Lecture 4 – Testability Measures

- Purpose, origins
- Analysis, measures and computation
- 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
- Brglez 1984 -- COP
- Seth, Pan & Agrawal 1985

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

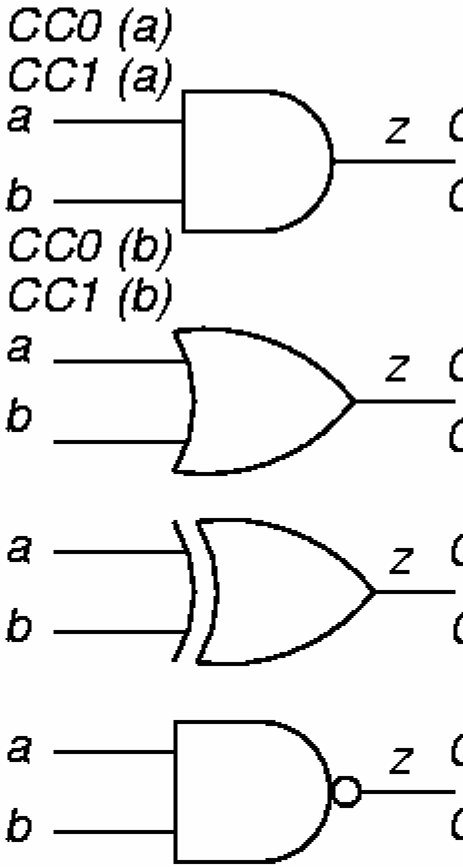
Types of Measures

- 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

Controllability Examples



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

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

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

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

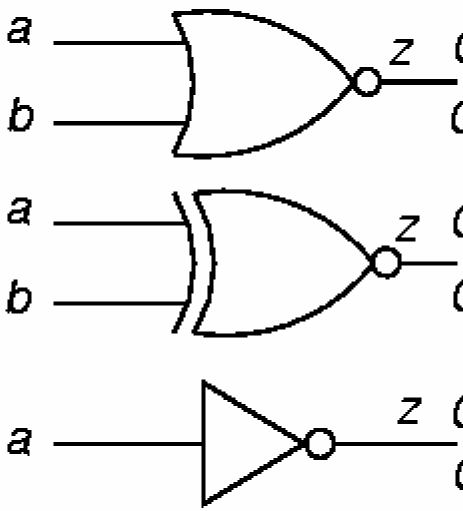
$CC0(z) = \min(CC0(a) + CC0(b), CC1(a) + CC1(b)) + 1$

