill phase)

2. Scan In / Out sharing mode
   We decided to avoid inserting 64 scan-in and 64 scan-out by sharing them with some PIs and POs, so that the only overhead (in terms of pins) is due to the Scan Enable pin.

3. Test point insertion
   Due to the fact that the UnitUnderTest (RISCV core) is a random resistant hardware, it is very difficult to archive good coverage only with random patterns. The estimated coverage with random patterns was only 70.32% (Tmax report), this means that in order to have an higher coverage a huge amount of patterns may be required.
   We decided to take advantage of a very powerful tool offered by Tmax: SpyGlass, that is able to automatically insert Test Points in order to reduce the random resistant faults.
   After the insertion of test points, Tmax analysis showed that the random patterns coverage increased by more than 16%, so theoretically with more patterns the coverage could almost reach 90%.

Figure4: Tmax SpyGlass test point analysis

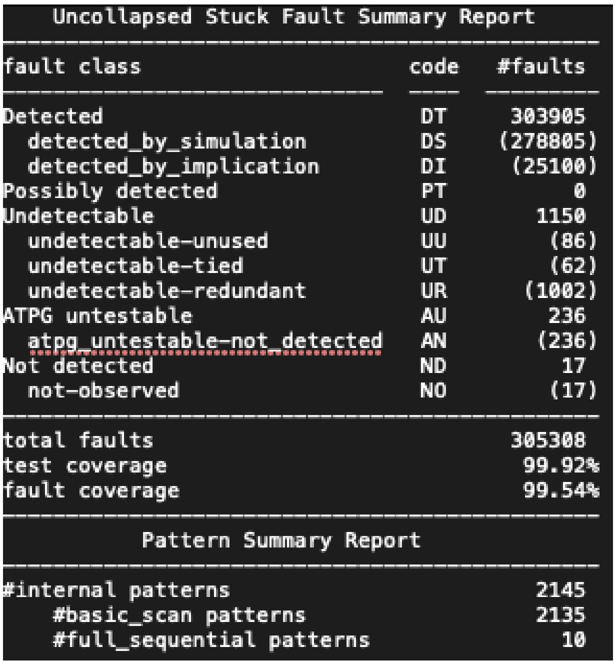


```
run_test_point_analysis
Information: Modelling clock gating cell core_clock_gate_i/cgc. Associated lib cell CLKGATETST_X1 has clock gating attribute.
Information: Starting test point analysis.
Information: Test Point analysis directory is /home/s281316/testing_fault_tolerance/RISCV_LBIST/syn/run/_snpDft_s281316.12218.0
Information: SpyGlass run started at 04:39:45 PM on Jan 12 2022
Information: SpyGlass Predictive Analyzer(R) - Version SpyGlass_vR-2020.12
Information: Synthesis completed.
Information: Flattening completed.

Estimated stuck-at test coverage: 100.00%

Analysis for 'random_resistant' target started.
 No per-target limit specified.
 Estimated random pattern test coverage: 70.32%
 Analyzing top design
 8 test points found with estimated random pattern coverage: 75.57%
 16 test points found with estimated random pattern coverage: 77.15%
 24 test points found with estimated random pattern coverage: 78.37%
 32 test points found with estimated random pattern coverage: 79.39%
 40 test points found with estimated random pattern coverage: 80.32%
 48 test points found with estimated random pattern coverage: 81.23%
 56 test points found with estimated random pattern coverage: 81.89%
 64 test points found with estimated random pattern coverage: 82.50%
 72 test points found with estimated random pattern coverage: 83.01%
 80 test points found with estimated random pattern coverage: 83.48%
 88 test points found with estimated random pattern coverage: 83.92%
 96 test points found with estimated random pattern coverage: 84.31%
 104 test points found with estimated random pattern coverage: 85.05%
 112 test points found with estimated random pattern coverage: 85.38%
 120 test points found with estimated random pattern coverage: 85.67%
 128 test points found with estimated random pattern coverage: 85.97%
 136 test points found with estimated random pattern coverage: 86.26%
 144 test points found with estimated random pattern coverage: 86.53%
 152 test points found with estimated random pattern coverage: 86.73%
 152 test points found.
Analysis for 'random_resistant' target completed.

Analysis for 'x_blocking' target started.
 0 test points found.
Analysis for 'x_blocking' target completed.

SpyGlass-DFT Technology Summary:
 Random pattern test coverage = 86.73% (initially 70.32%)
 Percentage of scannable flops = 100.0%
```

