Chapter 4
Testability Analysis

Arnaud Virazel

virazel@lirmm.fr

Purpose

- Measurements used to guide
  - Automatic Test Pattern Generation (ATPG)
  - Design-for-Testability (DfT)
- Testability?
  - Characteristic of a circuit that influences various costs associated with the test (length, complexity of generating a sequence, etc.)

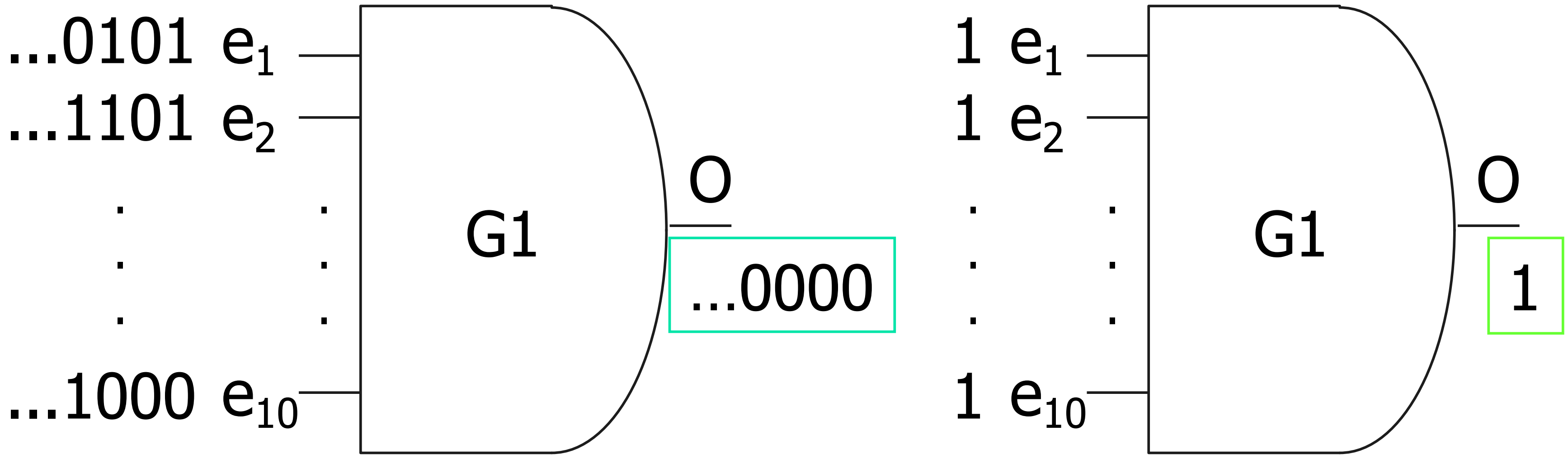
Measurements

- Controllability
  - Indicates the relative difficulty of positioning a line at 0 or at 1 from the Primary Inputs (PIs)
- Observability
  - Indicates the relative difficulty of propagating an error from a line to the Primary Outputs (POs)

Example

Sa0 test at a AND-gate output

- Generation of random vectors
  - Control-at-1 at G1 output is "hard" (1/2^n probability)
- Generation of deterministic vectors
  - Control-at-1 at G1 output is "simple"



Computation Methods

- Testability measurements for a deterministic test
  - **SCOAP** (Sandia Controllability and Observability Analysis Program)
- Testability measurements for a random or pseudo-random test
  - **COP** (Controllability Observability Probability)

SCOAP Principle

- Based on a structural analysis of the DUT
- DUT = cells (combinatorial, sequential)
  + nets (combinatorial, sequential)
- Increasing SCOP measurements with the required testing effort