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

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

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

More Controllability Examples



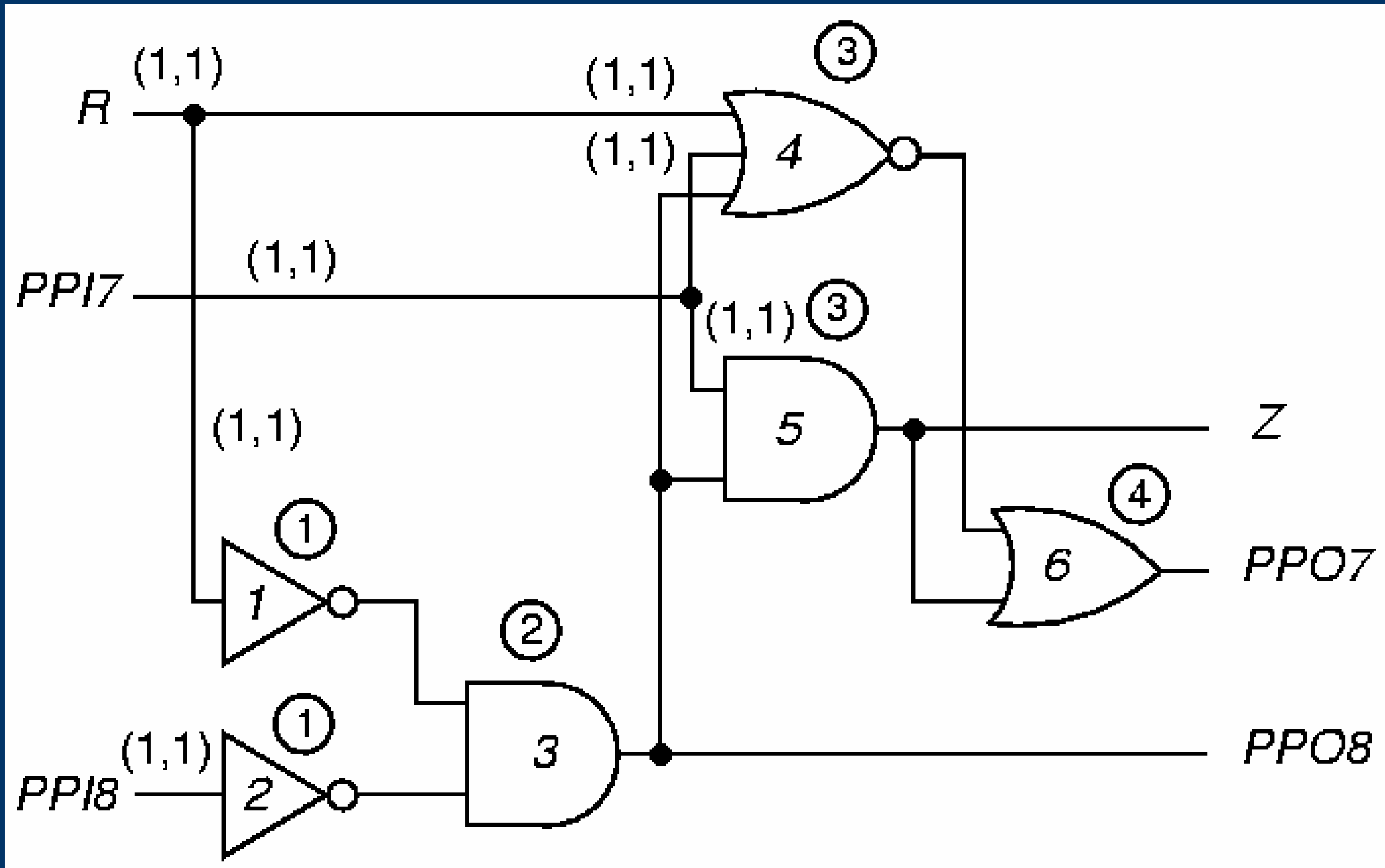
