Digital Testing
Lecture 8: Testability Measures

Instructor: Shaahin Hessabi
Department of Computer Engineering
Sharif University of Technology
Adapted from lecture notes prepared by the book authors
Outline

- Controllability and observability
- SCOAP measures
  - Sources of correlation error
  - Combinational circuit example
  - Sequential circuit example
- Test vector length prediction
- High-Level testability measures
- Summary
Purpose

■ Need approximate measure of:
  ➢ Difficulty of setting internal circuit lines to 0 or 1 by setting primary circuit inputs
  ➢ Difficulty of observing internal circuit lines by observing primary outputs

■ Uses:
  ➢ Analysis of difficulty of testing internal circuit parts → redesign or add special test hardware
  ➢ Guidance for algorithms computing test patterns → avoid using hard-to-control lines
  ➢ Estimation of fault coverage
  ➢ Estimation of test vector length
Testability Analysis

- Involves Circuit Topological analysis, but no test vectors and no search algorithm
  - Static analysis
- Linear computational complexity
  - Otherwise, is pointless; might as well use ATPG and calculate:
    - Exact fault coverage
    - Exact test vectors
Types of Measures

- **SCOAP**: Sandia Controllability and Observability Analysis Program

- **Combinational measures:**
  - **CC0**: Difficulty of setting circuit line to logic 0
  - **CC1**: Difficulty of setting circuit line to logic 1
  - **CO**: Difficulty of observing a circuit line

- **Sequential measures – analogous:**
  - **SC0**: Sequential 0-Controllability
  - **SC1**: Sequential 1-Controllability
  - **SO**: Sequential Observability
Range of SCOAP Measures

- Controllabilities: 1 (easiest) to infinity (hardest)
- Observabilities: 0 (easiest) to infinity (hardest)
- Combinational measures:
  - Roughly proportional to # circuit lines that must be set to control or observe given line
- Sequential measures:
  - Roughly proportional to # times a flip-flop must be clocked to control or observe given line
Goldstein’s SCOAP Measures

- **AND gate O/P (output) 0 controllability:**
  \[\text{output}_{\text{controllability}} = \min (\text{input}_{\text{controllabilities}}) + 1\]

- **AND gate O/P 1 controllability:**
  \[\text{output}_{\text{controllability}} = \sum (\text{input}_{\text{controllabilities}}) + 1\]

- **XOR gate O/P controllability**
  \[\text{output}_{\text{controllability}} = \min (\text{controllabilities of each “input set”}) + 1\]

- **Fan-out Stem observability:**
  \[\sum \text{ or } \min (\text{some or all fan-out branch observabilities})\]
Controllability Examples

\[
\begin{align*}
CC0 (a) & \quad CC1 (a) \\
CC0 (b) & \quad CC1 (b) \\
\hline
a & \quad z \quad CC0 (z) = \text{min} (CC0 (a), CC0 (b)) + 1 \\
b & \quad \quad \quad CC1 (z) = CC1 (a) + CC1 (b) + 1 \\
\hline
CC0 (a) & \quad CC1 (a) \\
CC0 (b) & \quad CC1 (b) \\
\hline
a & \quad z \quad CC0 (z) = CC0 (a) + CC0 (b) + 1 \\
b & \quad \quad \quad CC1 (z) = \text{min} (CC1 (a), CC1 (b)) + 1 \\
\hline
CC0 (a) & \quad CC1 (a) \\
CC0 (b) & \quad CC1 (b) \\
\hline
a & \quad z \quad CC0 (z) = \text{min} (CC0 (a) + CC0 (b), CC1 (a) + CC1 (b)) + 1 \\
b & \quad \quad \quad CC1 (z) = \text{min} (CC1 (a) + CC0 (b), CC0 (a) + CC1 (b)) + 1 \\
\hline
CC0 (a) & \quad CC1 (a) \\
CC0 (b) & \quad CC1 (b) \\
\hline
a & \quad z \quad CC0 (z) = CC1 (a) + CC1 (b) + 1 \\
b & \quad \quad \quad CC1 (z) = \text{min} (CC0 (a), CC0 (b)) + 1
\end{align*}
\]
More Controllability Examples