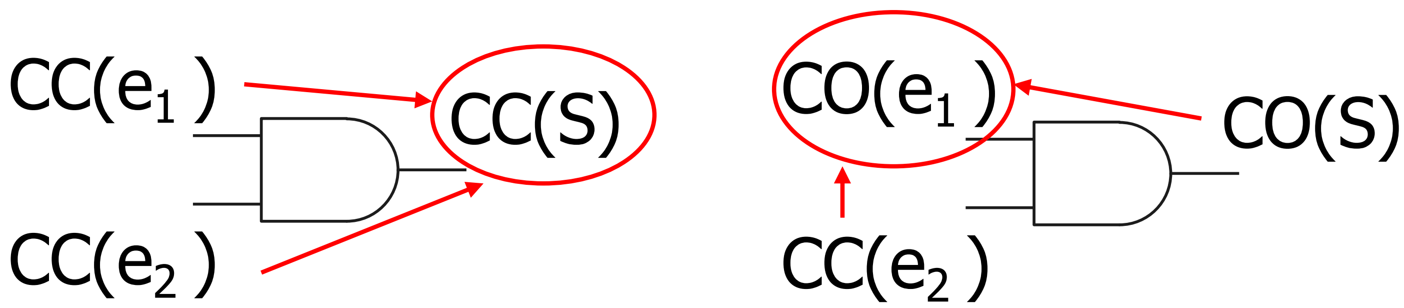
SCOAP Measurements

- Six measurements associated to each net:
  - CC0(N) = Combinatorial Controllability at 0 = # minimum of comb. nets that must be controlled to bring a logic 0 at net N
  - CC1(N) = Combinatorial Controllability at 1 = # minimum of comb. nets that must be controlled to bring a logic 1 at net N
  - CO(N) = Combinatorial Observability = # minimum of comb. nets that must be controlled so that the effect of the fault on net N is propagated towards a PO

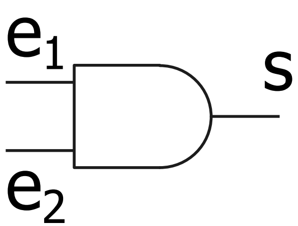
  - SC0(N) = Sequential Controllability at 0 = # minimum of seq. nets that must be controlled to bring a logic 0 at net N
  - SC1(N) = Sequential Controllability at 1 = # minimum of seq. nets that must be controlled to bring a logic 1 at net N
  - SO(N) = Sequential Observability = # minimum of seq. nets that must be controlled so that the effect of the fault on net N is propagated towards a PO

SCOAP Evaluation

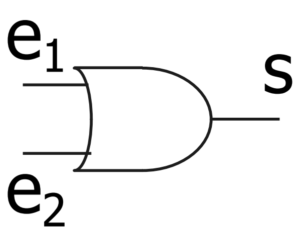
- Controllability at the output of a cell as a function of the controllability of its inputs
- Observability at the input of a cell as a function of the observability at the output and the controllability of the other inputs



SCOAP Combinatorial Cells



- CC0(s) = min (CC0(e1),CC0(e2)) + 1
- CC1(s) = CC1(e1) + CC1(e2) + 1
- CO(e1) = CC1(e2) + CO (s) + 1

