Testability Measures

Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,

IIIT Hyderabad, India

Email: src.vlsi@iiit.ac.in

Testability Measures

- Origins
- 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

Origins

- **Control theory**
- **Rutman 1972 -- First definition of controllability**
- **Goldstein 1979 -- SCOAP**
  - **First definition of observability**
  - **First elegant formulation**
  - **First efficient algorithm to compute controllability and observability**
- **Parker & McCluskey 1975**
  - **Definition of Probabilistic Controllability**
- **Brglez 1984 -- COP**
  - **1st probabilistic measures**
- **Seth, Pan & Agrawal 1985 – PREDICT**
  - **1st exact probabilistic measures**

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 automatic test-pattern generation 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***
  - ***SC1***
  - ***SO***

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 0 controllability:**

  output_controllability = min (input_controllabilities) + 1

- **AND gate O/P 1 controllability:**

  output_controllability = Σ (input_controllabilities) + 1

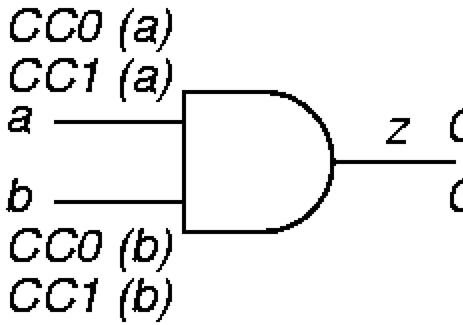
- **XOR gate O/P controllability**

  output_controllability = min (controllabilities of each input set) + 1

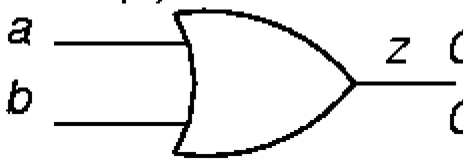
- **Fanout Stem observability:**

  Σ or min (some or all fanout branch observabilities)

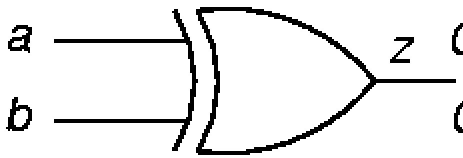
Controllability Examples



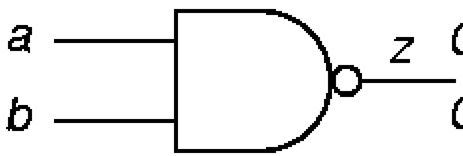
$$CC0\ (z) = \min\ (CC0\ (a),\ CC0\ (b)) + 1$$
$$CC1\ (z) = CC1\ (a) + CC1\ (b) + 1$$



$$CC0\ (z) = CC0\ (a) + CC0\ (b) + 1$$
$$CC1\ (z) = \min\ (CC1\ (a),\ CC1\ (b)) + 1$$

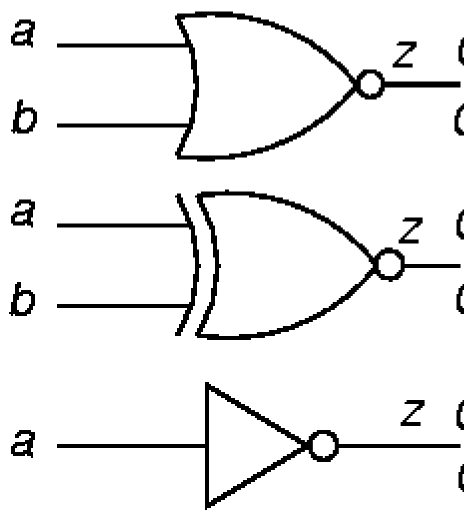


$$CC0\ (z) = \min\ (CC0\ (a) + CC0\ (b),\ CC1\ (a) + CC1\ (b)) + 1$$
$$CC1\ (z) = \min\ (CC1\ (a) + CC0\ (b),\ CC0\ (a) + CC1\ (b)) + 1$$



$$CC0\ (z) = CC1\ (a) + CC1\ (b) + 1$$
$$CC1\ (z) = \min\ (CC0\ (a),\ CC0\ (b)) + 1$$

