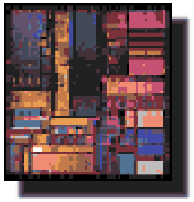
ECE 595Z
Digital VLSI Design Automation

Module 6 (Lectures 21-24): Timing Analysis and Optimization

Lecture 23



Anand Raghunathan

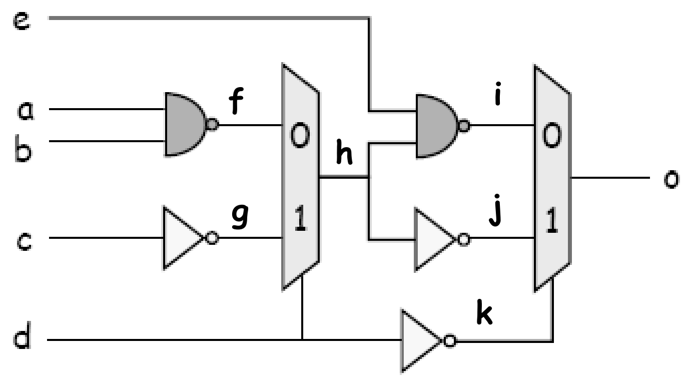
MSEE 318

raghunathan@purdue.edu

The False Path Problem

- Problem: Topological timing analysis may be pessimistic!
  - Ignores functionality of the nodes in the circuit
- Some paths can never be responsible for determining the delay of a circuit
  - Called **"false" paths**
- A path is false if no sequence of input vectors can result in an event propagating along it

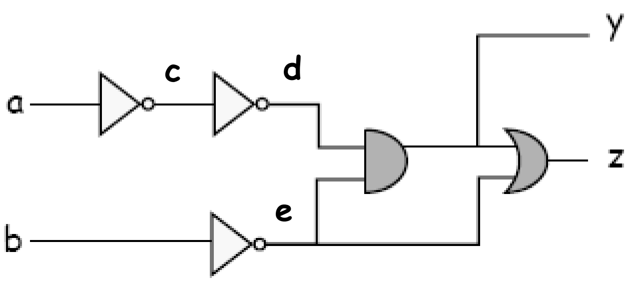
False Path Examples



Delay(NAND2) = 2, Delay(INV)= 1,
Delay(MUX) = 2

Longest Path:

False?



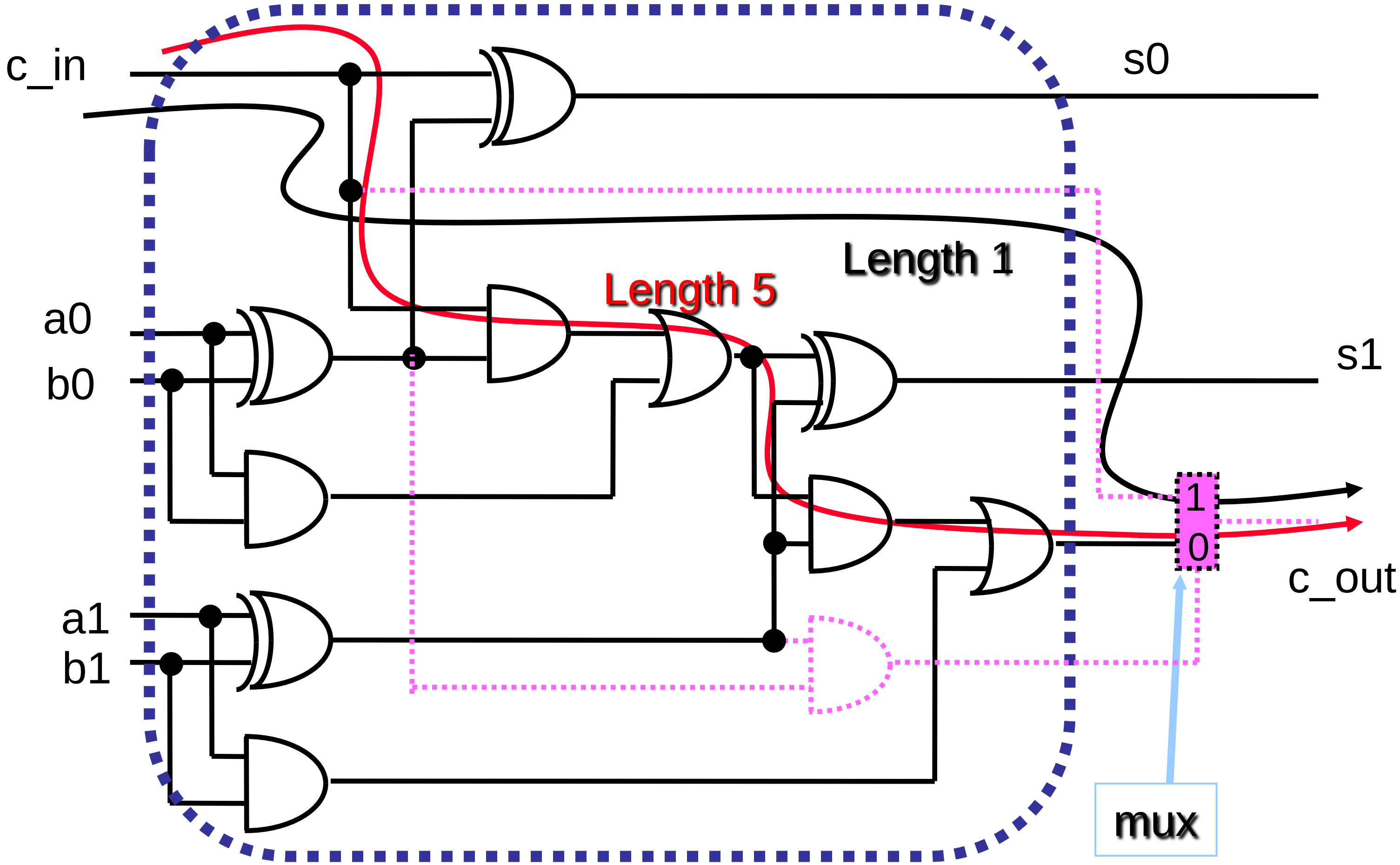
Delay(AND) = 2, Delay(INV)= 1,
Delay(OR) = 2

Longest Path:

False?