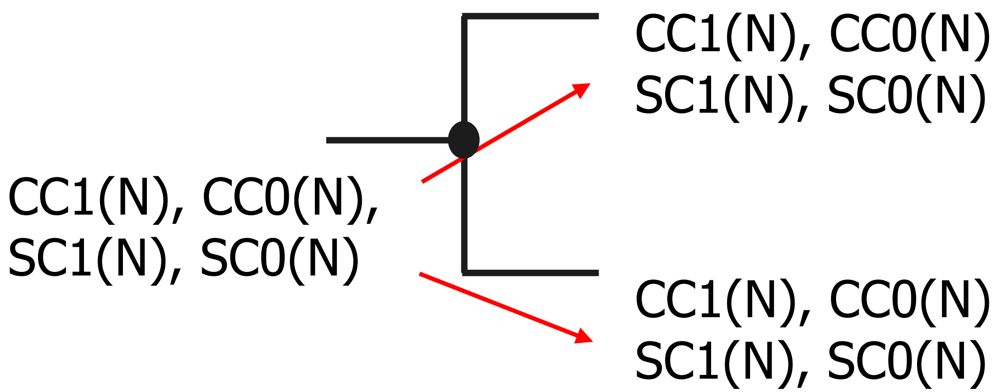


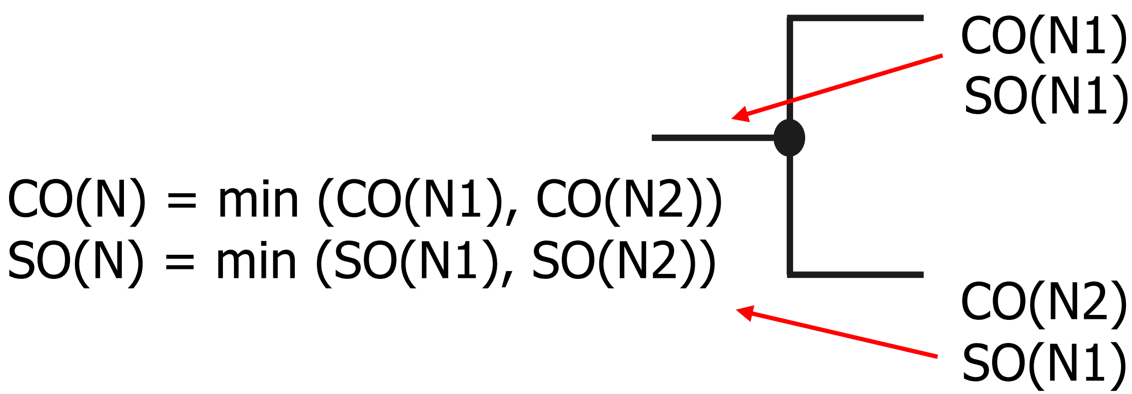
- CC0(s) = CC0 (e1) + CC0 (e2) + 1
- CC1(s) = min (CC1(e1),CC1(e2))+1
- CO(e1) = CC0(e2) + CO (s) + 1

Classical Logic Gates

Combinatorial Controllability at 0	AND2	CC0(OUT) = min(CC0(IN1),CC0(IN2)) + 1
	NAND2	CC0(OUT) = CC1(IN1) + CC1(IN2) + 1
	OR2	CC0(OUT) = CC0(IN1) + CC0(IN2) + 1
	NOR2	CC0(OUT) = min(CC1(IN1),CC1(IN2)) + 1
	INV	CCO(OUT) = CC1(IN) + 1
	BUF	CCO(OUT) = CC0(IN) + 1
Combinatorial Controllability at 1	AND2	CC1(OUT) = CC1(IN1) + CC1(IN2)) + 1
	NAND2	CC1(OUT) = min(CC0(IN1),CC0(IN2) + 1
	OR2	CC1(OUT) = min(CC1(IN1),CC1(IN2)) + 1
	NOR2	CC1(OUT) = CC0(IN1) + CC0(IN2)) + 1
	INV	CC1(OUT) = CC0(IN) + 1
	BUF	CC1(OUT) = CC1(IN) + 1
Combinatorial Observability	AND2	CO(IN1) = CC1(IN2) + CO(OUT) + 1
	NAND2	CO(IN1) = CC1(IN2) + CO(OUT) + 1
	OR2	CO(IN1) = CC0(IN2) + CO(OUT) + 1
	NOR2	CO(IN1) = CC0(IN2) + CO(OUT) + 1
	INV	CO(IN1) = CO(OUT) + 1
	BUF	CO(IN1) = CO(OUT) + 1

