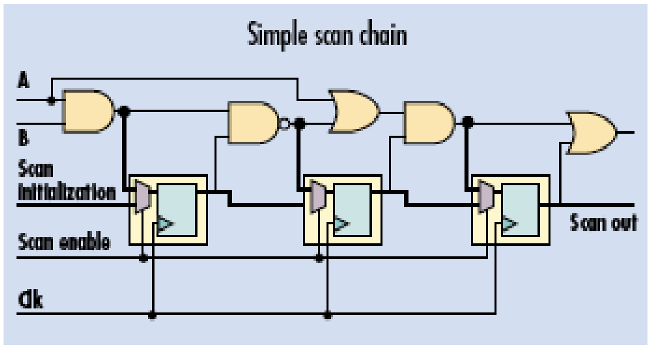
DFT SCAN CELLS NETWORK DESIGN

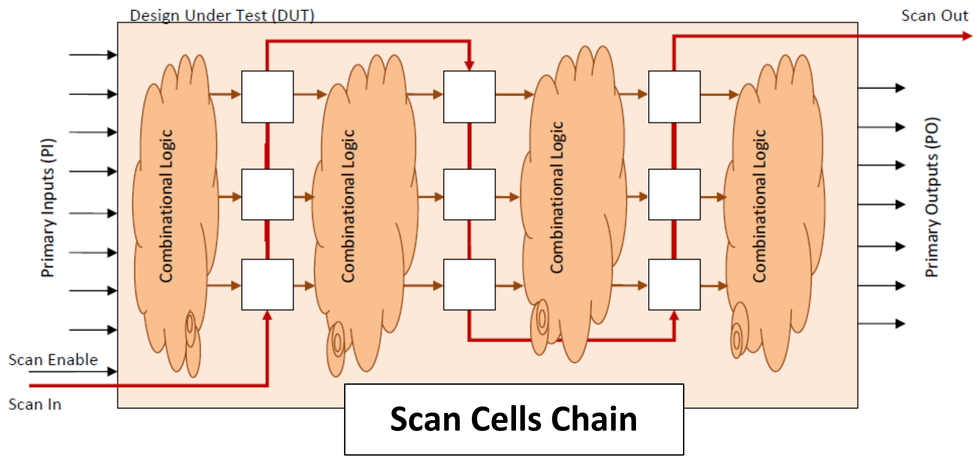
College Mentor: Samy Zafrany

Due to the growing complexity of electronic designs, **Design For Testability** (DFT) field has become an essential part of almost any highly complex VLSI design for the past 10 years. It refers to a set of techniques for generating test applications for verifying chip correctness and functionality.

The major DFT technique consists of placing and connecting a large number of special test devices call "SCAN CELLS". These cells are placed in proximity to almost all important chip junctions that need to be tested and verified. Almost every chip primary input or output can be sampled and verified at varying clock phases and cycles by properly programming the SCAN CELLS NETWORK.

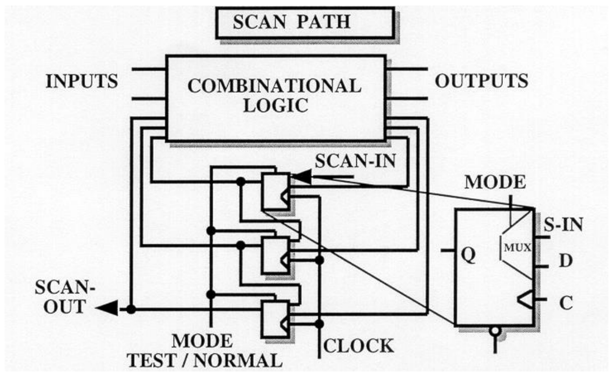


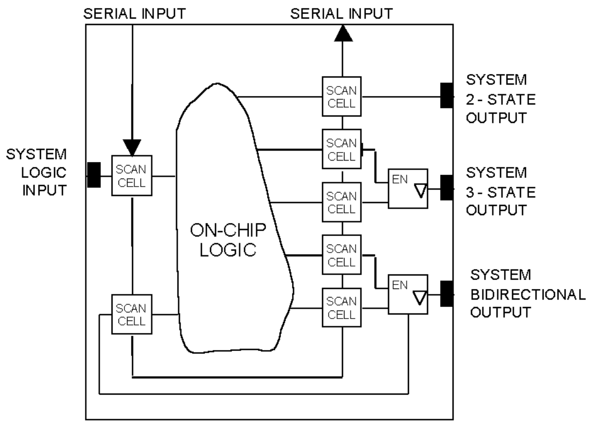
However due to the large distribution of scan cells all over the chip (roughly 5% of its area!), one of the major concerns is how to minimize its total routing length? As chip routing resources are scarce, wasting too much routing for DFT can come on the expense of routing for the real chip functionality.