$$CC0(z) = \min(CC1(a), CC1(b)) + 1$$
$$CC1(z) = CC0(a) + CC0(b) + 1$$

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

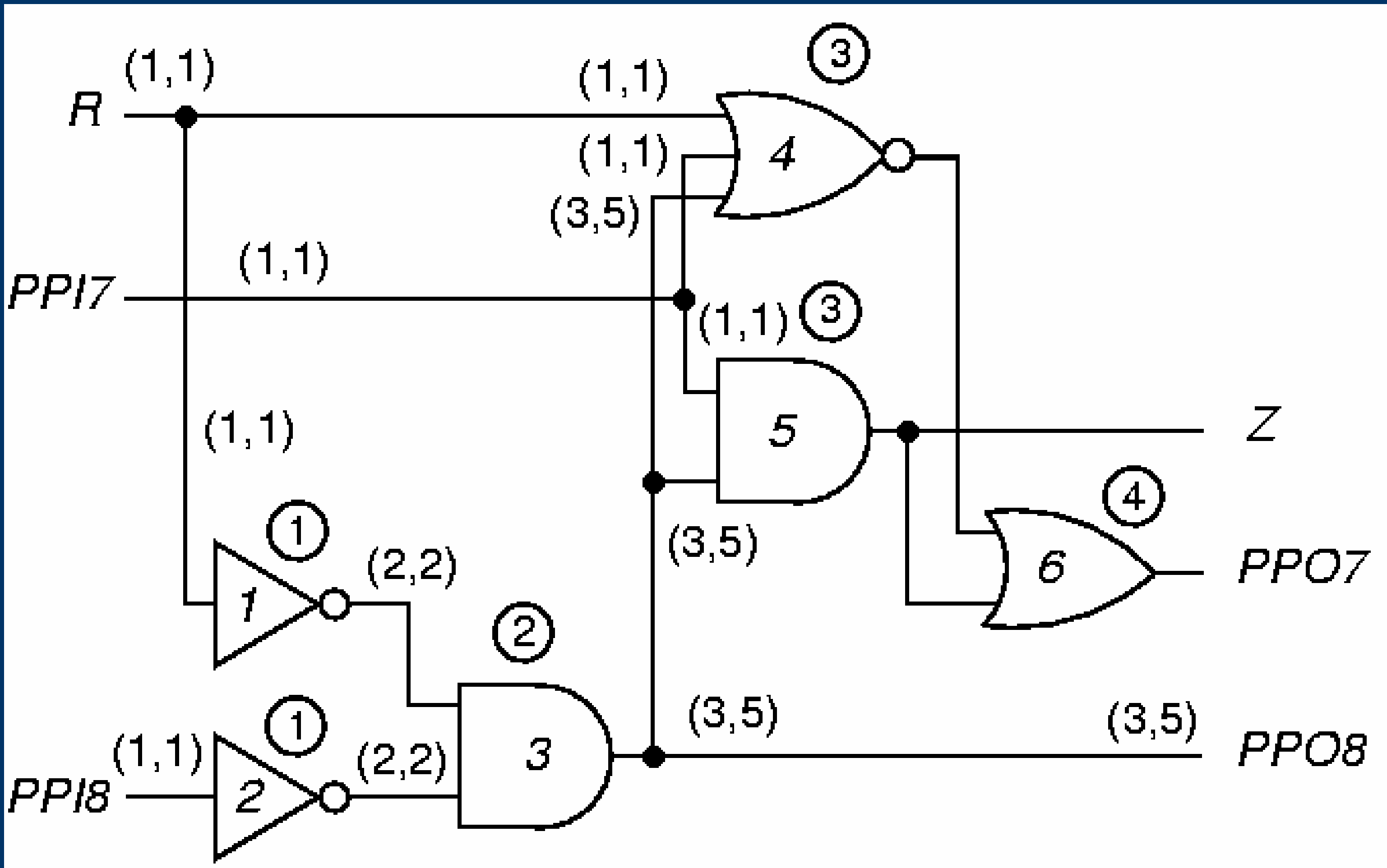
$$CC0(z) = CC1(a) + 1$$
$$CC1(z) = CC0(a) + 1$$