## Results

- ATPG full sequential results:
  Stuck-at-fault summary from the ATPG demonstrate a test coverage of 99.92% out of 300000+ stuck at faults, and a fault coverage of 99.52% reached with 2145 patterns

Figure5: ATPG results

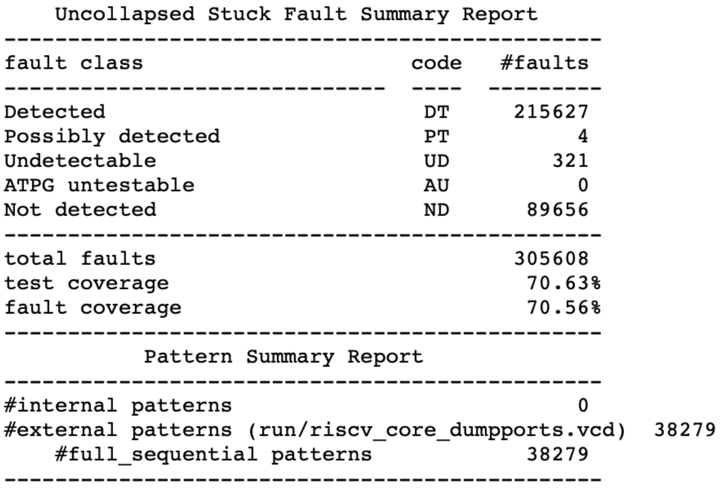


```
    Uncollapsed Stuck Fault Summary Report
-----------------------------------------------
fault class                     code   #faults
------------------------------  ----  ---------
Detected                         DT     303905
  detected_by_simulation         DS    (278805)
  detected_by_implication        DI     (25100)
Possibly detected                PT          0
Undetectable                     UD       1150
  undetectable-unused            UU       (86)
  undetectable-tied              UT       (62)
  undetectable-redundant         UR     (1002)
ATPG untestable                  AU        236
  atpg_untestable-not_detected   AN      (236)
Not detected                     ND         17
  not-observed                   NO       (17)
-----------------------------------------------
total faults                            305308
test coverage                           99.92%
fault coverage                          99.54%
-----------------------------------------------
           Pattern Summary Report
-----------------------------------------------
#internal patterns                        2145
    #basic_scan patterns                  2135
    #full_sequential patterns               10
```

- AREA Overhead with DFT:
  The total area overhead of the Core+DFT+LBIST with respect to the standalone core is: 3.43 %

- Test coverage obtained with 40000 patterns (Figure6) proofs that 70% of coverage can be reached, further fault simulation on all the 60000 patterns generated have not been tried due to huge execution time (more than 3 days). As said before, with test points is possible to reach 85, 86% of coverage but the number of patterns to generate would be too high to be fault_simulated in a reasonable time with Tmax.

Figure6: Fault simulation results with 40000 out of 60000 patterns (Tmax report)



```
    Uncollapsed Stuck Fault Summary Report
-----------------------------------------------
fault class                     code   #faults
------------------------------  ----  ---------
Detected                         DT     215627
Possibly detected                PT          4
Undetectable                     UD        321
ATPG untestable                  AU          0
Not detected                     ND      89656
-----------------------------------------------
total faults                            305608
test coverage                           70.63%
fault coverage                          70.56%
-----------------------------------------------
           Pattern Summary Report
-----------------------------------------------
#internal patterns                           0
#external patterns (run/riscv_core_dumpports.vcd)  38279
    #full_sequential patterns            38279
-----------------------------------------------
```