More Controllability Examples



$CC0(z) = \min(CC1(a), CC1(b)) + 1$

$CC1(z) = CC0(a) + CC0(b) + 1$

$CC0(z) = \min(CC1(a) + CC0(b), CC0(a) + CC1(b)) + 1$

$CC1(z) = \min(CC0(a) + CC0(b), CC1(a) + CC1(b)) + 1$

$CC0(z) = CC1(a) + 1$

$CC1(z) = CC0(a) + 1$

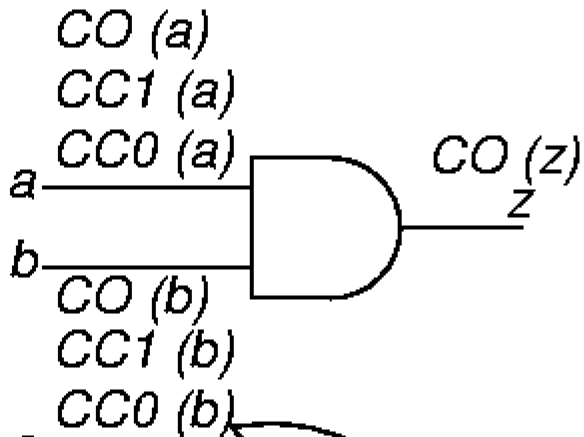
Observability Examples

To observe a gate input:

Observe output and make other input values non-controlling

$$CO(a) = CO(z) + CC1(b) + 1$$

$$CO(b) = CO(z) + CC1(a) + 1$$

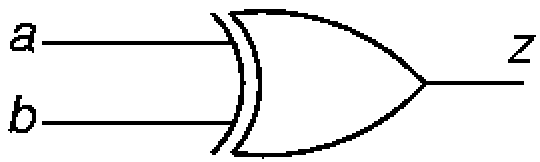


$$CO(a) = CO(z) + CC0(b) + 1$$

$$CO(b) = CO(z) + CC0(a) + 1$$

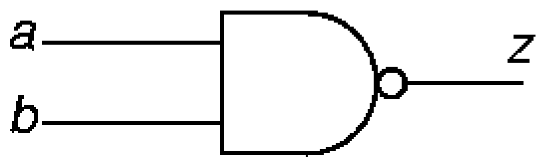
$$CO(a) = CO(z) + \min(CC0(b), CC1(b)) + 1$$

$$CO(b) = CO(z) + \min(CC0(a), CC1(a)) + 1$$



$$CO(a) = CO(z) + CC1(b) + 1$$

$$CO(b) = CO(z) + CC1(a) + 1$$



More Observability Examples

To observe a fanout stem:

Observe it through branch with best observability

$CO\ (a) = CO\ (z) + CC0\ (b) + 1$

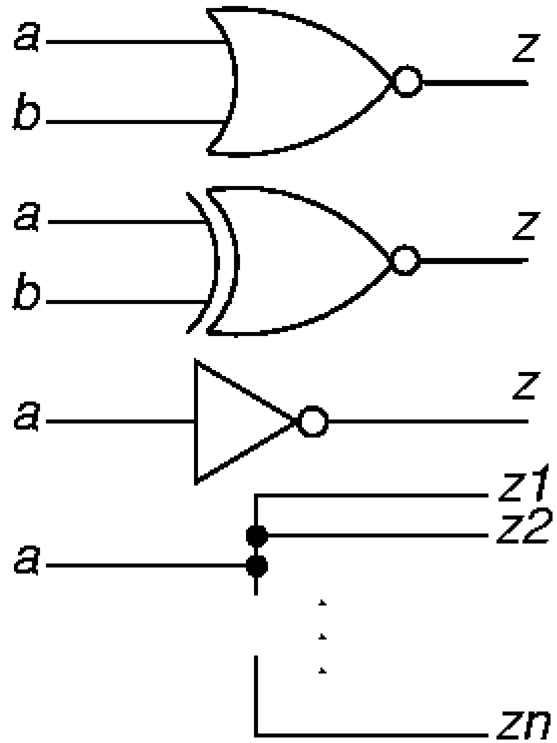
$CO\ (b) = CO\ (z) + CC0\ (a) + 1$

$CO\ (a) = CO\ (z) + \min\ (CC0\ (b), CC1\ (b)) + 1$

$CO\ (b) = CO\ (z) + \min\ (CC0\ (a), CC1\ (a)) + 1$

$CO\ (a) = CO\ (z) + 1$

$CO\ (a) = \min\ (CO\ (z1), CO\ (z2), ..., CO\ (zn))$

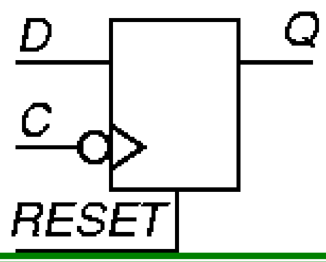


Sequential Measure Differences

- **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, Q, D, C, SET*, or *RESET***
- **Both**
  - **Must iterate on feedback loops until controllabilities stabilize**