Controllability Through Level 0

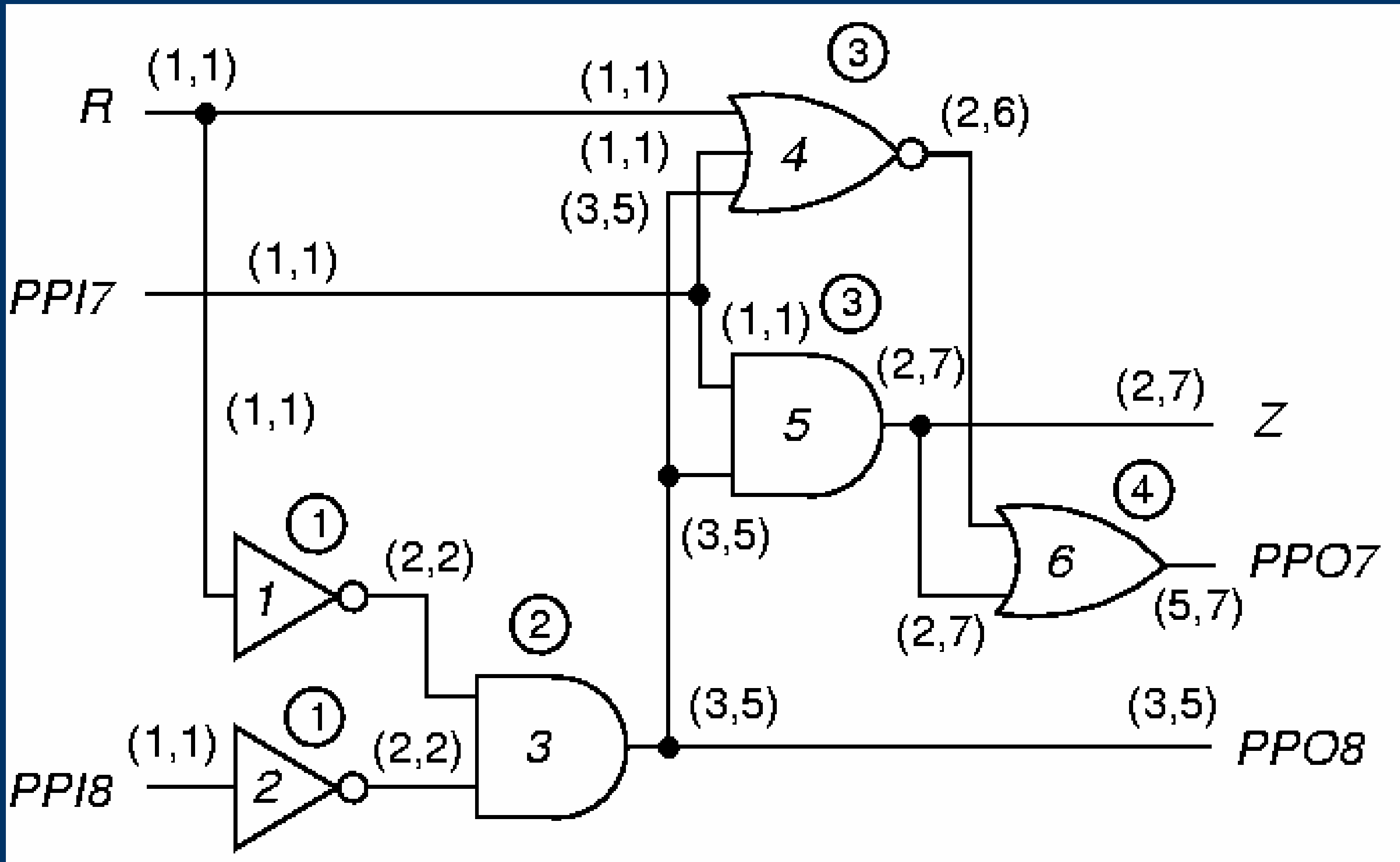
Circled numbers give level number. (CC0, CC1)