SCOAP Fanout

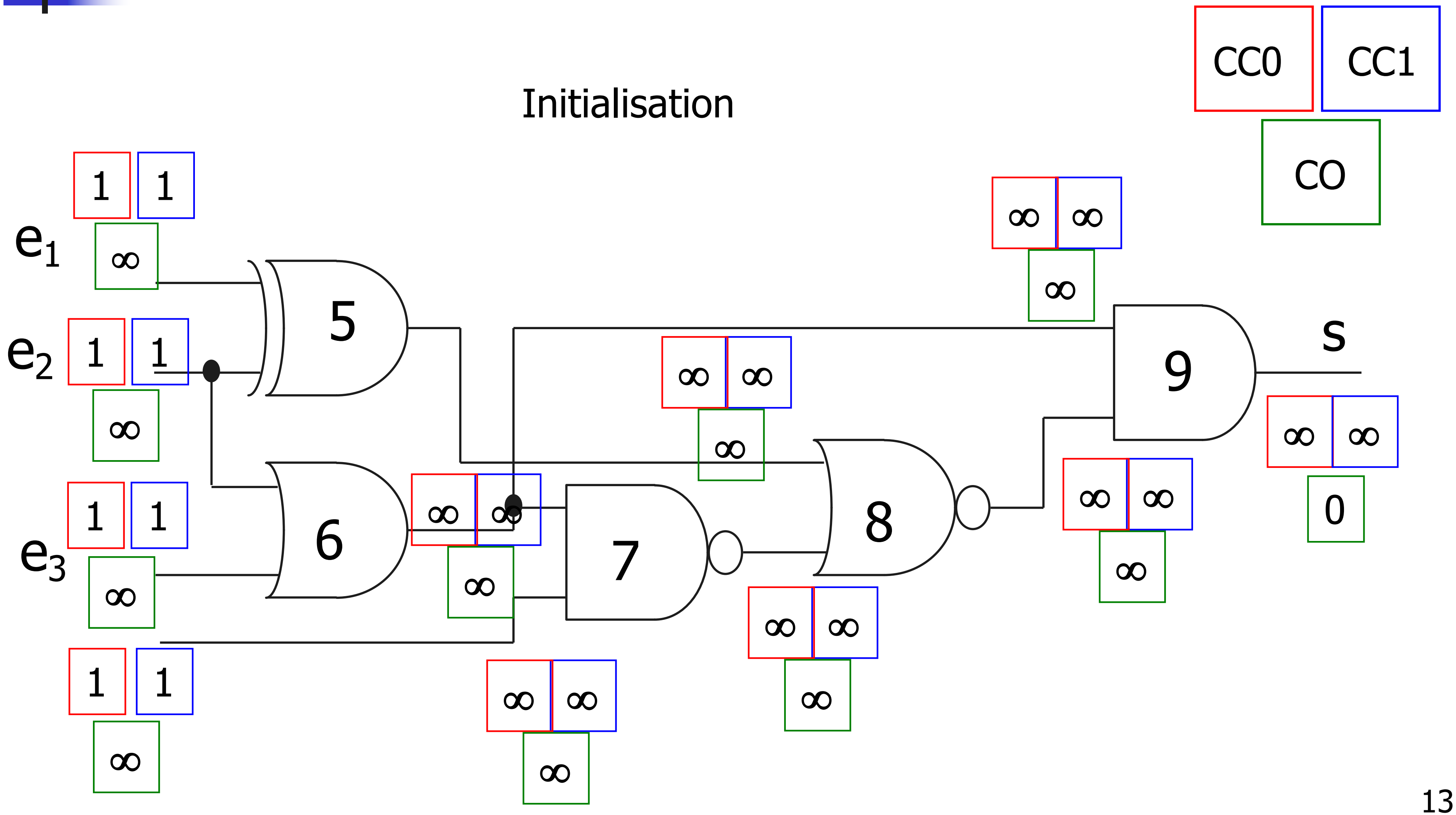




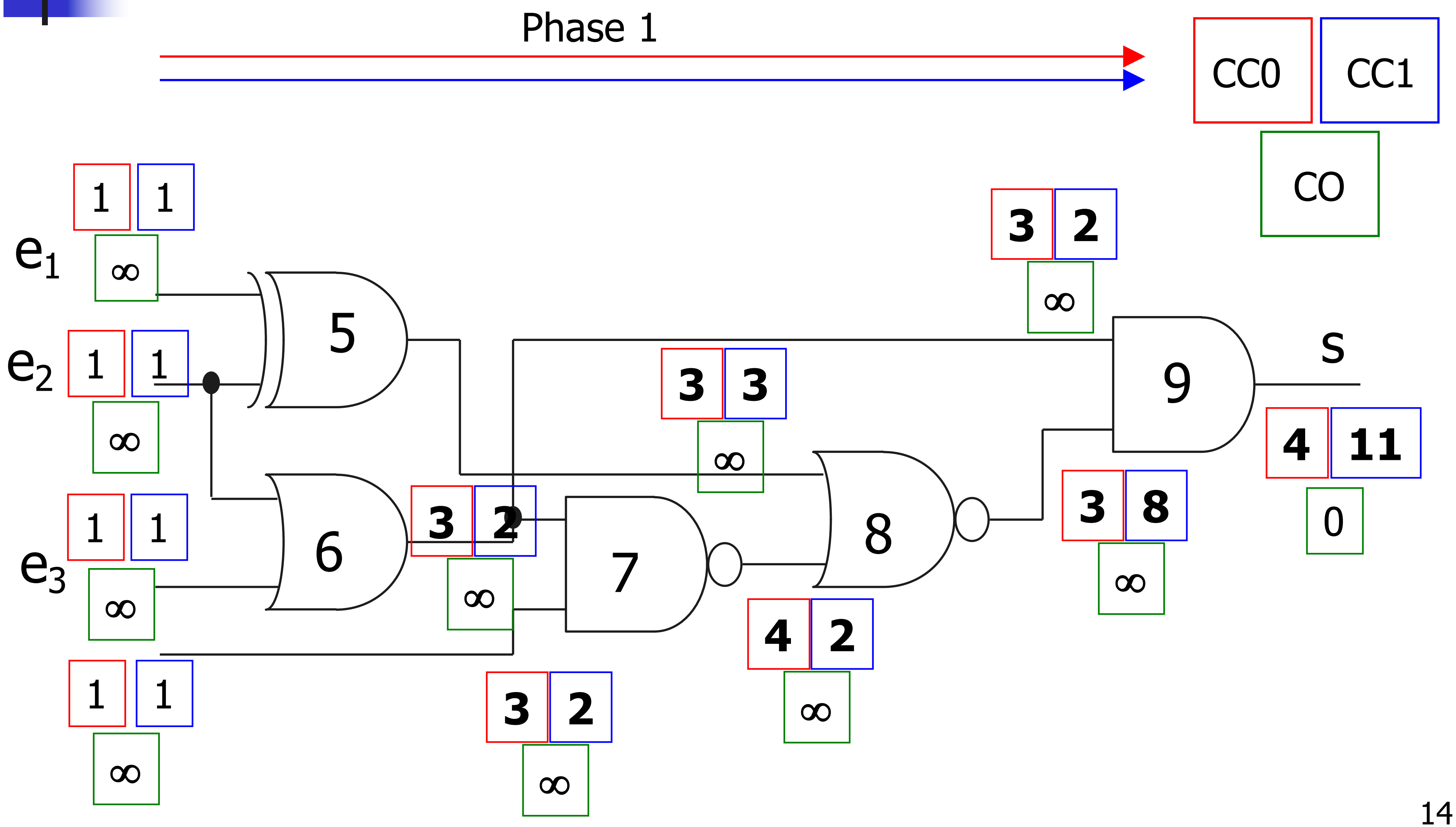
SCOAP Process

- Net initialisation:
  - If N = PI ➔ $CC0(N) = CC1(N) = 1$
    ➔ $SC0(N) = SC1(N) = 0$
  - If N = PO ➔ $CO(N) = SO(N) = 0$
  - Else ➔ $CC0 = CC1 = SC0 = SC1 = \infty$
- Phase 1:
  - Controllability computation form PIs to POs
- Phase 2:
  - Observability computation form POs to PIs

