Understanding DFT & VLSI Testing

Mohammed Rafi, Ann Zi Lau, Gokulkrishnan Harikrishnan, Richard VonTersch, Michael Fuchs, Pranay Jaggi, Roan Numa, Praneel Bhandari
Varun Gannavarapu

Purdue University

Abstract

VLSI has a strong need for increased reliability and testability. As the process for semiconductor technology has rapidly reduced from 180 nm to 3 nm in a span of 21 years, there are difficulties in overall testability as there are very low controllability and observability capabilities from the top-level.

The goal of this project is to demonstrate the application of DFT (Design for Testing) to digital logic circuits and how it can be used to improve the reliability and testability of digital circuit designs.

For our future work, we will be implementing DFT for the AFTx07 (the next chip tapeout by the SoCET Team) and enhancing the overall efficiency, reliability, and testability of the tapeout process.

Background

What is VLSI (Very Large-Scale Integration) ?

- The process of designing integrated circuits (ICs) by integrating thousands or more transistors on a single chip.
- Mainly used to design microprocessors and memory chips for consumer devices.

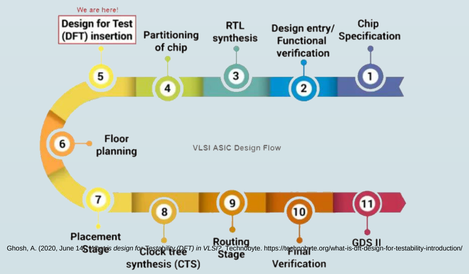
What is DFT (Design for Testability)?

- A design technique used to increase the testability of all logic in a chip
- Allows for testing a chip to be cost-effective with the addition of test circuitry.
- DFT increases the two components of testability which are observability and controllability.
  - Controllability is the difficulty of controlling a logic signal to 0 or 1
  - Observability is the difficulty of checking the value of a signal at a certain point

What is ATPG?

- ATPG (Automatic Test Pattern Generation) is a tool that allows the user to produce and send a test pattern to detect faults within a digital logic circuit
- Effectiveness is determined by number of faults detected from number of test patterns sent

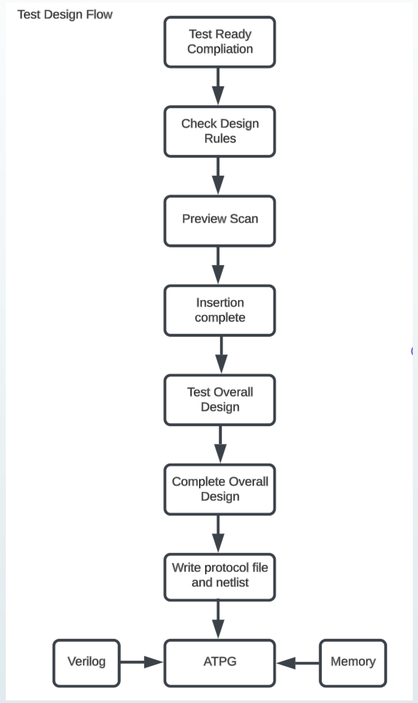
This project will focus on applying the DFT techniques in VSLI design to demonstrate how DFT can be utilized for testability and reliability in digital circuit designs.



Ghosh, A. (2020, June 14). *Design for testability (DFT) in VLSI*. Technobyte. https://technobyte.org/what-is-dft-design-for-testability-introduction/

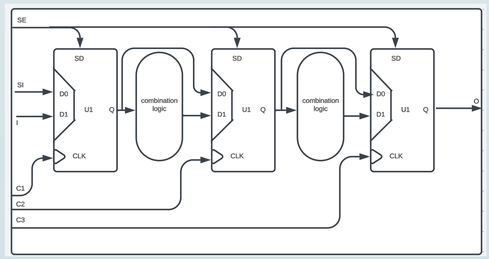
Methodology

DFT Flow



Scan Insertion:

- Design sequence of 0s and 1s for input
- Cycle through registers, process for one clock cycle
- Analyze for correctness
- Repeat process for numerous patterns to find faults



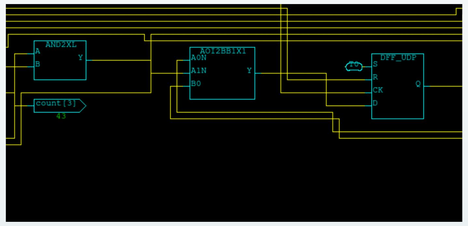
This team aims to implement DFT through Scan Testing which is done by inserting Muxed-D flip-flops. These Muxed-D flip-flops allows external inputs to be passed into the logic circuit at the points they are installed in as well as reading out values for testing.