\[
\begin{align*}
\text{CC0} (z) &= \min (\text{CC1} (a), \text{CC1} (b)) + 1 \\
\text{CC1} (z) &= \text{CC0} (a) + \text{CC0} (b) + 1
\end{align*}
\]

\[
\begin{align*}
\text{CC0} (z) &= \min (\text{CC1} (a), \text{CC0} (b), \text{CC0} (a) + \text{CC1} (b)) + 1 \\
\text{CC1} (z) &= \min (\text{CC0} (a), \text{CC0} (b), \text{CC1} (a) + \text{CC1} (b)) + 1
\end{align*}
\]

\[
\begin{align*}
\text{CC0} (z) &= \text{CC1} (a) + 1 \\
\text{CC1} (z) &= \text{CC0} (a) + 1
\end{align*}
\]
Observability Examples

To observe a gate input:
Observe output and make other input values non-controlling

\[
\begin{align*}
CO(a) &= CO(z) + CC1(b) + 1 \\
CO(b) &= CO(z) + CC1(a) + 1 \\
CO(a) &= CO(z) + CC0(b) + 1 \\
CO(b) &= CO(z) + CC0(a) + 1 \\
CO(a) &= CO(z) + \min(CC0(b), CC1(b)) + 1 \\
CO(b) &= CO(z) + \min(CC0(a), CC1(a)) + 1 \\
CO(a) &= CO(z) + CC1(b) + 1 \\
CO(b) &= CO(z) + CC1(a) + 1
\end{align*}
\]
More Observability Examples

\[
\begin{align*}
CO (a) &= CO (z) + CC0 (b) + 1 \\
CO (b) &= CO (z) + CC0 (a) + 1 \\
CO (a) &= CO (z) + \min (CC0 (b), CC1 (b)) + 1 \\
CO (b) &= CO (z) + \min (CC0 (a), CC1 (a)) + 1 \\
CO (a) &= CO (z) + 1 \\
CO (a) &= \min (CO (z1), CO (z2), \ldots, CO (zn))
\end{align*}
\]

To observe a fan-out stem: Observe it through branch with best observability
Error Due to Stems & Reconverging Fanouts

SCOAP measures wrongly, assuming that controlling or observing $x$, $y$, $z$ are independent events

- $CC0\ (x)$, $CC0\ (y)$, $CC0\ (z)$ correlate
- $CC1\ (x)$, $CC1\ (y)$, $CC1\ (z)$ correlate
- $CO\ (x)$, $CO\ (y)$, $CO\ (z)$ correlate
Correlation Error Example

- Exact computation of measures is NP-Complete and impractical
- Italicized (green) measures show correct values; SCOAP measures are in red or bold CC0, CC1 (CO)
  - $(4, \infty)$ refers to observability for 0 and 1.
Sequential Example
Levelization Algorithm 6.1

- Label each gate with max # of logic levels from primary inputs or with max # of logic levels from primary output
- Assign level # 0 to all primary inputs (PIs)
- For each PI fan-out:
  - Label that line with the PI level number, &
  - Queue logic gate driven by that fan-out
- While queue is not empty:
  - Dequeue next logic gate
  - If all gate inputs have level #'s, label the gate with the maximum of them + 1;
  - Else, requeue the gate
Controllability Through Level 0

Circled numbers give level number. (CC0, CC1)
Controllability Through Level 2
Final Combinational Controllability
Combinational Observability for Level 1

Number in square box is level from primary outputs (POs).

(CC0, CC1) CO
Combinational Observabilities for Level 2
Final Combinational Observabilities
Sequential Measure Differences

- **Combinational**
  - Increment $CC0$, $CC1$, $CO$ whenever you pass through a gate, either forwards or backwards

- **Sequential**
  - Increment $SC0$, $SC1$, $SO$ only when you pass through a flip-flop, either forwards or backwards, to $Q$, $\overline{Q}$, $D$, $C$, $SET$, or $RESET$

- **Both**
  - Must iterate on feedback loops until controllabilities stabilize
D Flip-Flop Equations

Assume a \textit{synchronous} RESET line.