False Path: Real Example

- 2-bit carry-bypass adder



Functional Timing Analysis

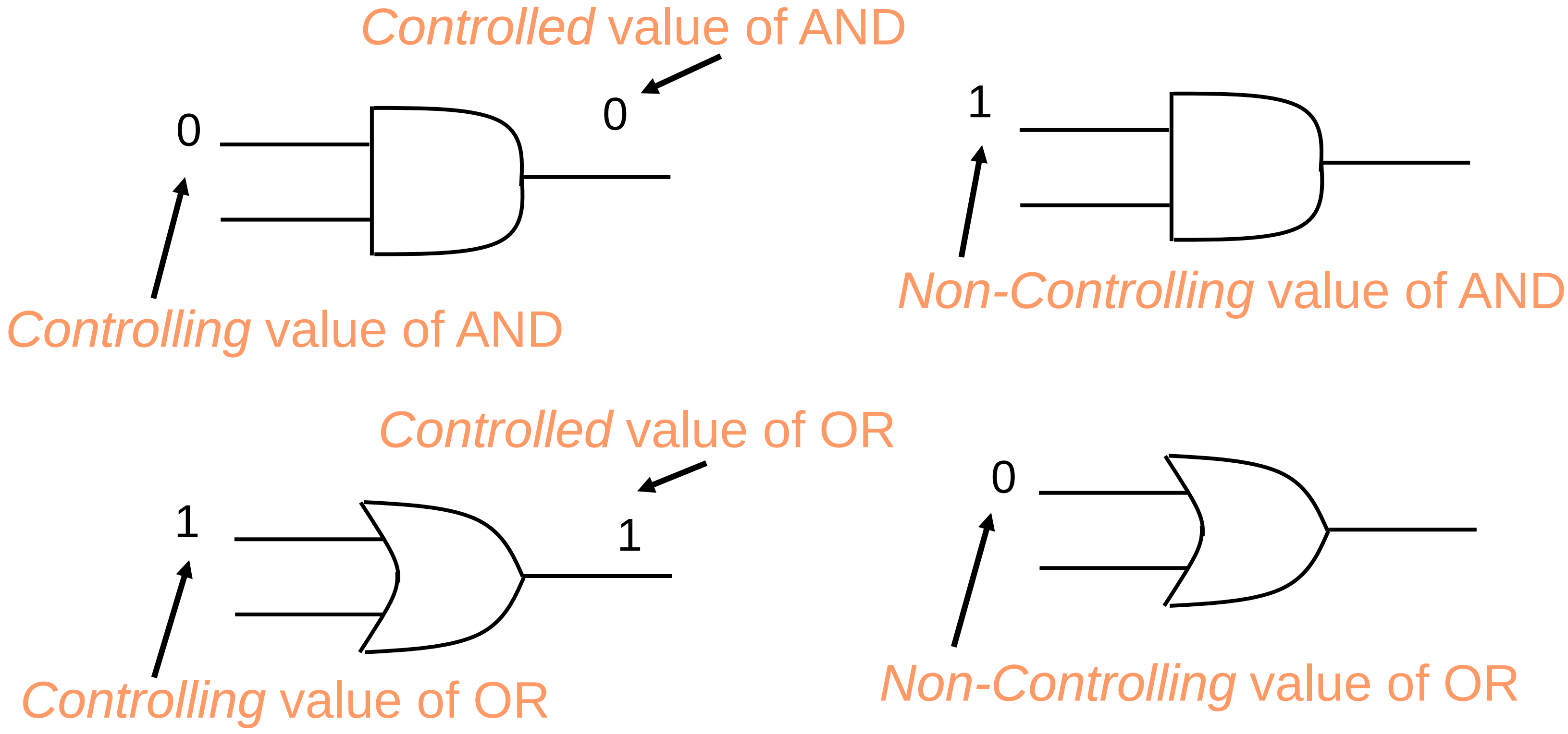
- If all topological longest paths are false, the delay estimate produced by topological analysis is an **overestimate**
- **Goal of functional timing analysis**: Determine the delay of a circuit considering **only true paths**
  - False path aware timing analysis
- Delay underestimation is **unacceptable**
  - Can lead to overlooking a timing violation
- Delay overestimation is **undesirable**
- Topological timing analysis can produce overestimates, but will never give an underestimate

Path Sensitization Criteria

- Need to formally define conditions under which a path is true (or sensitized)
- Much trickier than you may think!
- We will look at two sensitization criteria
  - Static sensitization
  - Static Co-sensitization

Background: Controlling and Non-controlling values

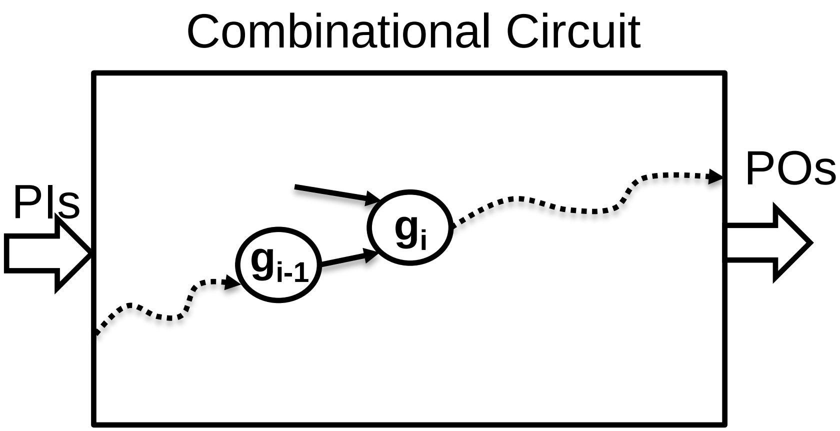
- Controlling value: Value at a gate input that is sufficient to determine gate output



Question: What is the controlling value for an XOR gate?

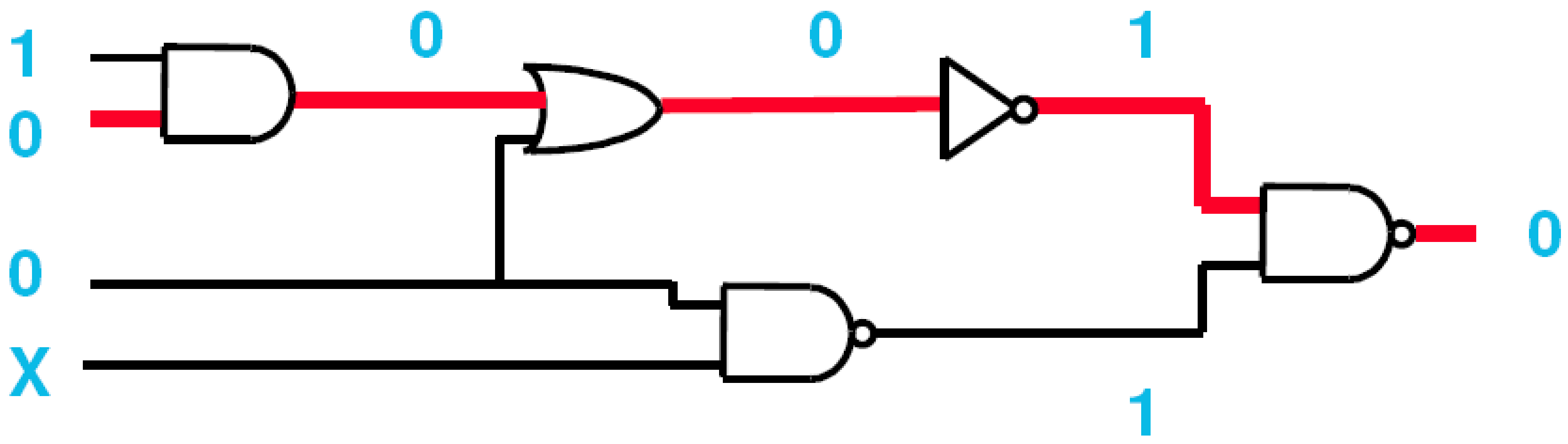
Static Sensitization of Paths

- A path in a combinational circuit is a sequence of vertices and edges (gates and wires) from a primary input to a primary output
- Each gate on the path has one **path input** and (zero or more) **side inputs**
- A path is **statically sensitizable** if there exists an input vector that sets all the side inputs to gates on the path to non-controlling values
  - NOTE: This criterion is independent of gate delays



