ECE 753: FAULT-TOLERANT COMPUTING

Kewal K.Saluja
Department of Electrical and Computer Engineering

Test Generation and Fault Simulation
Lectures Set 3

Overview

- Introduction
- Basics of testing
- Complexity reduction
  - test generation complexity
  - reduction of fault list
- Fault Simulation
- Test generation
  - combinational circuits
  - sequential circuits
  - design for testability
  - built-in self-test

Introduction

- References
- What is testing?
- How is it done?
- Why test?
- What to test for? - fault model
- Justification of the model
- Relation between testing and fault-tolerance

Introduction (contd.)

References

- [goel:81] P. Goel, An implicit enumeration algorithm to generate tests for combinational circuits, IEEE TC, March 1981
- Many books and papers in the area of testing digital circuits
- Text does not deal with testing issues
- Book by Johnson [john:89] contains a simple and necessary material for this course
- Book by Siewiorek and Swartz [siew:99] discusses some of the classical methods

Introduction (contd.)

What is testing?

- A method of determining if a given product/device is “good” or “faulty”
  - Normally a Go/NoGo approach - “detection”
  - Occasionally determine the location of fault site - “diagnosis”
- apply input - observe outputs to
  - device under test (DUT) - many other acronyms such as CUT, PUT, BUT, ...

Introduction (contd.)

What is testing?

- Typical constraints
  - no internal probing of the device
  - internal details may or may not be available - hence may have to take a black box approach to testing

How is it done?

- Tester based
  - gold unit
  - simulation
- Non-tester based
  - self-test

Introduction (contd.)

Why Test?

- Determine if a product is good or faulty
- Business/cost - cost of not testing is too high
  - rule of 10
  - quality - closely related to testing

Introduction (contd.)

What to test for? - Fault model

- Assumptions
  - digital circuit
  - gate level description available
  - apply input and observe output - no internal probing or any other measurements
  - logic testing - observe logic level
- Which model to chose?
  - Single stuck-at fault model

Introduction (contd.)

Justification of the model

- Empirical evidence - it works
- Simple and practical
  - tractable
  - easy to use - many existing hardware and software tools use this model
- It has stood the test of time

Introduction (contd.)

Relation between testing and fault-tolerance

- Follows from the definitions of reliability and availability - conditions at t = 0
- It is the basic method of “fault avoidance” for fault-tolerance

Basics of testing

- Truth table approach
- How to reduce number of tests
- Quality of tests

Basics of testing (contd.)

Truth table approach

- An example
- Limitations
  - large number of inputs
  - large number of faults
  - large number of tests
  - difficulty in handling sequential circuits

Basics of testing (contd.)

How to reduce number of tests

- An example
- Methods
  - sequential approach
    - find a test for the fault not yet detected
    - determine all faults detected by it - fault simulation
    - do not generate tests for the faults so detected
  - cover table approach
    - ability to provide an optimal solution (test set containing fewest number of tests)

Basics of testing (contd.)

Quality of tests

- Intuitive figure of merit
  - more fault a test set detects, the better the test set is
- Fault coverage
  - ratios of faults detected by a test set to the total number of possible faults in the circuit
- Methods to obtain coverage metric
  - create fault list
  - simulate circuit with and without fault and determine detected faults
  - obtain fault coverage

Complexity reduction

- Test generation complexity
  – equivalent to satisfiability problem
  – an NP complete problem for combinational circuits
  – clearly NP for sequential circuits

Complexity reduction

- Fault list reduction
  – fault equivalence of stuck-at faults
  Two faults are said to be equivalent if the circuit behavior in the presence of either of these two faults is identical
  – example to show fault equivalence
  – methods to identify fault equivalence and their application to reduce fault list

Complexity reduction (contd.)

- Modeling other faults using stuck-at fault model
  – example - stuck-on fault
  – multiple faults using a single fault model
    - using extra inputs and logic
    - using extra logic only

Simulation and fault simulation

– 2 value simulation
– 3 value simulation
– more values (5 and 9)
– symbolic simulation

Test generation

- Combinational circuit test generation
  - random pattern test generation
    - algorithm
      - generate a random input
      - simulate and determine new faults detected
      - continue till desired stopping condition is met
    - advantages and issues
      - simple
      - when to quit?
      - how does it perform?

Test generation (contd.)

- Combinational circuit test generation
  - PODEM
    - basics of test generation
      - fault excitation
      - fault propagation
    - D notation
      - explain 5-value logic - 0, 1, x, D, u (D_bar)

Test generation (contd.)

- Combinational circuit test generation
  - PODEM (contd.)
    - algorithm sketch - informal
      - excite fault
        - » choose an unassigned input
        - » place it on decision tree
        - » assign a value to the input and check fault site is D, U, X , or a constant.
          D or U - excited
          X - not yet excited
          constant - same as fault value - BACKTRACK

Test generation (contd.)

- Combinational circuit test generation
  - PODEM (contd.)
      - propagate fault
        - » choose an unassigned input
        - » place it on decision tree
        - » assign a value to the input and check if still D or U in the circuit and if propagated
          if no D or U in the circuit D-frontier (intutively speaking - no gate with in input of D or U and output of X) then backtrack

Test generation (contd.)

- Combinational circuit test generation
  - PODEM (contd.)
      - flow chart from the paper
      - an example circuit for test generation to explain the concepts
        - » back cone
        - » backtrace - different from backtrack
        - » backtracing for desired effects at the correct location
        - » backtracing for desired value
        - » backtracing using easy/hard heuristic

Test generation (contd.)

- Combinational circuit test generation
  - PODEM (contd.)
    - we have a test
      - can it detect more faults?
        - » Fault simulate
        - » fill x's to detect even more faults
          random fill
          deterministic fill
      - fault dropping

Test generation (contd.)

- Sequential circuit test generation
  - checking sequence approach
    - assume knowledge of state description
  - structural approach - gate level description
    - random testing
      - try random input
      - fault simulate
      - compute fault coverage
      - NOT VERY EFFECTIVE GENERALLY

ECE 753 Fault Tolerant Computing 25

Test generation (contd.)

- Sequential circuit test generation
  - structural approach - (contd.)
    - sequential test generation
      - time frame expansion model
      - example of a circuit
      - generate a test using combinational method
      - convert the combinational test to a test seequence

ECE 753 Fault Tolerant Computing 26

Test generation (contd.)

- Design for testabilitiy
  - model of sequential circuit
  - convert the memory elements to a string of connected elements - shift register
  - generate test for combinational circuit
  - test application consists of
    - scan-in
    - apply system clock (apply test and capture responses)
    - scan-out
  - overlapping of scan-in and scan-out

ECE 753 Fault Tolerant Computing 27

Test generation (contd.)

- Built-in self-test
  - concept of Linear Feedback Shift Register (LFSR)
    - a random pattern generator
    - a signature analyser
  - model a sequential circuit as combinational circuit with inputs and outputs
  - convert the input memory elements as a random pattern generator LFSR
  - convert the output memory elements to a signature analyzer

ECE 753 Fault Tolerant Computing 28

Summary

- Basics of testing
- Fault list reduction
- Fault simulation
  - fault coverage computation
- Test Generation
  - combinational circuits - PODEM
  - sequential circuits - time frame expansion
  - DFT - full scan approach
  - BIST - key element LFSR

ECE 753 Fault Tolerant Computing 29