Example

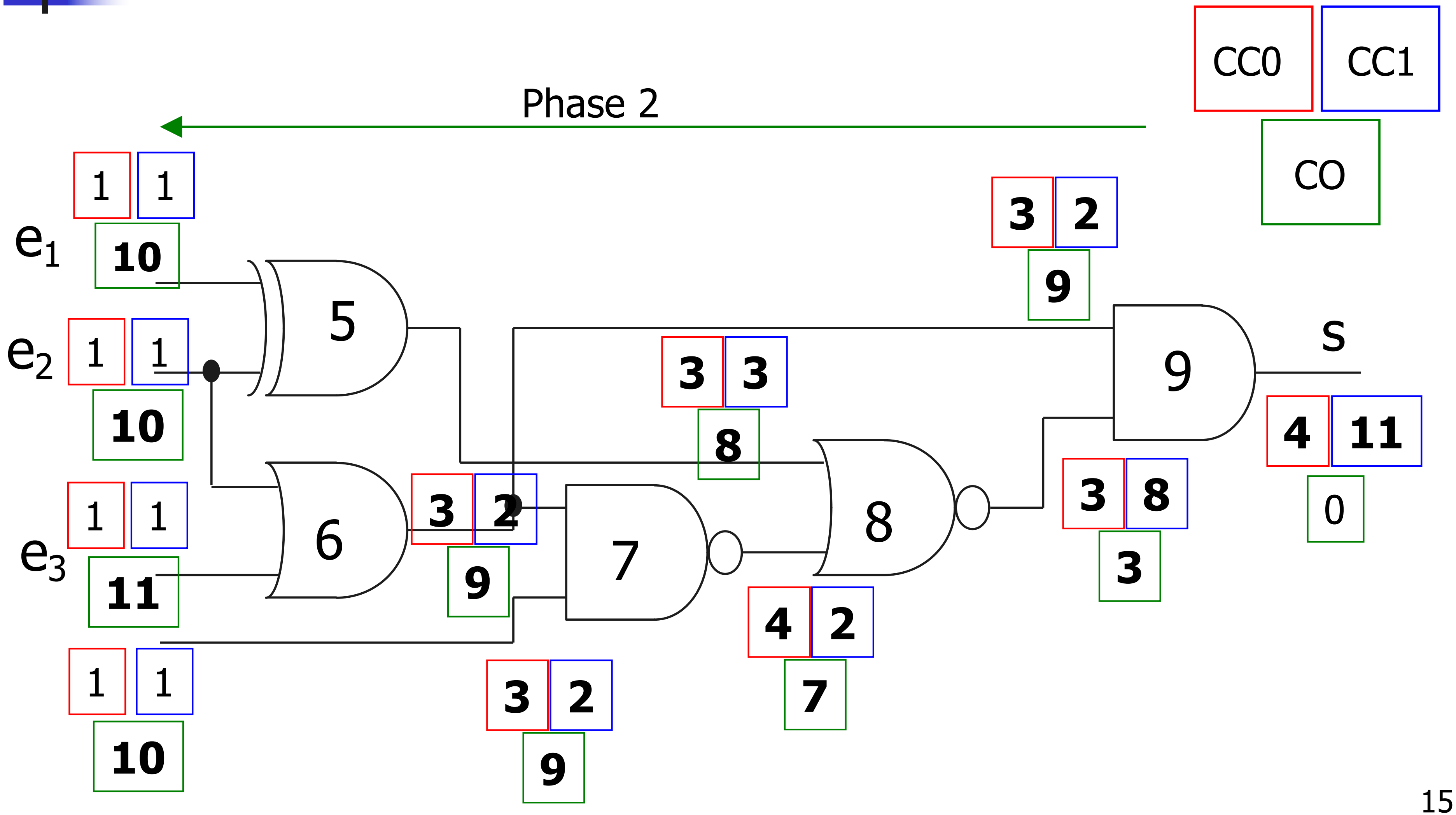


Example



Phase 1
CC0
CC1
CO
e1
e2
e3
1
1
∞
1
1
∞
1
1
∞
1
1
∞
5
6
7
8
9
s
3
2
∞
3
2
∞
3
3
∞
4
2
∞
3
8
∞
3
2
∞
4
11
0

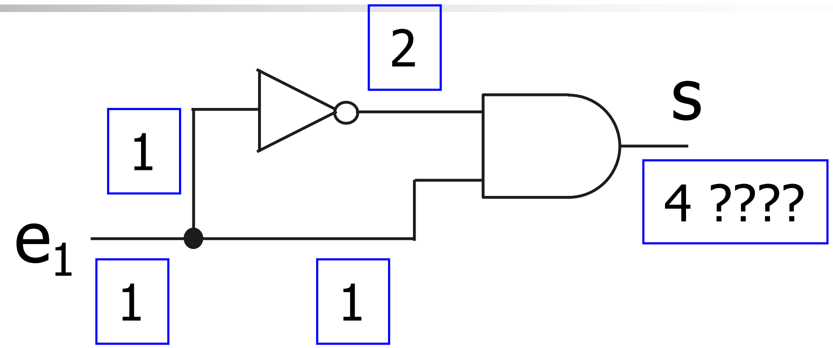
Example



CC0
CC1
CO
Phase 2
e1
e2
e3
s
5
6
7
8
9
1 1 10
1 1 10
1 1 11
1 1 10
3 2 9
3 3 8
3 2 9
3 2 9
4 2 7
3 8 3
4 11 0

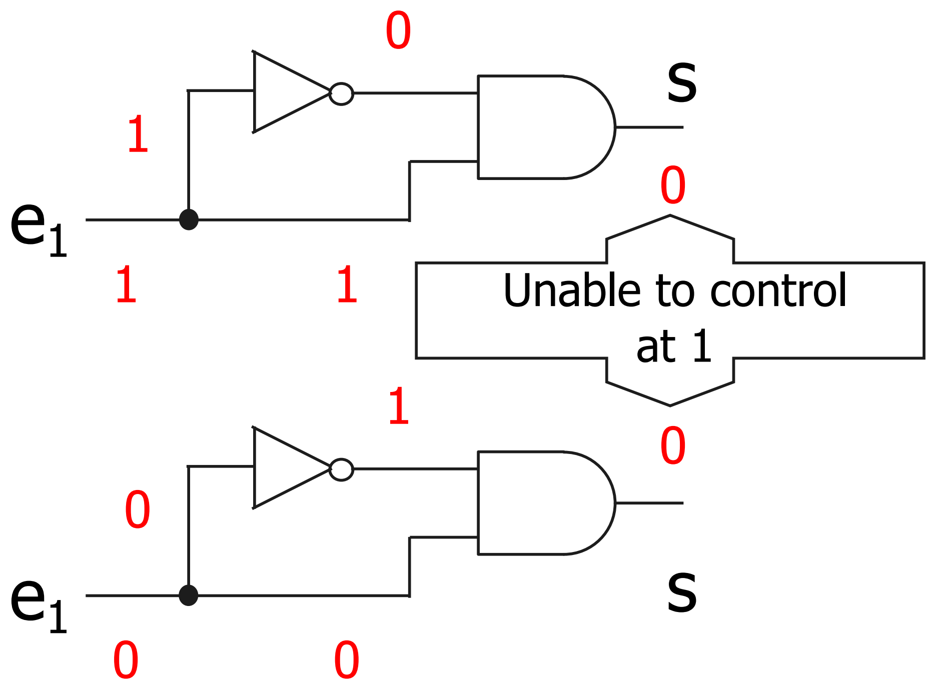
SCOAP Pros and Cons (1)