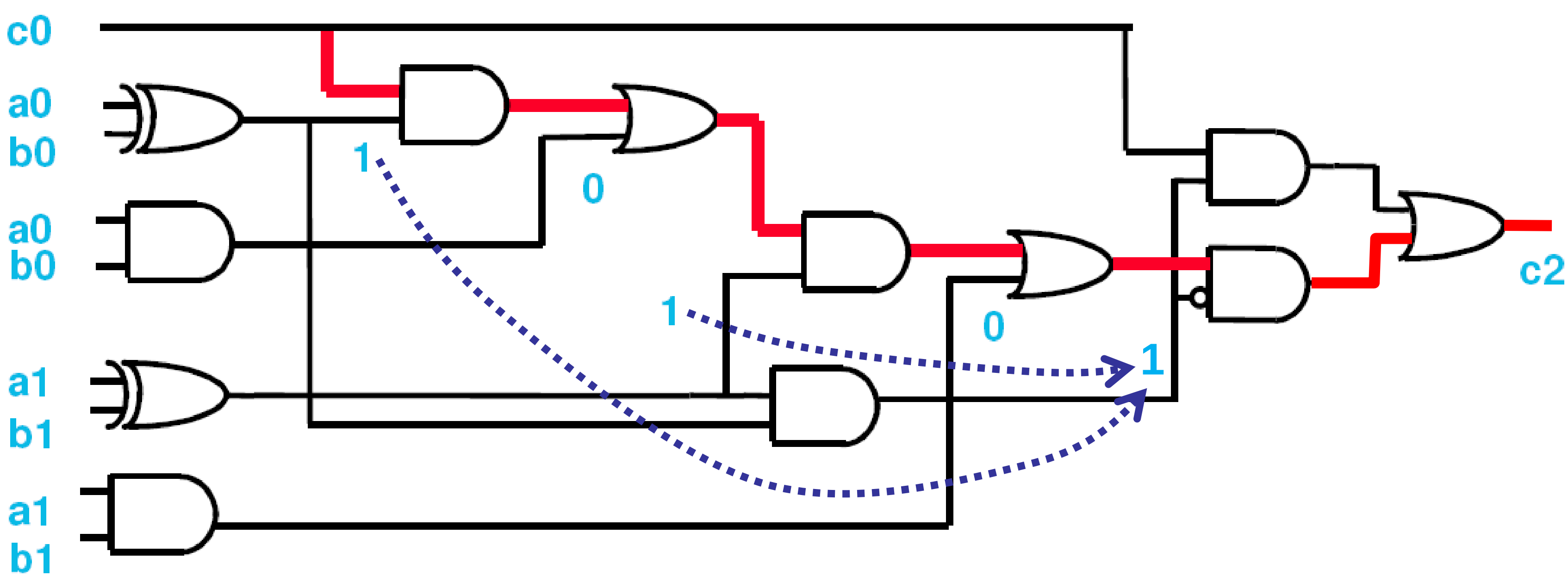
Static Sensitization: Example

- A statically sensitizable path



Static Sensitization: Example

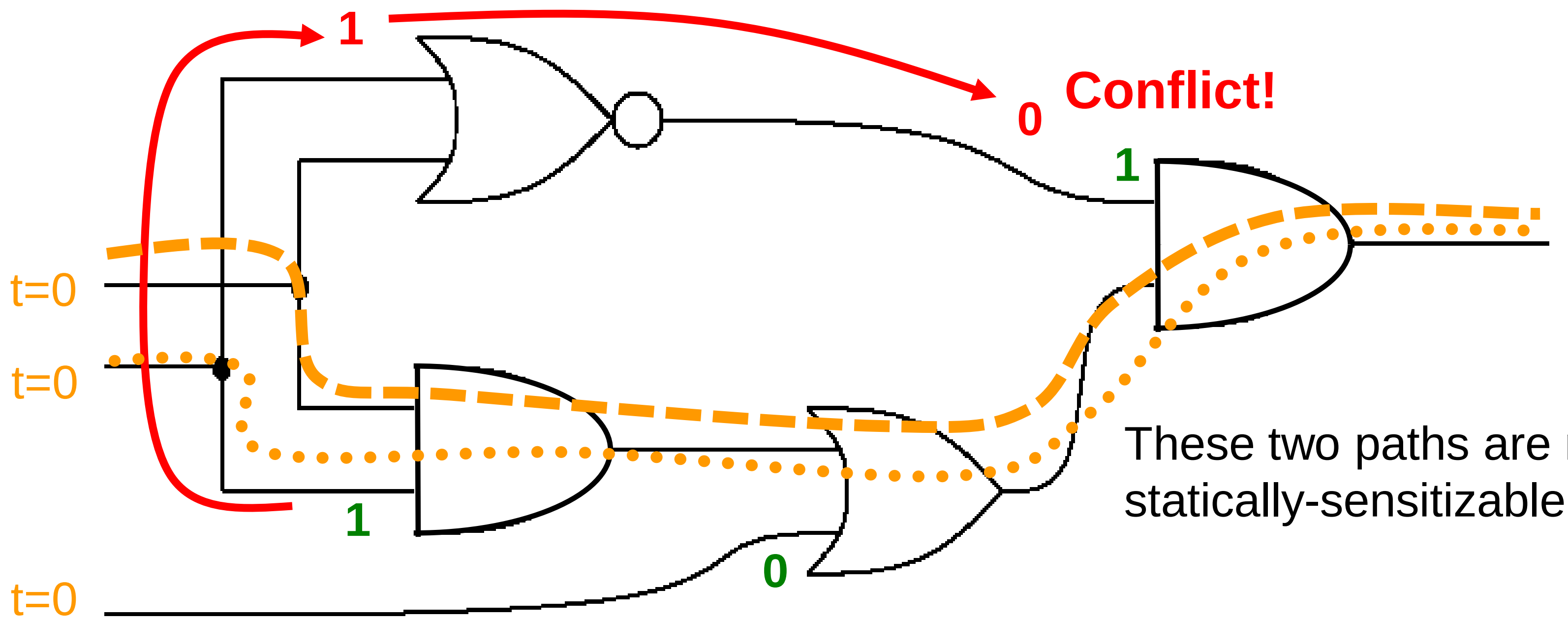
- A path that cannot be statically sensitized



Static Sensitization: Example

- What is the relationship between static sensitization and delay?

Example: Assume unit delay

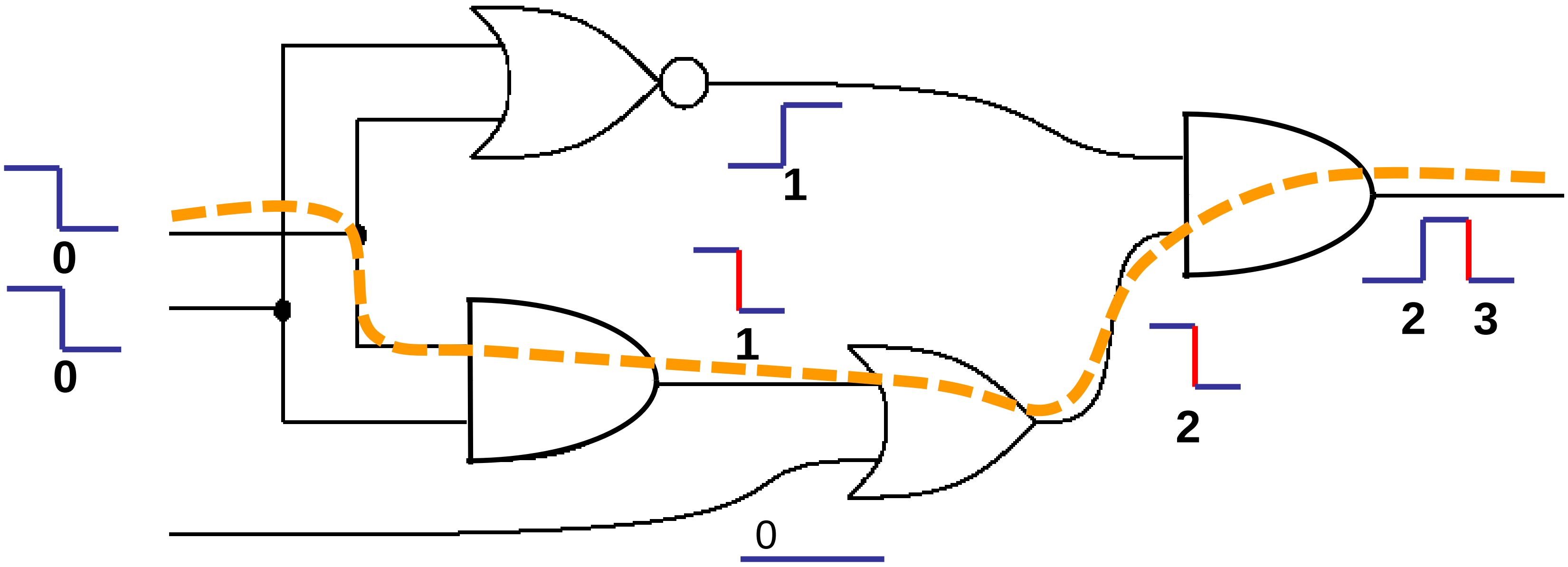


The longest statically sensitizable path is of length 2

Question: If inputs are applied at t = 0, does the output always stabilize by t = 2?

