

EE 709

Testing and Verification of VLSI Circuits

Project1 - Combinational Test Generation

Arun. C
Siddharth. Mohta

Dept. of Electrical Engg
IIT Bombay

May 2, 2012



Implement ATPG Algorithm

- Algorithm - D Algorithm.
- Input - Netlist in Verilog language. (ex1.v)
 - Only verilog builtin primitives - and, nand, not etc.
 - Only structural combinational logic supported.
- Output
 - Fault Pattern file. (ex1.v_fault.lst)
 - Log elaborating run time details. (output to console).
Can be captured in unix shell using tee command.
- Language - C++
- Dependencies - Boost library (To be a part of C++-11 standard)



Brief D Algorithm:

Basic TG algo has 2 parts

- Line Justification.
 - Justify the opposite value of the fault to the line from PI.
 - If its a SA-1, justify a 0 value to the fault line and vice-versa
 - May need to backtrack if needed.
- Fault Propagation.
 - Propagate the D/Dbar to one of the PO.
 - May need to Justify if needed.
 - Required in case of multiple input gates.
 - May need to backtrack.



Algorithm and Usability

- Each fault line is noted as “component-name+net-name”
 - Eg: or_gate1 or(out1, in1, in2);
fault lines are or_gate1_out1, or_gate1_in1, or_gate1_in2
 - TG done for each output and input lines separately.
- SA-1 and SA-0 patterns generated for each of these lines.
- Results are printed to the console and a file, fault.lst
- TG algorithm traverses through the data structure using maps.



Program organized into 3 parts

- Verilog Parser - Front End.
- Data Structure - Graphs, Nodes and Maps (Hash Arrays).
- ATPG algorithm.

1. Program Front End - Verilog Parser.

- Verilog Netlist parsed using regexp functions in boost library.
(class VerilogLineParser)
- Boost version - 1.49 is used. ^a

^awww.boost.org



2. Datastructure - Graph, Nodes and Maps (Hash arrays)

- Each component is stored as a class VerilogNode.
 - Stores information like, name, cv, iv etc.
- Each line is stored as another class Line. (class LineType)
 - Properties - direction, value, type etc.
- Data structure is traversed using maps. (Global STL)
 - linemap : line to node.
 - line_to_faninline_map : line to driver line.
 - line_to_fanoutline_map : line to fanoutlines. (driven lines).
 - line_to_inputline_map : o/p line to i/p lines of the component.
 - line_to_outputline_map : i/p line to o/p lines of the component.



3. ATPG Algorithm

- GenerateTP ()
 - Main function that generates test patterns. (class Atpg)
- Justify()
 - Justifies the given node with the given value. (class Atpg)
- Propagate()
 - Propagates the given value to final PO. (class Atpg)
- Supported by other functions and classes.



Testing

Tested with the following 4 basic circuits.

- Each circuit exhaustively verified manually for the patterns generated.

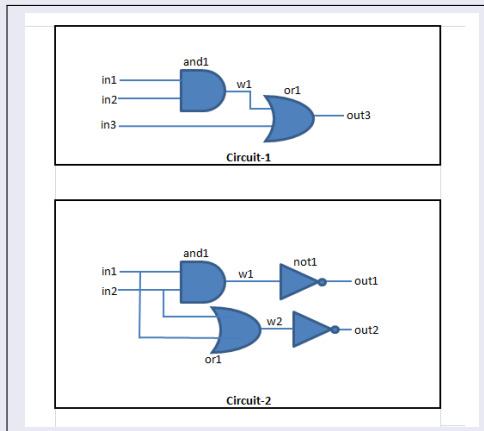


Figure: Test Circuits 1&2

Testing...

Circuits 3 & 4. (4th is reconvergent circuit)

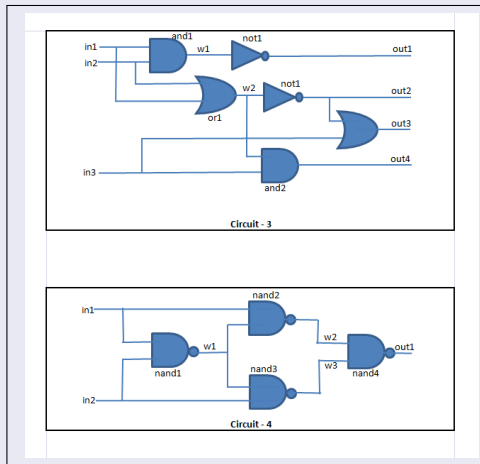


Figure: Test Circuits 3&4

Current Problems and Testing

- Apply the program to more circuits and figure out the bugs.
- Generate the test pattern for ISCAS circuits.
 - Not attempted - No golden response available.

Enhancements and other algorithms

- Code more robustly.
 - Currently lot of workarounds - Plan properly.
- Use better algorithms.



References



Michael L. Bushnell and Vishwani D. Agrawal

Essentials of Electronic Testing, for Digital, Memory and Mixed-Signal VLSI Circuits

Kluwer Academic Press



Prof. Virendra Singh. (<http://www.ee.iitb.ac.in/~viren>)

Class Notes and Lectures on Course - Testing and Verification of VLSI Circuits.

EE 709 - Dept of Elect. Engg, IIT-Bombay



THE END



Appendix-1. Main Classes Used

Verilog Netlist Parser - Front End

- VerilogLineParser

Data Structure

- FaultLine
- BuildGraph

Algorithm

- Atpg