- $CC1 (Q) = CC1 (D) + CC1 (C) + CC0 (C) + CC0 (RESET)$ (how many lines must be set, to make $Q=1$)
- $SC1 (Q) = SC1 (D) + SC1 (C) + SC0 (C) + SC0 (RESET) + 1$ (how many FFs must be set, to make $Q=1$)
- $CC0 (Q) = \min [CC1 (RESET) + CC1 (C) + CC0 (C), CC0 (D) + CC1 (C) + CC0 (C)]$
- $SC0 (Q)$ is analogous ($\ldots +1$)
- $CO (D) = CO (Q) + CC1 (C) + CC0 (C) + CC0 (RESET)$
- $SO (D)$ is analogous

![D Flip-Flop Diagram](diagram.png)
D Flip-Flop Clock and Reset

- \( CO (\text{RESET}) = CO (Q) + CC1 (Q) + CC1 (\text{RESET}) + CC1 (C) + CC0 (C) \)
- \( SO (\text{RESET}) \) is analogous
- Three ways to observe the clock line:
  1. Set \( Q \) to 1 and clock in a 0 from \( D \)
  2. Reset the flip-flop and clock in a 1 from \( D \)
  3. Set the flip-flop and then reset it
- \( CO (C) = \min \left( CO (Q) + CC1 (Q) + CC0 (D) + CC1 (C) + CC0 (C), CO (Q) + CC0 (Q) + CC0 (\text{RESET}) + CC1 (D) + CC1 (C) + CC0 (C), CO (Q) + CC1 (Q) + CC1 (\text{RESET}) + CC1 (C) + CC0 (C) \right) \)
- \( SO (C) \) is analogous
Algorithm 6.2: Testability Computation

1. For all PIs, $CC0 = CC1 = 1$ and $SC0 = SC1 = 0$
2. For all other nodes, $CC0 = CC1 = SC0 = SC1 = \infty$
3. Go from PIs to POS, using CC and SC equations to get controllabilities -- Iterate on loops until SC stabilizes -- convergence guaranteed
4. For all POs, set $CO = SO = 0$
5. For all other nodes, $C0 = S0 = \infty$
6. Work from POs to PIs, Use $CO$, $SO$, and controllabilities to get observabilities
7. Fan-out stem $(CO, SO) = \min$ branch $(CO, SO)$
8. If a $CC$ or $SC$ $(CO$ or $SO)$ is $\infty$, that node is uncontrollable (unobservable)
Sequential Example: Initialization
After 1 Iteration
After 2 Iterations
After 3 Iterations
Stable Sequential Measures
Final Sequential Observabilities
Test Vector Length Prediction

- First compute *testabilities* for stuck-at faults
  - $T(x\ sa0) = CC1(x) + CO(x)$
  - $T(x\ sa1) = CC0(x) + CO(x)$
  - *Testability index* $= \log \sum T(f_i)$
    - $\text{all } f_i$
Number of Test Vectors vs. Testability Index

![Graph showing the relationship between number of test vectors and testability index.](image)
High Level Testability

- Build *data path control graph* (DPCG) for circuit
- Compute sequential depth: \# arcs along path between PIs, registers, and POs
- Improve Register Transfer Level Testability with redesign
Improved RTL Design

(a) Untestable implementation.  (b) DFG of untestable implementation.

(c) Testable implementation.  (d) DFG of testable implementation.
Summary

- Testability approximately measures:
  - Difficulty of setting circuit lines to 0 or 1
  - Difficulty of observing internal circuit lines

- Uses:
  - Analysis of difficulty of testing internal circuit parts
    - Redesign circuit hardware or add special test hardware where measures show bad controllability or observability
  - Guidance for algorithms computing test patterns: avoid using hard-to-control lines
  - Estimation of fault coverage: 3-5 % error
  - Estimation of test vector length