Fundamentals of Logic Systems
Design For Test

Dr. Paul D. Franzon

Outline
1. Scan-based testing
2. Exhaustive testing, Memory testing, and BIST
3. DFT in Synopsys Environment

References
2. M. Smith, Chapter 14.
Scan-Based Testing

Synchronous Logic Under Normal Mode of Operation:

Replace Register Cells with Scan Cell:
Scan-Based Testing...

Test Mode of Operation

---

Dr. Paul D. Franzon, www.ece.ncsu.edu/erl/faculty/paulf.html
Objective of Scan-Based Test

- To maximize the number of potential faults that can be detected.

A fault model must be assumed.
The most common fault model is stuck-at0/1 where the output of a gate is stuck-at-0 (S-A-0) or 1 (S-A-1) independent of its inputs.

For example,

```
 a  b  c
  |  |  |
  --|--|--
    |  |
    |  |
```

What are the test vectors that will detect S-A-0 and S-A-1?

Testing folklore tells us that we need 99-100% S-A-0/1 fault coverage. Why?
Scan-Based Testing...

Test Escape

It can be shown that the percentage of parts that are still faulty but are not detected as being faulty, is given as \(^1\):

\[
TE = 1 - Y^{(1 - T)}
\]

where \(TE\) is the test escape, \(T\) is fault coverage, \(Y\) is the yield.

For example, for \(Y = 50\%\) and \(T = 95\%\), what is the test escape?

Typically, using the verification vectors (from your Verilog test fixture) for test gives a 80% coverage. What is the test escape then?

Typically, fault coverages above 99% give acceptably low test escapes.

---

Scan-Based Testing...

Redundant Logic

Redundant Logic is impossible to test. Example:

Other Fault Models

• Stuck-At-Open model is a good physics-based model for CMOS circuits (S-A-open means a transistor is an open circuit).

• Delay fault capture delay errors.

Gate-level synthesis tools generate test vectors to detect S-A-0/1 only.
Exhaustive Testing

- Occasionally, exhaustive testing provides a good test if the complete set of all possible inputs can be obtained and this vector set is smaller than the 99%+ fault coverage vector set.

- Sometimes functional units can be isolated with scan registers and exhaustively tested.
Memory Testing

• Memories are exhaustively tested.

• Memories are tested for opens, shorts at individual locations and shorts between neighboring locations by marching test patterns through them.

• Memories are isolated by scan registers for testing purposes.
**Built-In Self Test (BIST)**

- In the most common BIST approach, the scan cells are modified to generate psuedorandom test vectors at the input to a logic block, and then to collect a signature at the output (using a linear feedback shift register).

- BIST takes more silicon area and longer time, but saves on the cost of generating and storing test vectors.
DFT in the Synopsys Environment

Sample Synopsys script with a concentration on DFT. (Please see the tutorial [available in Iview] and manual for more details.)

```
read -f Verilog fsm.v
link_library = target_library = lsi_10k.db
create_clock clock81 -period 12.3

/* must expand design and click on clock81 first */
set_input_delay -clock clock81 -max -rise 2 “RW”
set_test_methodology full_scan

/* menu: attributes -> optimize directives -> design */
set_scan_style multiplexed_flip_flop

/* compile including scan */
compile map_effort low
```
DFT in the Synopsys Environment...

/* check for testability analysis -- look at result */
insert_test
cHECK_test

/* create test patterns to check for ATPG conflicts (additional option in */
/* create pattern menu), also checks fault coverage */
create_test_patterns -sample 5

/* full test pattern creation and scan insertion are done complete chip at */
/* the end but try them if you want */
insert_test -scan_chains 1
create_test_patterns -output fsm.vdb \
-compaction_effort low \
-check_contention_float true -backtrack_effort low \
-random_pattern_failure_limit 64 -sample 100
Summary

• Have to design the chip to make the following tasks easy and quick:
  • First Silicon debug
  • Manufacturing test
    • Manufacturing test can be up to 40% of the chip cost

• Solutions available include:

  • Insertion of scan chains
    • Replacing every register with a scannable register gives 100% coverage (full scan)
    • Close to full coverage can be achieved with judicious selection of registers to be scanned
    • Scan chains are useful in debug too
    • Supported by Synopsys

  • Built In Self Test can be cheap in many cases
    • Consider impact on debugging though