Static Sensitization: Example

- What is the relationship between static sensitization and delay?



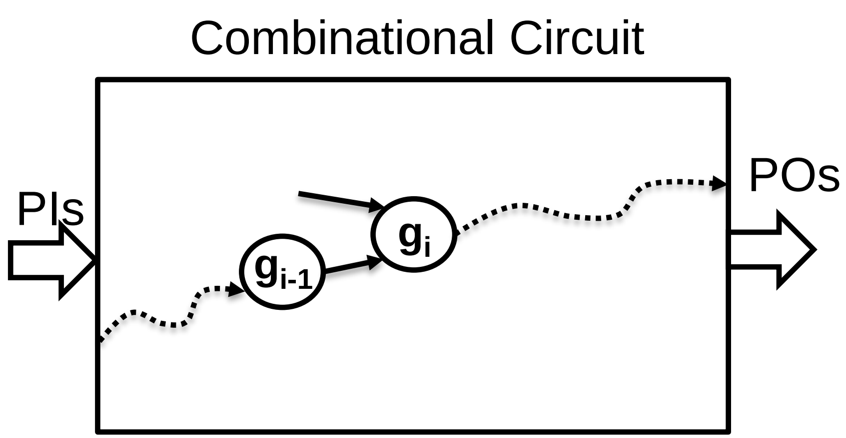
The longest statically sensitizable path is of length 2
Output stabilizes only at t = 3 !

Inadequacy of Static Sensitization

- Longest statically sensitizable path is an underestimate of circuit delay
- What is wrong?
  - The idea of forcing non-controlling values to side inputs is okay, but … timing was ignored
  - The same signal can have a controlling value at one time and a non-controlling value at another time.
- Lesson: Timing and functionality are intricately intertwined

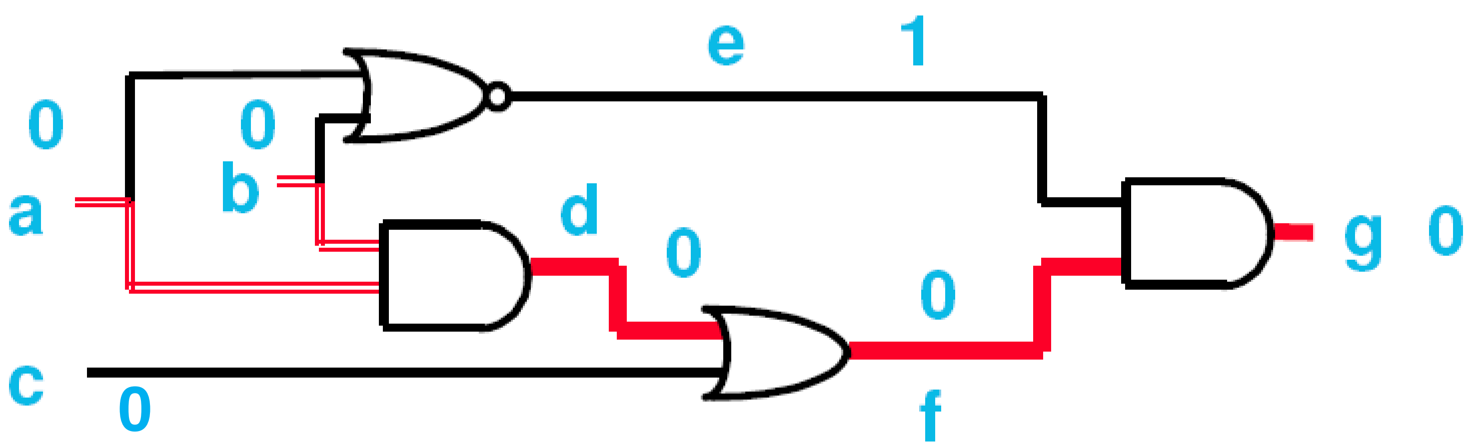
Static Co-Sensitization

- An input vector **statically co-sensitizes** a path $\{g_1 \rightarrow g_2 \rightarrow \ldots \rightarrow g_m\}$ if for each gate g_i whose output has a controlled value, the path input g_{i-1} has a controlling value
  - Difference from static sensitization: If path input is controlling, side inputs can also be controlling
  - NOTE: This criterion is still independent of gate delays
- A path is statically co-sensitizable if there exists an input vector that statically co-sensitizes it



Co-sensitization: Example

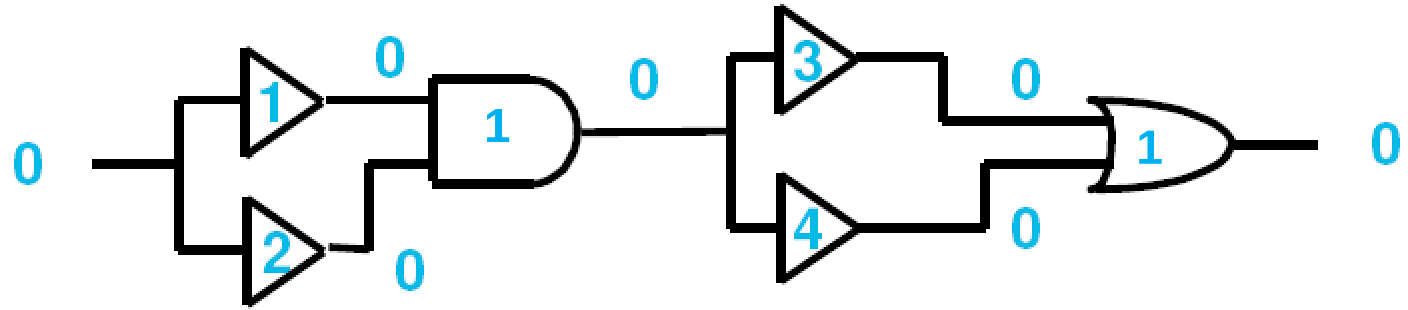
- For each gate with a controlled output value, path input must be a controlling value



Paths a→d→f→g and b→d→f→g are co-sensitized by the input vector a=0,b=0,c=0

Co-sensitization: Example

- What is the relationship between static co-sensitization and delay?