Controllability Through Level 2



Final Combinational Controllability



Observability Examples

To observe a gate input:

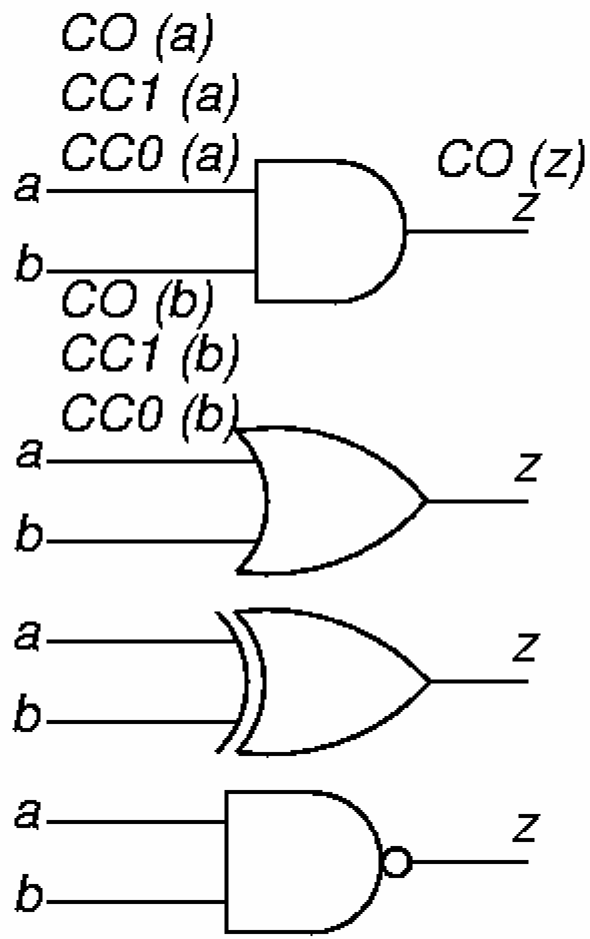
Observe output and make other input values non-controlling

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

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

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

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



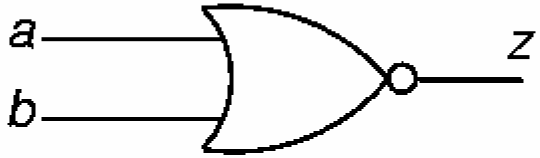
More Observability Examples

To observe a fanout stem:

Observe it through branch with best observability

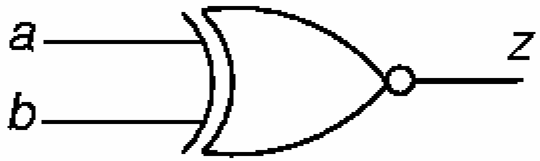
$$CO\ (a) = CO\ (z) + CC0\ (b) + 1$$

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

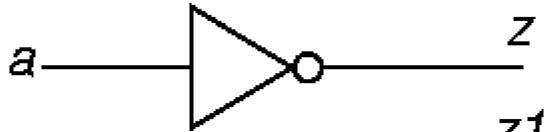


$$CO\ (a) = CO\ (z) + \min\ (CC0\ (b),\ CC1\ (b)) + 1$$

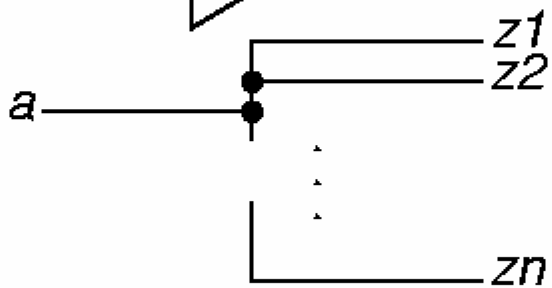
$$CO\ (b) = CO\ (z) + \min\ (CC0\ (a),\ CC1\ (a)) + 1$$



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



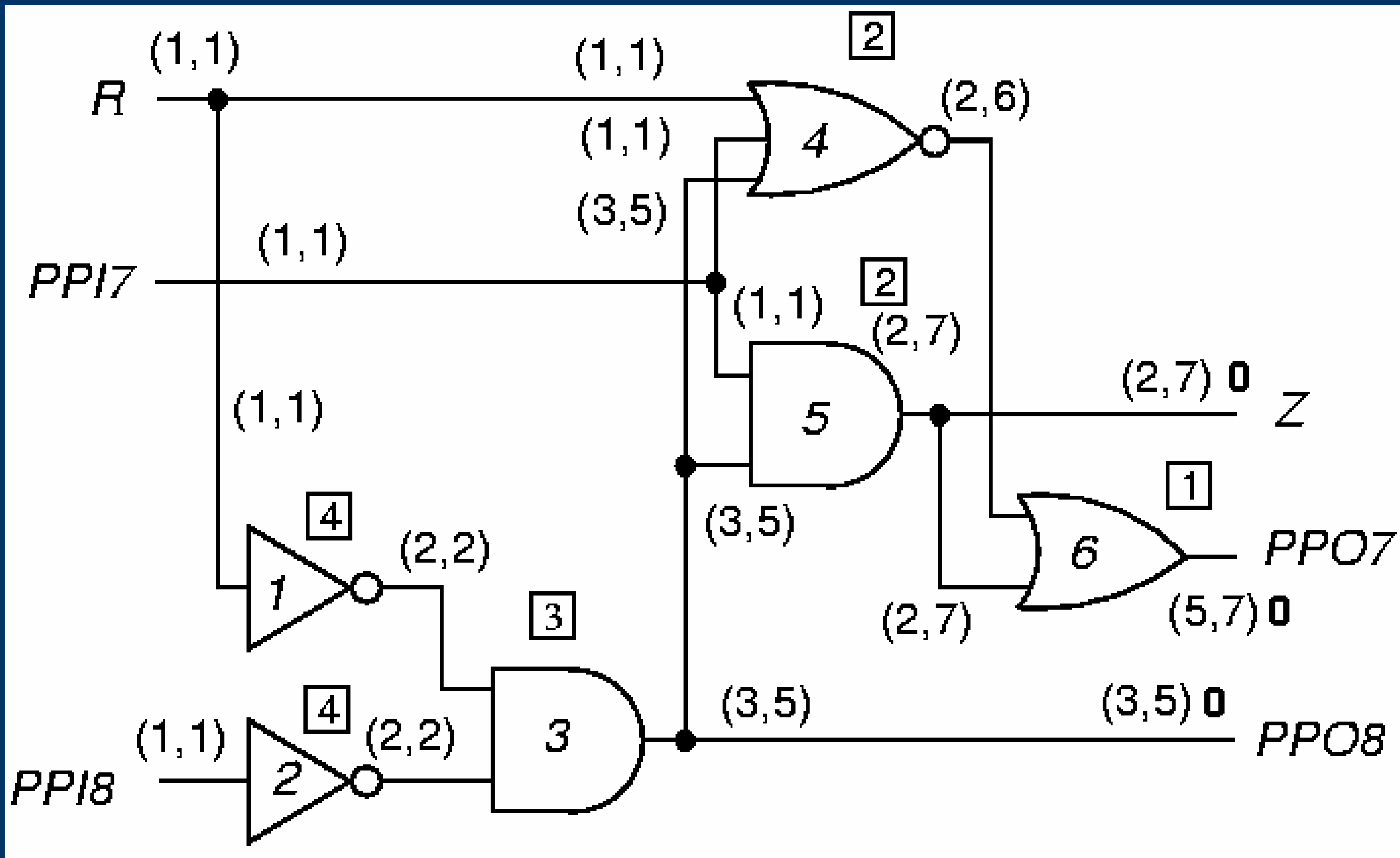
$$CO\ (a) = \min\ (CO\ (z1),\ CO\ (z2),\ \ldots,\ CO\ (zn))$$