D Flip-Flop Equations

- **Assume a synchronous RESET line.**
- ***CC1* (*Q*) = *CC1* (*D*) + *CC1* (*C*) + *CC0* (*C*) + *CC0* (*RESET*)**
- ***SC1* (*Q*) = *SC1* (*D*) + *SC1* (*C*) + *SC0* (*C*) + *SC0* (*RESET*) + 1**
- ***CC0* (*Q*) = min [*CC1* (*RESET*) + *CC1* (*C*) + *CC0* (*C*), *CC0* (*D*) + *CC1* (*C*) + *CC0* (*C*)]**
- ***SC0* (*Q*) is analogous**
- ***CO* (*D*) = *CO* (*Q*) + *CC1* (*C*) + *CC0* (*C*) + *CC0* (*RESET*)**
- ***SO* (*D*) is analogous**



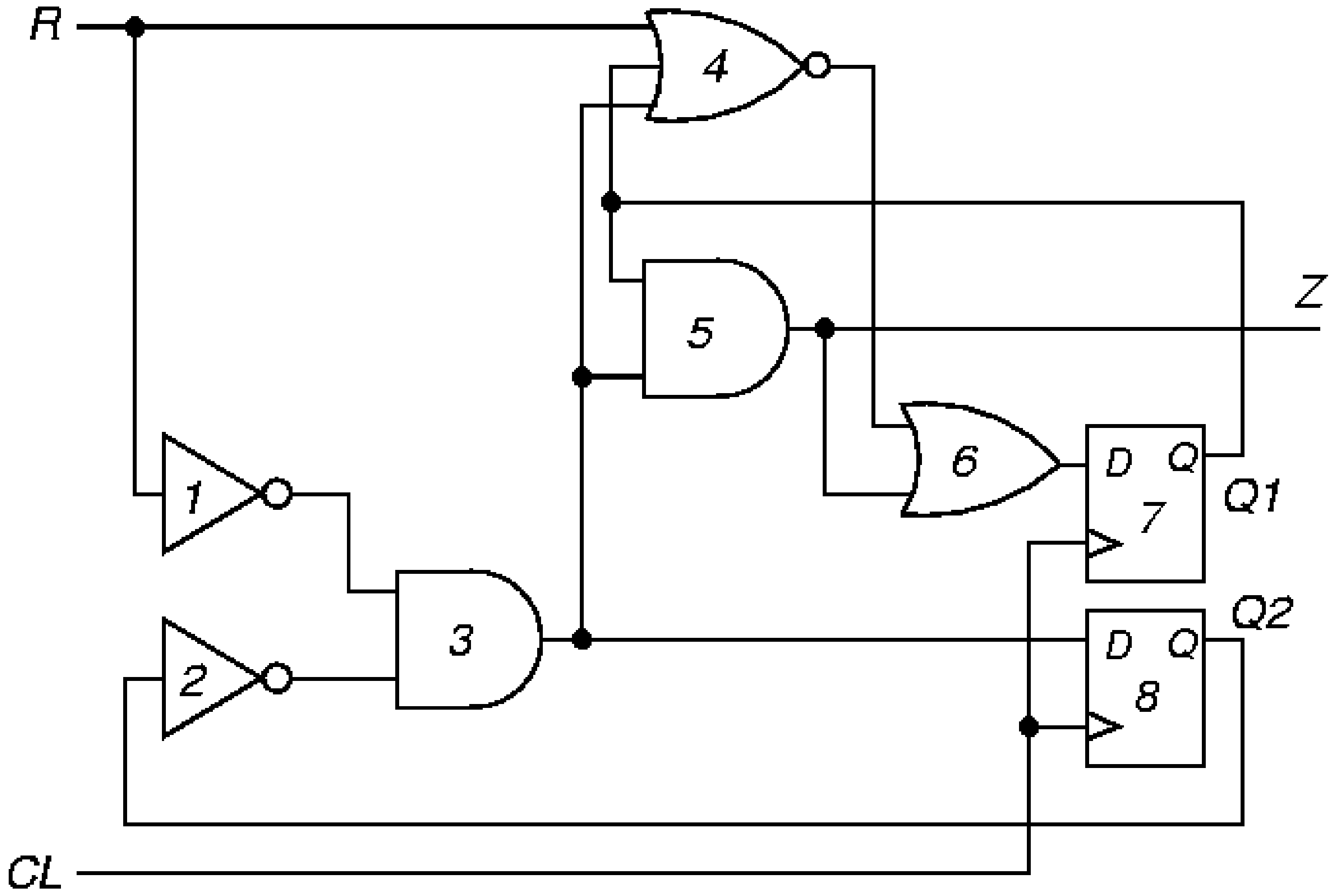
D Flip-Flop Clock and Reset

- CO ($RESET$) = CO (Q) + $CC1$ (Q) + $CC1$ (D) + $CC1$ (C) + $CC0$ (C)
- SO ($RESET$) is analogous
- Three ways to observe the clock line:
  1. Set Q to 1 and clock in a 0 from D
  2. Set the flip-flop and then reset it
  3. Reset the flip-flop and clock in a 1 from D
- CO (C) = min [CO (Q) + $CC1$ (Q) + $CC0$ (D) +
  $CC1$ (C) + $CC0$ (C),
  CO (Q) + $CC1$ (Q) + $CC1$ ($RESET$) +
  $CC1$ (C) + $CC0$ (C),
  CO (Q) + $CC0$ (Q) + $CC0$ ($RESET$) +
  $CC1$ (D) + $CC1$ (C) + $CC0$ (C)]
- SO (C) is analogous

Algorithm for Testability Computation

1. **For all PIs, *CC0* = *CC1* = 1 and *SC0* = *SC1* = 0**
2. **For all other nodes, *CC0* = *CC1* = *SC0* = *SC1* = ∞**
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, ∞ for other nodes**
5. **Work from POs to PIs, Use *CO*, *SO*, and controllabilities to get observabilities**
6. **Fanout stem (*CO*, SO) = min branch (CO, SO)**
7. **If a *CC* or *SC* (*CO* or *SO*) is ∞, that node is uncontrollable (unobservable)**