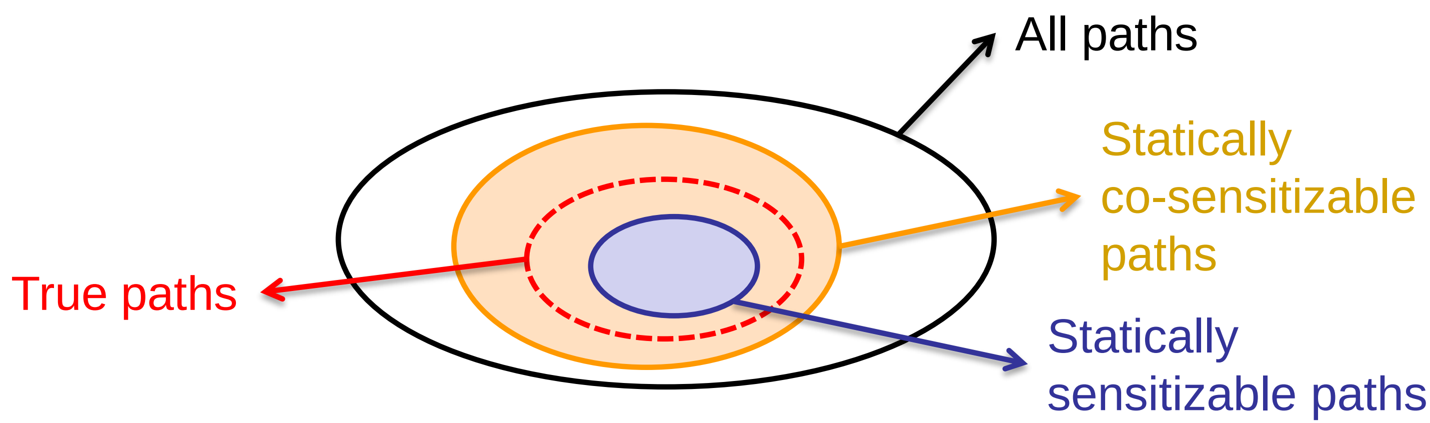


Longest co-sensitizable path:

Circuit delay:

Sensitization Criteria and Circuit Delay

- Static sensitization is a **sufficient** condition for a path to be true
  - The longest statically sensitizable path is a lower bound on the maximum delay of a circuit
- Static co-sensitization is a **necessary** condition for a path to be true
  - The longest statically co-sensitizable path is an upper bound on the maximum delay of a circuit

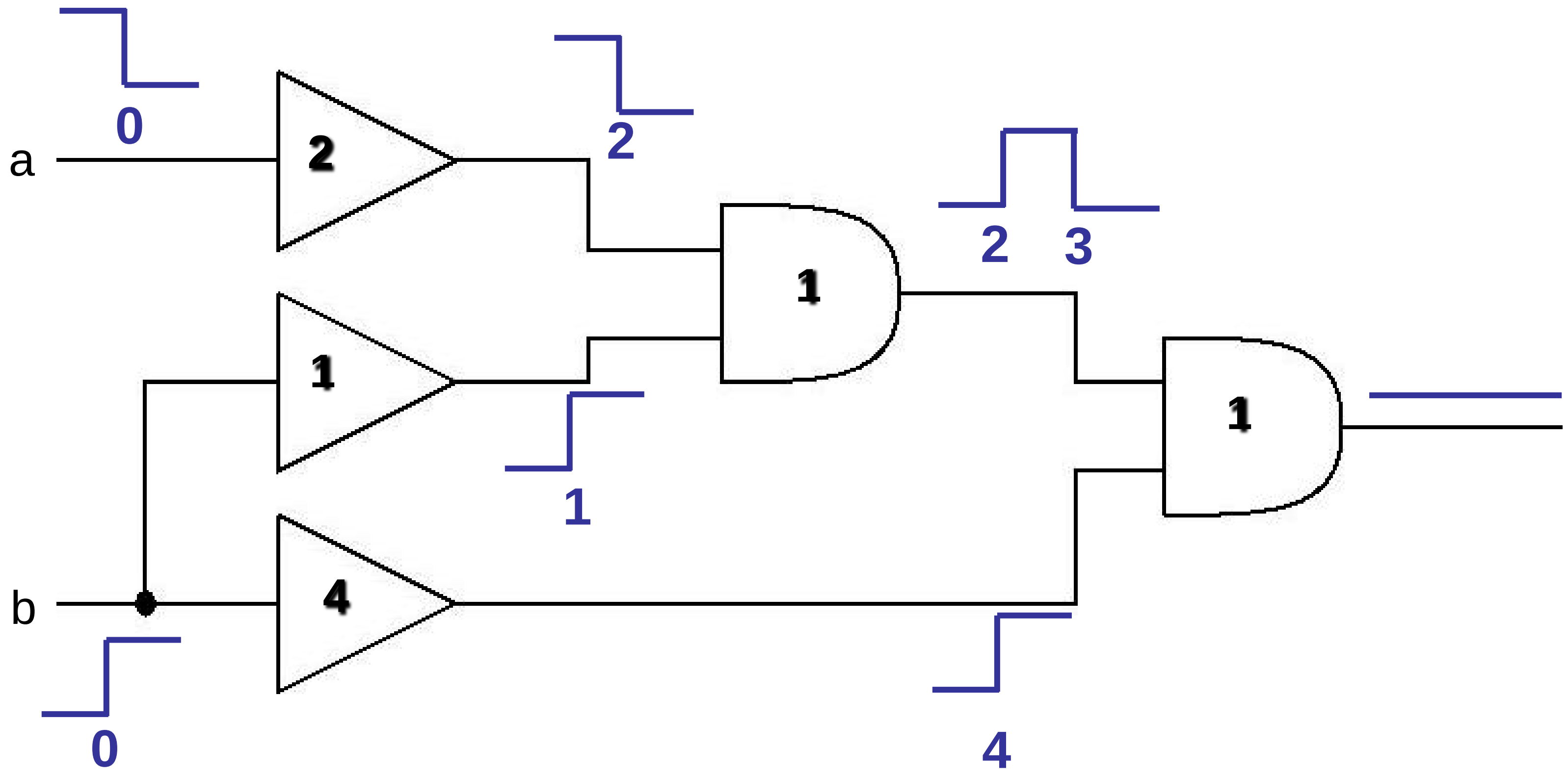


Question: Can we specify precise conditions to identify true paths?

Transition vs. Floating Mode Delay

- **Transition mode delay**: Delay of a combinational circuit under a pair of input vectors $<v_1, v_2>$
  - Search space to prove a path true/false: 2^{2n} (for a circuit with n inputs)
- **Floating mode delay**: Only look at a single vector
  - Assume that all signals in the circuit are "floating" before application of the vector
  - Make conservative assumptions
  - Reduces search space to 2^n