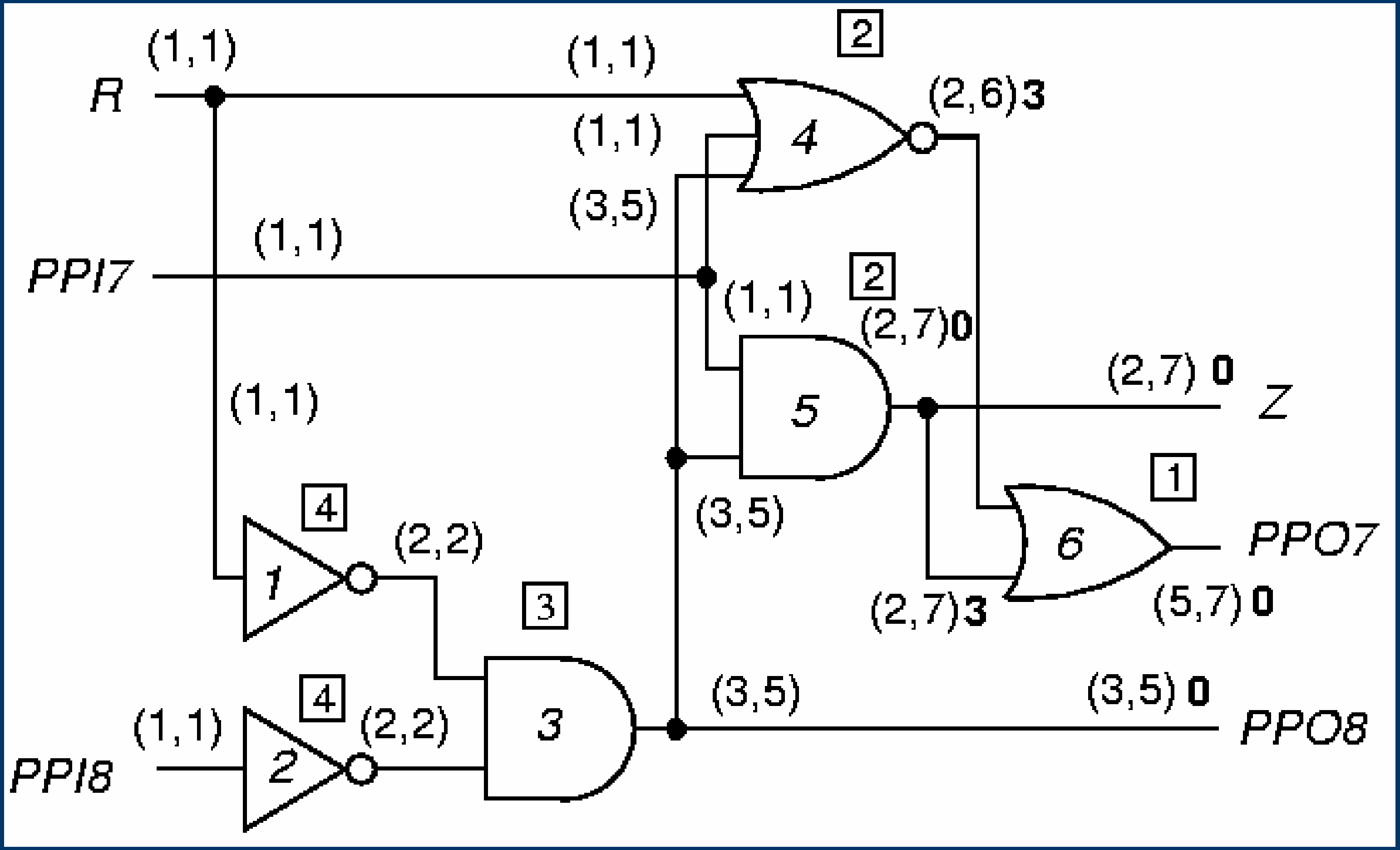
Combinational Observability for Level 1

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

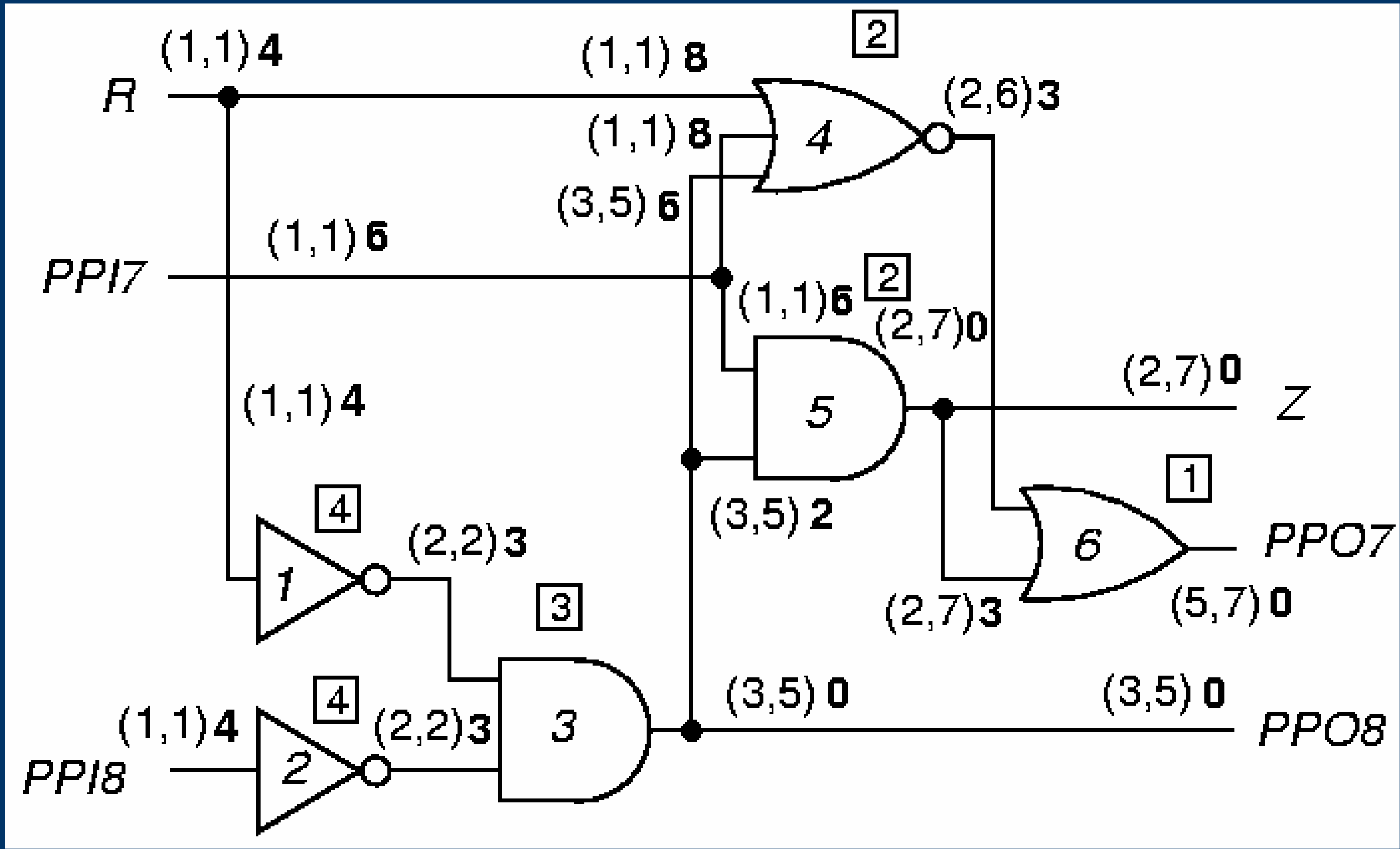
(CC0, CC1) CO



Combinational Observabilities for Level 2



Final Combinational Observabilities



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**
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)

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
  - Estimation of test vector length