SCOAP CC1



e_1 = 1

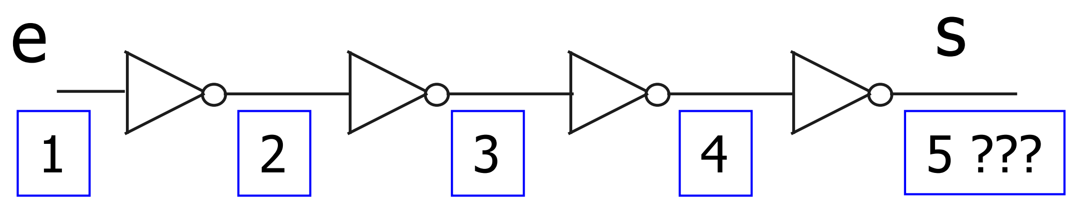
e_1 = 0



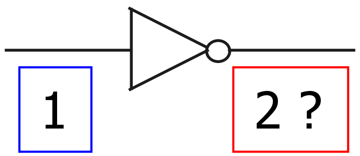
SCOAP Pros and Cons (2)

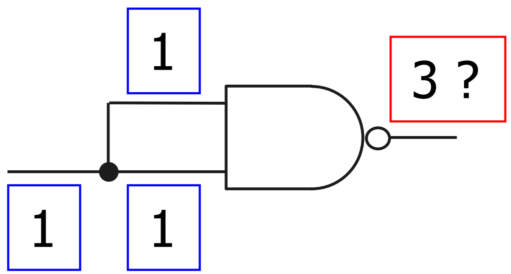
SCOAP CC1

SCOAP CC0



However, it is as easy to control s as e





The function structure influences the results

COP Principle

- Based on a structural analysis of the DUT
- COP Measurements → Probabilities
  - Proportion of input vectors that can control and observe a net
- For combinatorial parts of DUT only

COP Measurements

- Five measurements associated to each net:
  - C1(N) = Controllability at 1 of net N
    = (# input vectors generating a 1 on N) / 2^n (n = # bits)
  - C0(N) = Controllability at 0 of net N
    = (# input vectors generating a 0 on N) / 2^n
    = 1 - C1(N)
  - O(N) = Observability of net N
    = (# of 1 and 0 on N observable on at least one PO) / 2^n
  - O1(N) = Observability at 1 of net N
    = Detection probability of the Sa0 on net N
    = C1(N).O(N)
  - O0(N) = Observability at 0 of net N
    = Detection probability of the Sa1 on net N
    = C0(N).O(N)

Classical Logic Gates

- AND
  - C1(s) = C1(e1) . C1(e2)
  - O(e1) = C1(e2) . O(s)
- OR
  - C1(s) = 1 - (1-C1(e1)).(1-C1(e2))
  - O(e1) = (1-C1(e2)) . O(s)
- NOT
  - C1(s) = 1 - C1(e)
  - O(e) = O(s)

COP Process

- Net initialisation:
  - C1(PIs) = 0,5
  - O(POs) = 1
- Phase 1:
  - Controllability computation form PIs to POs
- Phase 2:
  - Observability computation form POs to PIs

Conclusion

- Controllability and Observability measurements
  - Can represent a “bad” image of the testability

- Possible misinterpretation **BUT** still widely used in industry
  - To help the test vector generation process (decision making)
  - To guide the Design-for-Test steps (test point insertion, …)