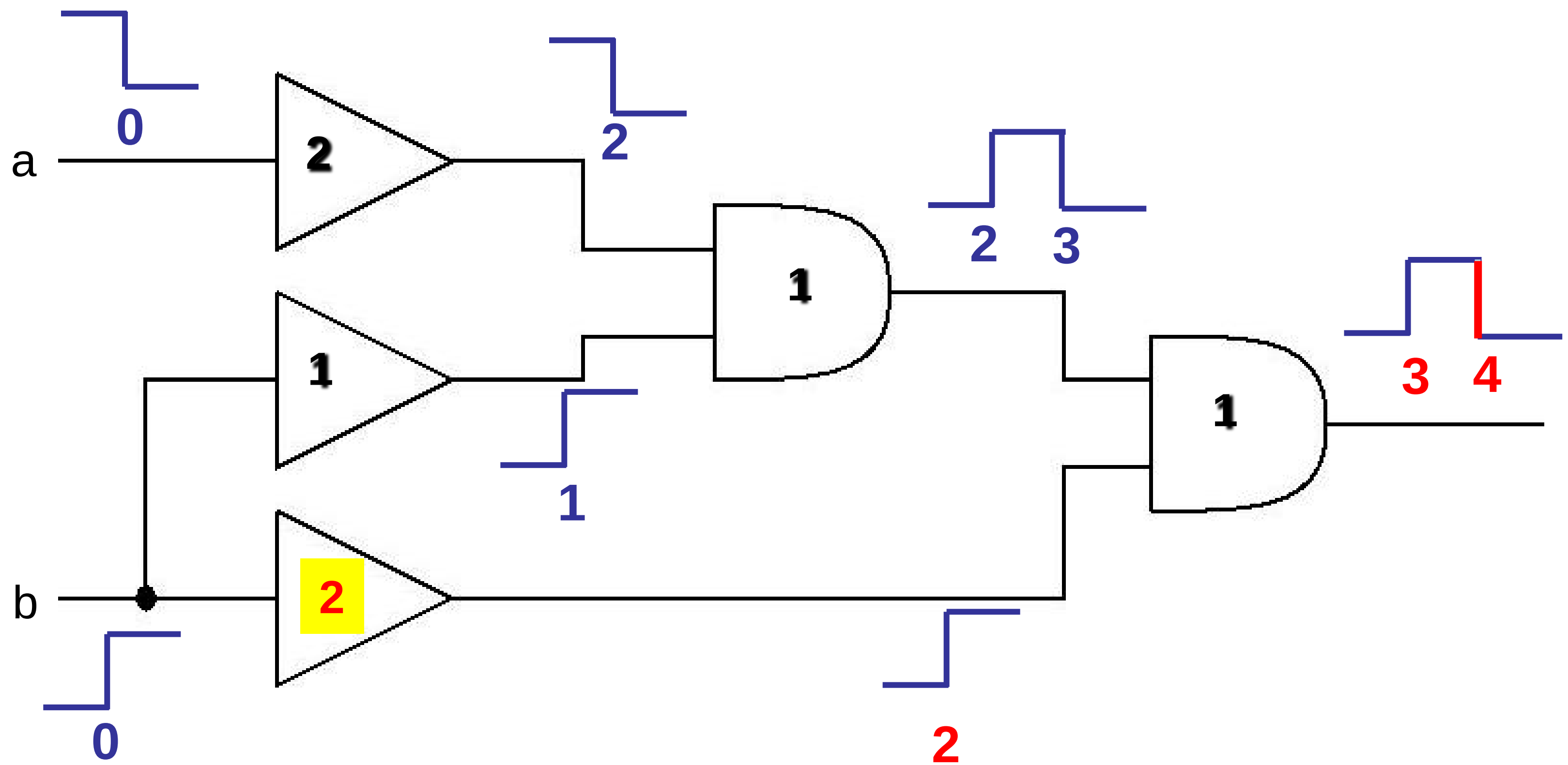
Transition Mode Delay Estimation



Output stabilizes at time t=0 under input vector pair a=1,b=0 → a=0,b=1

Transition Mode Delay Estimation

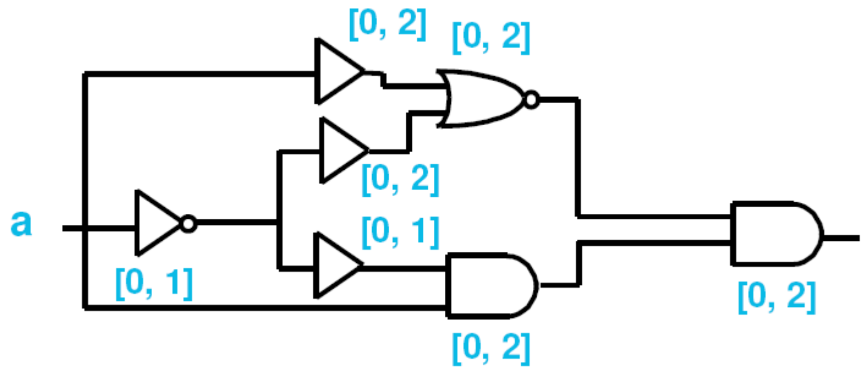
- What happens if a gate in the circuit is made faster?



Output stabilizes at time t=4 under input vector pair a=1,b=0 → a=0,b=1

Transition Mode Delay Estimation Pitfalls

- Circuit delay could increase if the delay of a gate decreases!
- In practice, due to uncertainty of modeling / variations, gate delays are only bounds
- We are implicitly analyzing a family of circuits where gate delays are within the bounds
  - We want timing analysis to report the critical path of the slowest circuit in the family



Monotone Speedup Property

Definition: For any circuit C and delay estimation procedure delay_estimate(), if

- C' is obtained from C by reducing some gate delays implies that
- delay_estimate(C') ≤ delay_estimate(C),

then delay_estimate satisfies the *Monotone Speedup* property

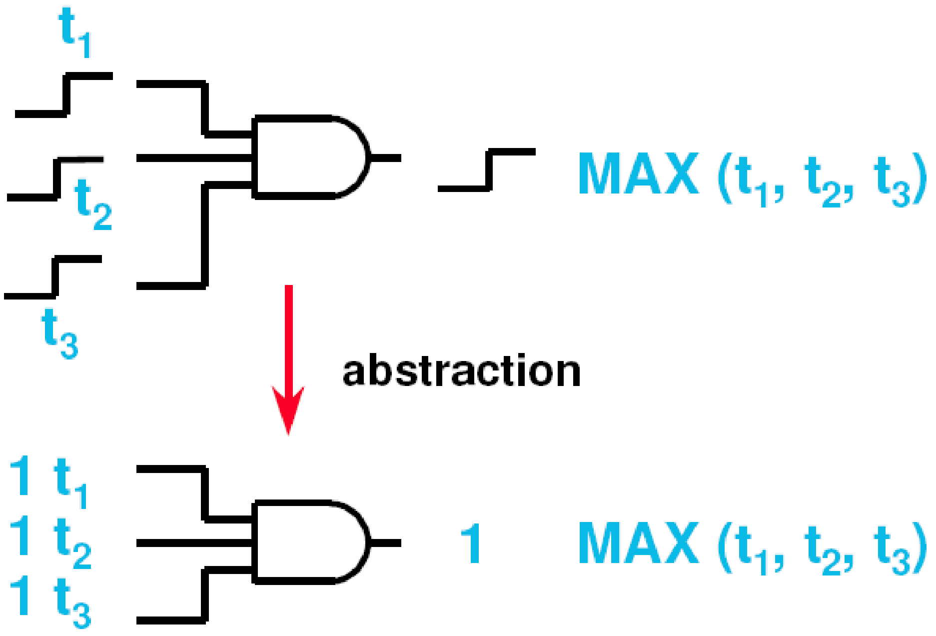
Timing simulation and Transition Mode delay analysis do not satisfy the monotone speedup property!

Floating mode analysis **does** satisfy monotone speedup.

Accurate Sensitization Criteria for Floating Mode Timing Analysis

- Start off with co-sensitization but augment with timing information

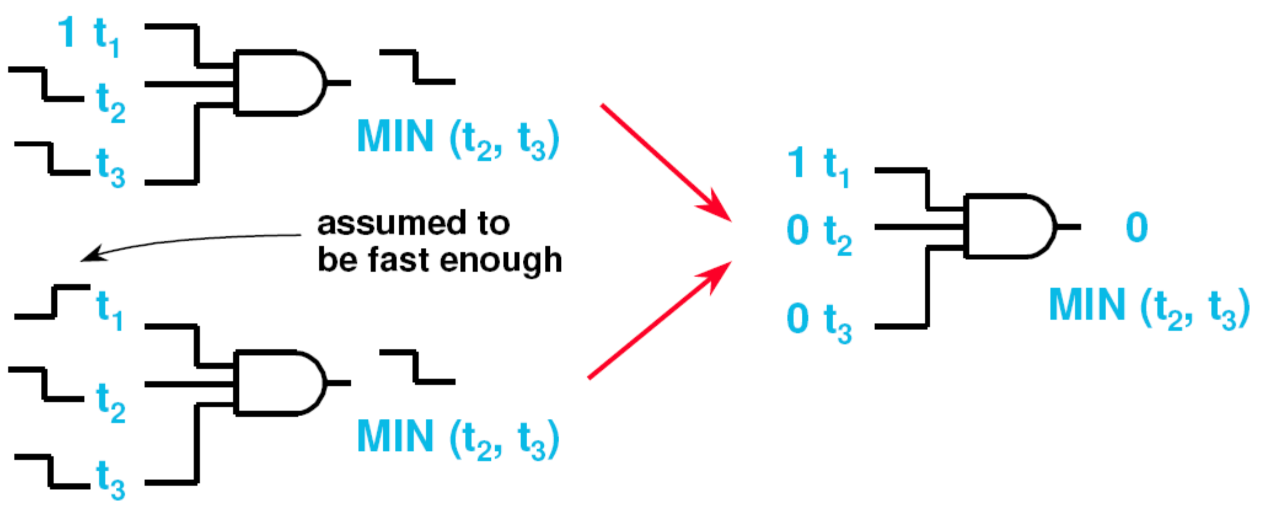
If a gate output has a non-controlled value, the time at which the output becomes stable is determined by the slowest of the non-controlling inputs