Results

Scan Insertion:

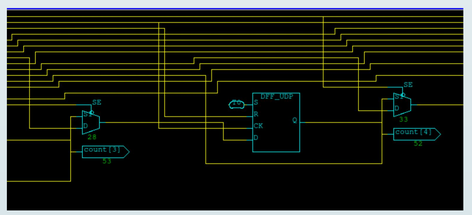
Original Counter Design

```
module counter(clk, rst, count);
  input clk, rst;
  output [7:0] count;
  wire clk, rst;
  wire [7:0] count;
  wire n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7;
  wire n_8, n_9, n_10, n_11, n_12, n_13;
  DFFRHQX1 \count_reg[7] (.RN (rst), .CK (clk), .D (n_13), .Q
       (count[7]));
  DFFRHQX1 \count_reg[6] (.RN (rst), .CK (clk), .D (n_12), .Q
       (count[6]));
```



Muxed-D Scan inserted Design

```
module counter(clk, rst, count, SE, scan_in, scan_out);
  input clk, rst, SE, scan_in;
  output [7:0] count;
  output scan_out;
  wire clk, rst, SE, scan_in;
  wire [7:0] count;
  wire scan_out;
  wire \count[0]_45 , n_0, n_1, n_3, n_4, n_5, n_6, n_7;
  wire n_8, n_9, n_10, n_11, n_12, n_13, n_14;
  assign scan_out = count[7];
  SDFFRHQX1 \count_reg[7] (.RN (rst), .CK (clk), .D (n_14), .SI
       (count[6]), .SE (SE), .Q (count[7]));
  SDFFRHQX1 \count_reg[6] (.RN (rst), .CK (clk), .D (n_13), .SI
       (count[5]), .SE (SE), .Q (count[6]));
```

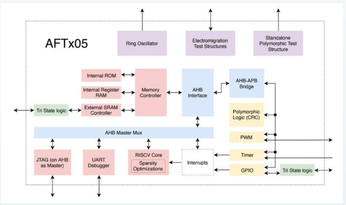


ATPG Test Coverage:

```
TEST-T> report_summaries
      Uncollapsed Stuck Fault Summary Report
-----------------------------------------------------
fault class                     code   #faults
------------------------------  ----  ---------
Detected                         DT         192
Possibly detected                PT           0
Undetectable                     UD           8
ATPG untestable                  AU           0
Not detected                     ND           0
-----------------------------------------------------
total faults                                 200
test coverage                             100.00%
-----------------------------------------------------
            Pattern Summary Report
-----------------------------------------------------
#internal patterns                            15
    #basic_scan patterns                      15
-----------------------------------------------------
```

Conclusion – Why DFT?

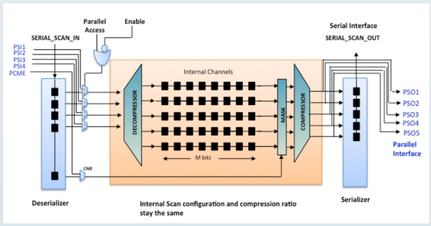
- SoCET does not currently have a way to test hardware
- Post-silicon testing is extremely important to catch errors
  - Errors in manufacturing can lead to non-working parts
  - Errors in design may only show up when manufactured
- DFT is important for final products, designs must be testable for consumers
  - Without testability your design is near-useless as you cannot guarantee end product is working as expected
- Testability provides the ability to create known-good parts, allowing for testing of other designs
- DFT allows you to check where it is failing and correct mistakes



Future Work

For Test Design Team

- FPGA_dft port mappings
- Implementing DFT styles and generating test vectors on the AFTx07 (and its submodules)
- Map the I/O ports of the FPGA to AFTx07_testvectors and control bits.
- Configure FPGA by loading the testbench design.
- Use the FPGA to apply test vectors to the chip and monitor the outputs.



Edn. (2014, October 7). *Test cost challenges in LPCT (low pin count test) designs*. EDN.

Acknowledgements

Adam Cron (Accomplished Design for Test Expert)
Dr. Mark Johnson
Abinands Ramshanker
Fan Jing Hoon
Used Cadence Genus and Synopsys Testmax

SoCET VIP Vertically Integrated Projects SYNOPSYS® cadence®