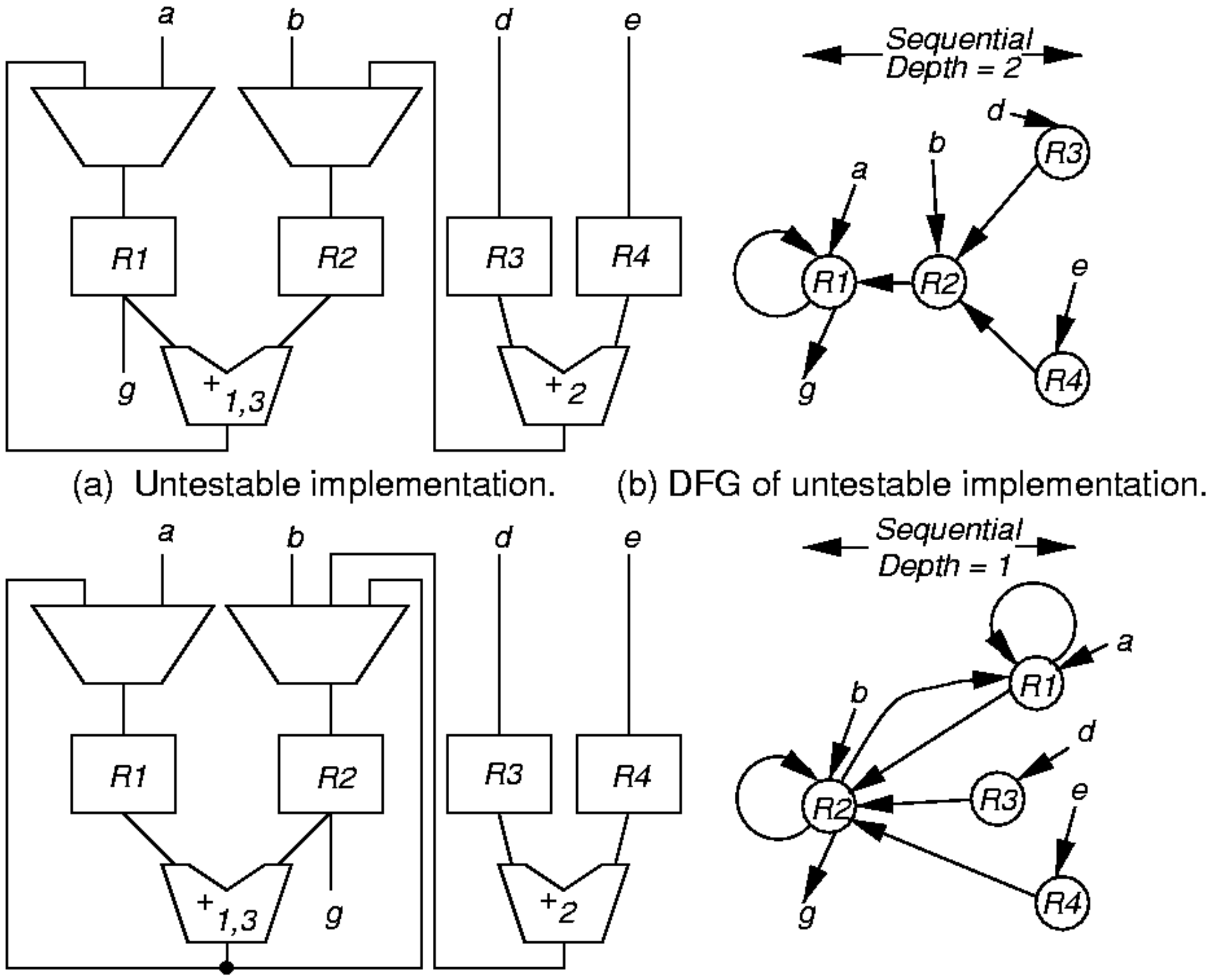
Sequential Example



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.

Probability based testability measures

- Used to analyze the **random testability** of the circuit
  - $C0(s)$: probability-based 0-controllability of s
  - $C1(s)$: probability-based 1-controllability of s
  - $O(s)$: probability-based observability of s
- Range between 0 and 1
- $C0(s) + C1(s) = 1$

Probability based controllability calculation rules

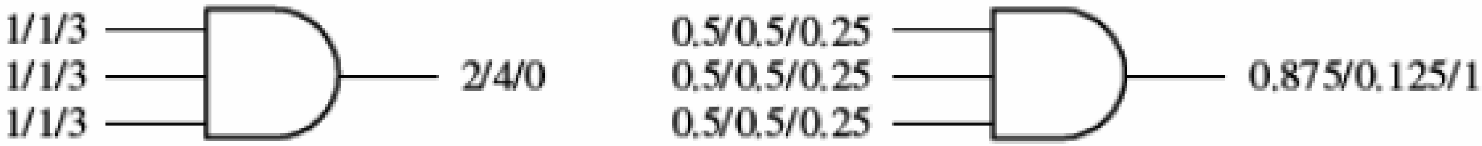
	0-controllability (Primary input, output, branch)	**1-controllability** (Primary input, output, branch)
Primary Input	p_0	$p_1 = 1 - p_0$
AND	1 – (output 1-controllability)	Π (input 1-controllabilities)
OR	Π (input 0-controllabilities)	1 – (output 0-controllability)
NOT	Input 1-controllability	Input 0-controllability
NAND	Π (input 1-controllabilities)	1 – (output 0-controllability)
NOR	1 – (output 1-controllability)	Π (input 0-controllabilities)
BUFFER	Input 0-controllability	Input 1-controllability
XOR	1 – 1-controllabilty	$\Sigma\,(C1(a) \times C0(b), C0(a) \times C1(b))$
XNOR	1 – 1-controllability	$\Sigma\,(C0(a) \times C0(b), C1(a) \times C1(b))$
Branch	Stem 0-controllability	Stem 1-controllability

Probability based observability calculation rules

	Observability (Primary output, input, stem)
Primary Output	1
AND / NAND	Π (output observability, 1-controllabilities of other inputs)
OR / NOR	Π (output observability, 0-controllabilities of other inputs)
NOT / BUFFER	Output observability
XOR / XNOR	a: Π (output observability, *max* {0-controllability of b, 1-controllability of b}) b: Π (output observability, *max* {0-controllability of a, 1-controllability of a})
Stem	*max* {branch observabilities}

a, b: inputs of an XOR or XNOR gate

Difference between SCOAP abd Probability based testability measures



(a) SCOAP combinational measures (b) Probability-based measures

$v1/v2/v3$ represents the signal's 0-controllability ($v1$), 1-controllability ($v2$), and observability ($v3$)

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**

Questions?