Accurate Sensitization Criteria for Floating Mode Timing Analysis

- Start off with co-sensitization but augment with timing information

If a gate output has a controlled value, the time at which the output becomes stable is determined by the earliest of the controlling inputs

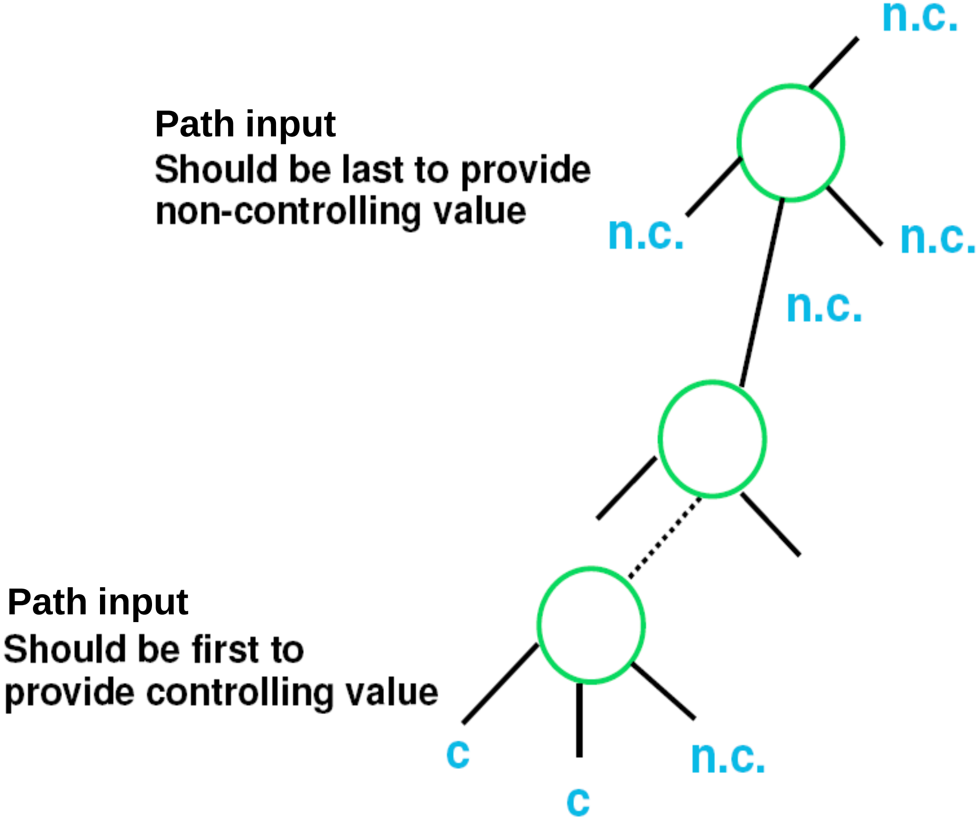


Accurate Sensitization Criteria for Floating Mode Timing Analysis

- **Necessary and sufficient** conditions for a path to be true under the floating mode

Condition #1: If a gate output has a non-controlled value, the path input provides the latest non-controlling value

Condition #2: If a gate output has a controlled value, the path input provides the earliest controlling value



Algorithms for Functional Timing Analysis

- Naïve algorithm
  1. Find longest topological path
  2. Check if path can be sensitized

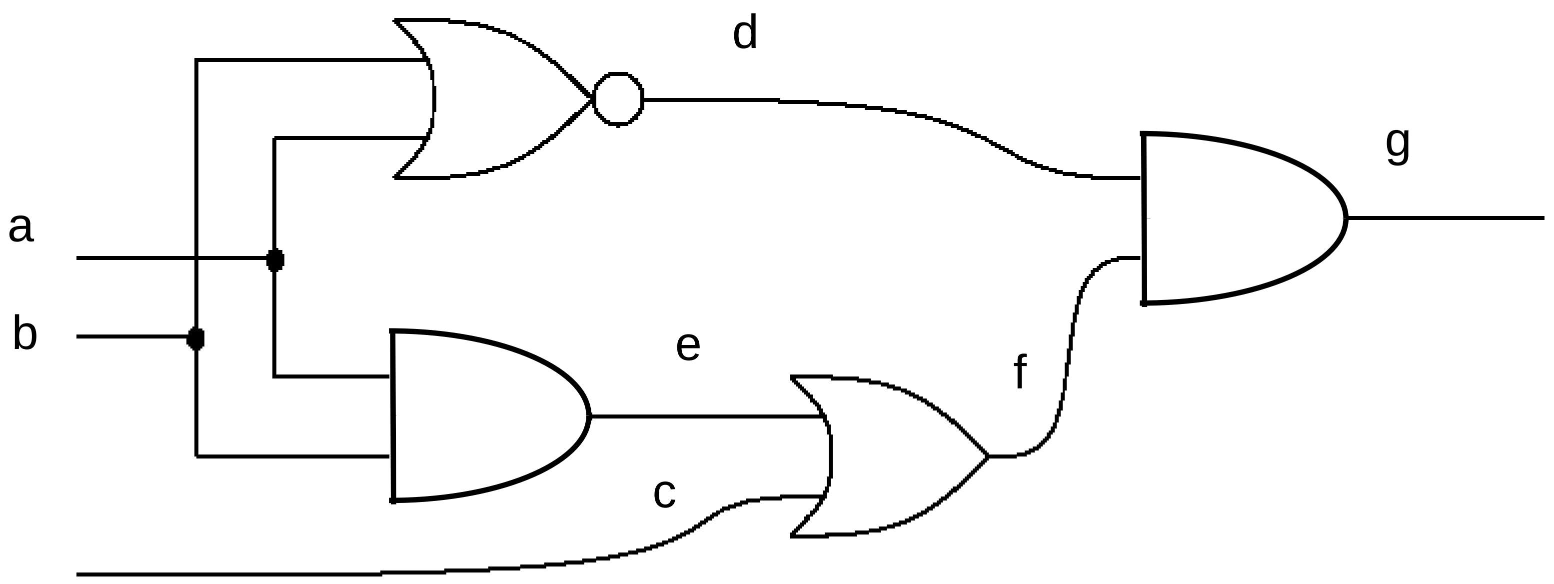
     Search problem (find input vector)
  3. If True, report path length as circuit delay and exit
  4. If False, Find next longest topological path and go to step 2

Problem: ______________________________

Algorithms for Functional Timing Analysis

- More efficient approach
  - Formulate a procedure that can check if the circuit has a True path with delay ≥ D
  - Perform a binary search on the interval $[0, D_{topological}]$
  - Avoids enumerating potentially exponential # of paths
- Two different techniques proposed to perform the above check
  - Timed D-calculus [Devadas,Keutzer,Malik – ICCAD 1991]
    - Based on well-known Automatic Test Pattern Generation Algorithm
  - SAT formulation [McGeer et al. – ICCAD 1991]