The aim of this project is to introduce the student to a small scale computerized model of the scan cells network design problem, and developing a heuristic algorithm for minimizing scan cells network size. We will focus on a simulated annealing approach in a level which is appropriate for college student.

Minimizing total wire length is a well-known NP-hard problem and no efficient analytical methods are yet known for solving such problem even for a 500 cells network, even for the simplest type of network (Hamiltonian path that traverses all scan cells).





The students will be exposed to the rich domain of DFT (Design For Testing) in VLSI.

Outline for Solution Stages

The following list is an estimated path on how to solve the problem and proceed to a successful project completion

1. The first thing is to get acquainted with the basic floor planning concepts, techniques, algorithms, and software tools.
2. We will be using Python and C++ programming languages and the Tkinter graphical user interface for visualizing VLSI components on top of a TK canvas. The student should invest the needed time to gain intuition, learn and get enough practice in these coding environments before wetting his hands with the real problems
3. The student should start with writing simple programs this includes mastering the VLSI GUI drawing tools (such as a basic VLSI canvas), communicating and manipulating VLSI canvas objects.
4. As a start, the student should implement a simple inefficient algorithm for generating scan cells to match a given floorplan design. The student should be responsible for generating inputs and checkers and verifiers for checking correctness, distribution, and floorplan coverage.
5. Next, the student should start implementing basic greedy algorithms for generating scan cells path (or network). This includes generation of appropriate sets of inputs for the algorithms, and an adequate set of checkers, benchmarks, and regression tests for verifying correctness and legalization. Regression tests should be general enough to be used later for the heuristic algorithms.
6. The main challenge in this project is to combine a heuristic approach such as simulated annealing (or genetic algorithm) with combinatorial method for generating good enough scan cell networks. As the problem complexity in NP-hard, and as all known industrial algorithms are based on heuristic methods, and there's still a lot of room for innovation and improvement. The student will have to come up with one or more ideas and heuristics on how to tackle this problem and then implement it on software.

7. For that purpose, the student may have to get acquainted with several advanced heurist algorithms such as probabilistic methods, simulated annealing, evolutionary computing (genetic algorithms), and may adopt one of them for his problem - on a smaller scale if they turn to be too powerful for our problem, or the student can come up with a new more appropriate heuristics for the problem. This is yet very open to many directions and we will be more informative once we approach this stage.
8. The last thing is to run our algorithm on real life cases (know internet industrial test cases) and see how we're doing in several benchmarks such as run time, memory consumption, floor quality, etc. If feasible, we may try to compare our performance against industrial algorithms if available (although most of them are proprietary and are not publically available).

The project is yet open for variations and extensions to other type of algorithms and layout quality constraints, and potential students are encouraged to come up with creative ideas, which can make this project suitable for several students (each focuses on a different algorithm/constraint set).

Development durability: One Semester (450 hours)

Resources:

1. http://people.ee.duke.edu/~krish/teaching/ECE269/how_does_scan_work.pdf
2. http://intranet.daiict.ac.in/~vlsi/dft.pdf
3. http://www.math.uwaterloo.ca/tsp/vlsi/page2.html
4. http://soc.cs.nchu.edu.tw/pllai/NCUT/VLSI%20Testing/
5. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.589&rep=rep1&type=pdf
6. http://www.csee.umbc.edu/~cpatel2/links/418/lectures/chap14_lect13_scan.pdf
7. http://www.wiley.com/legacy/wileychi/mblin/supp/student/LN16DesignforTestability.pdf
8. http://courses.ece.ubc.ca/578/
9. http://www.engr.uconn.edu/~tehrani/teaching/cad/14_floorplanning.pdf
10. http://book.huihoo.com/design-of-vlsi-systems/ch08/ch08.html
11. http://vlsicad.ucsd.edu/Publications/Journals/j46.pdf
12. http://www.or.uni-bonn.de/~vygen/files/analyt.pdf
13. http://effbot.org/tkinterbook/canvas.htm
14. http://effbot.org/tkinterbook/tkinter-index.htm
15. https://wiki.python.org/moin/TkInter
16. http://effbot.org/tkinterbook/tkinter-index.htm