Algorithms for Functional Timing Analysis

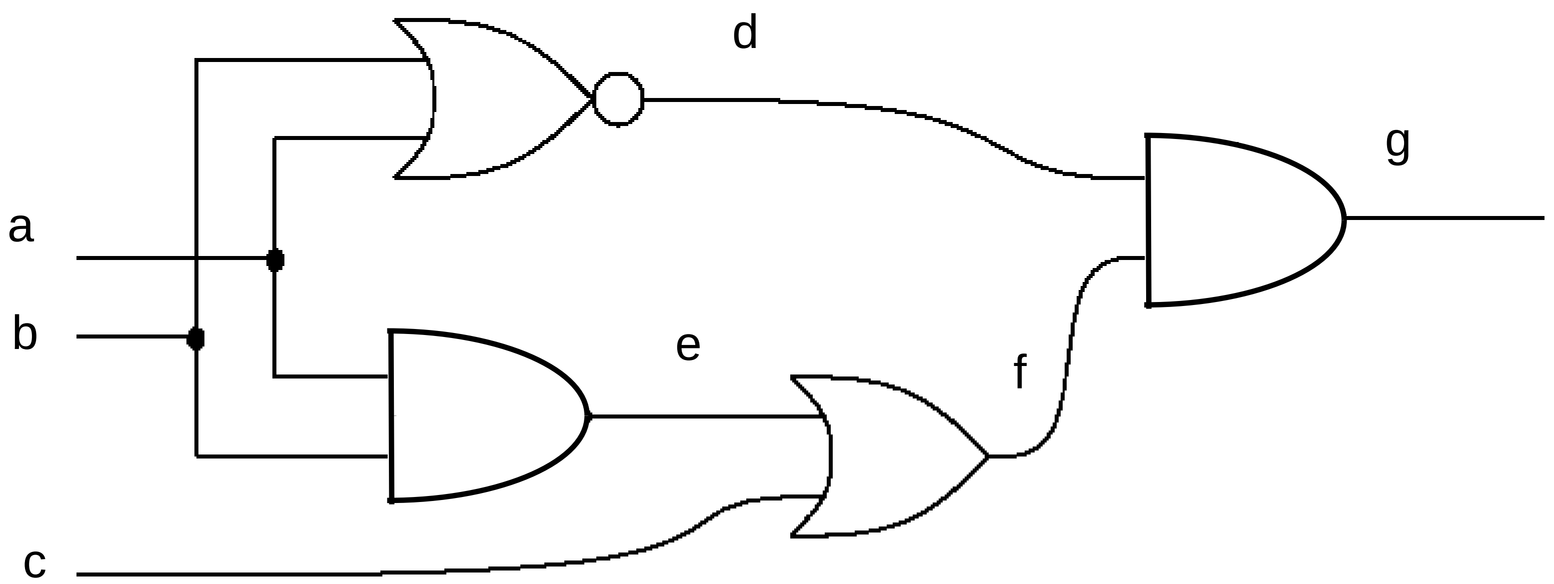


Assume all the PIs arrive at t = 0, all gate delays = 1

Can the output become stable at time t > 2?

Algorithms for Functional Timing Analysis

g(1,t=2) : the set of input vectors under which
g gets stable to value 1 no later than t = 2



$g(1,t=2) = d(1,t=1) \cap f(1,t=1)$

$= (a(0,t=0) \cap b(0,t=0)) \cap (c(1,t=0) \cup e(1,t=0))$

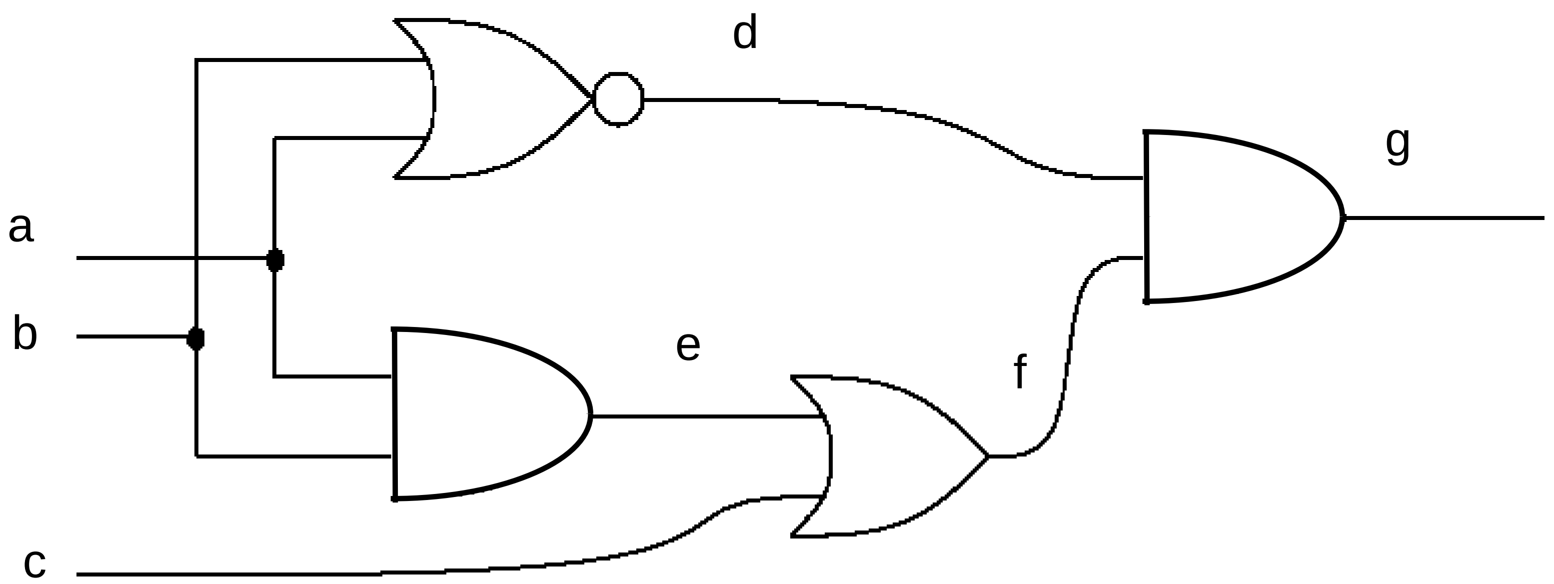
$= a'b'(c \cup \varnothing) = a'b'c$

$g(1,t=\infty) = \text{onset} = a'b'c$

Is g(1, t=2) XOR g satisfiable?
Answer: NO ⇒ output always stabilizes to 1 at t ≤ 2

Algorithms for Functional Timing Analysis

g(0,t=2) : the set of input vectors under which
g gets stable to value 0 no later than t =2



$g(0,t=2) = d(0,t=1) \cup f(0,t=1)$
$= (a(1,t=0) \cup b(1,t=0)) \cup (c(0,t=0) \cap e(0,t=0))$
$= (a+b) + (c' \cap \varnothing) = a+b$

$g(0,t=\infty)$ = offset = a+b+c'

Is g(0, t=2) XOR g' satisfiable?
Answer: YES! ⇒ output can stabilize to 0 at t > 2

Summary: Functional Timing Analysis

- Topological delay could be an overestimate when false paths are present
  - Quite common in practice
- Various "functional" sensitization criteria
  - Static sensitization, co-sensitization
- Transition vs. floating modes of delay computation
  - Desirable property: Monotone speedup
- Functional timing analysis without explicit path enumeration
  - Formulation based on Timed ATPG, SAT
- State-of-the-art in commercial tools:
  - Most tools have an option to allow designers to manually specify false paths
  - Advanced timing analysis tools (e.g., Synopsys PrimeTime™) can automatically identify false paths