

VLSI Design

Lecture 13: Interconnect Design

Shaahin Hessabi

Department of Computer Engineering
Sharif University of Technology

Adapted, with modifications, from lecture notes prepared
by the author (from Prentice Hall PTR)

Topics

- Interconnect design.
- Crosstalk.
- Power optimization.

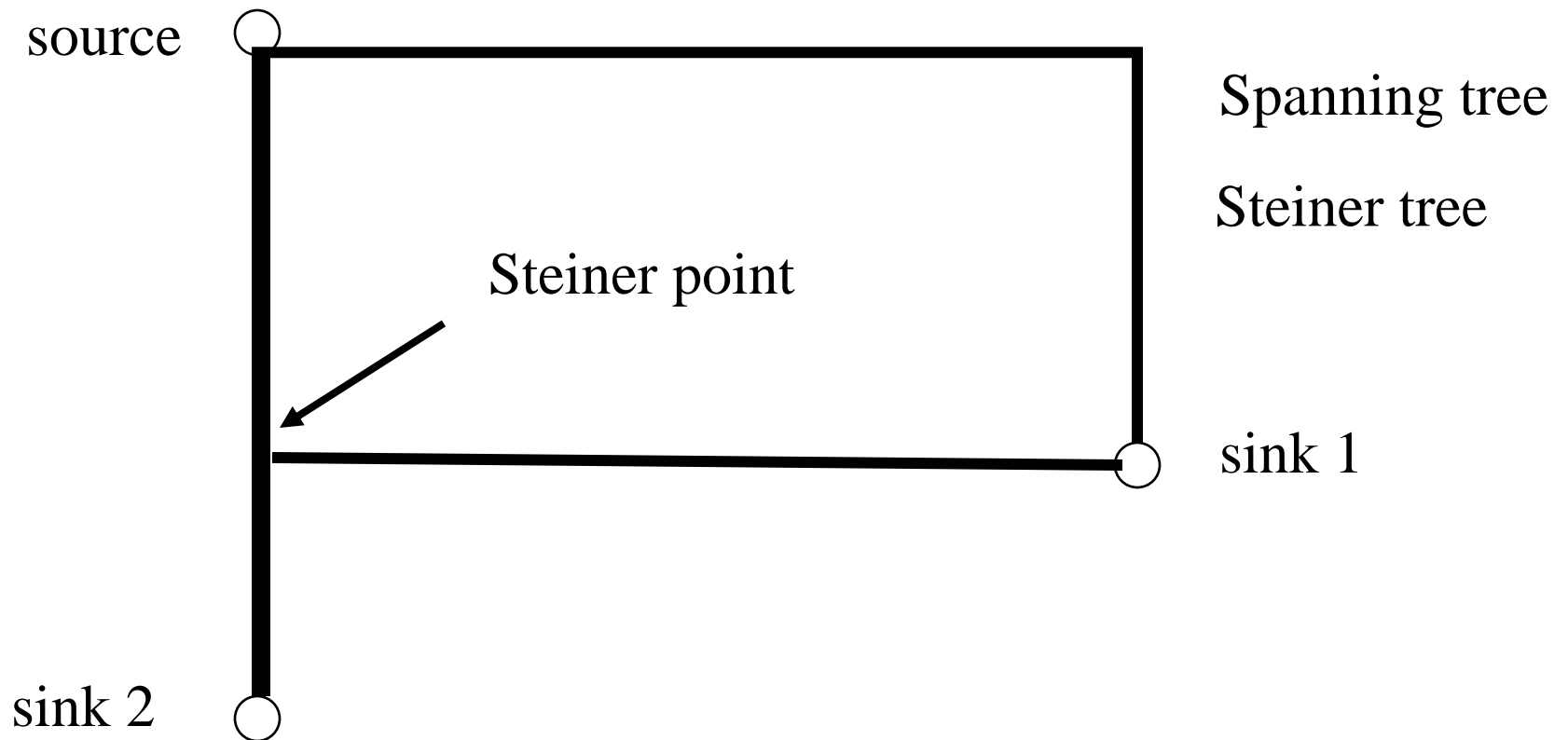
Interconnect

- Even assuming logic structure is fixed, we can:
 - change wire topology;
 - resize wires;
 - add buffers;
 - size transistors.

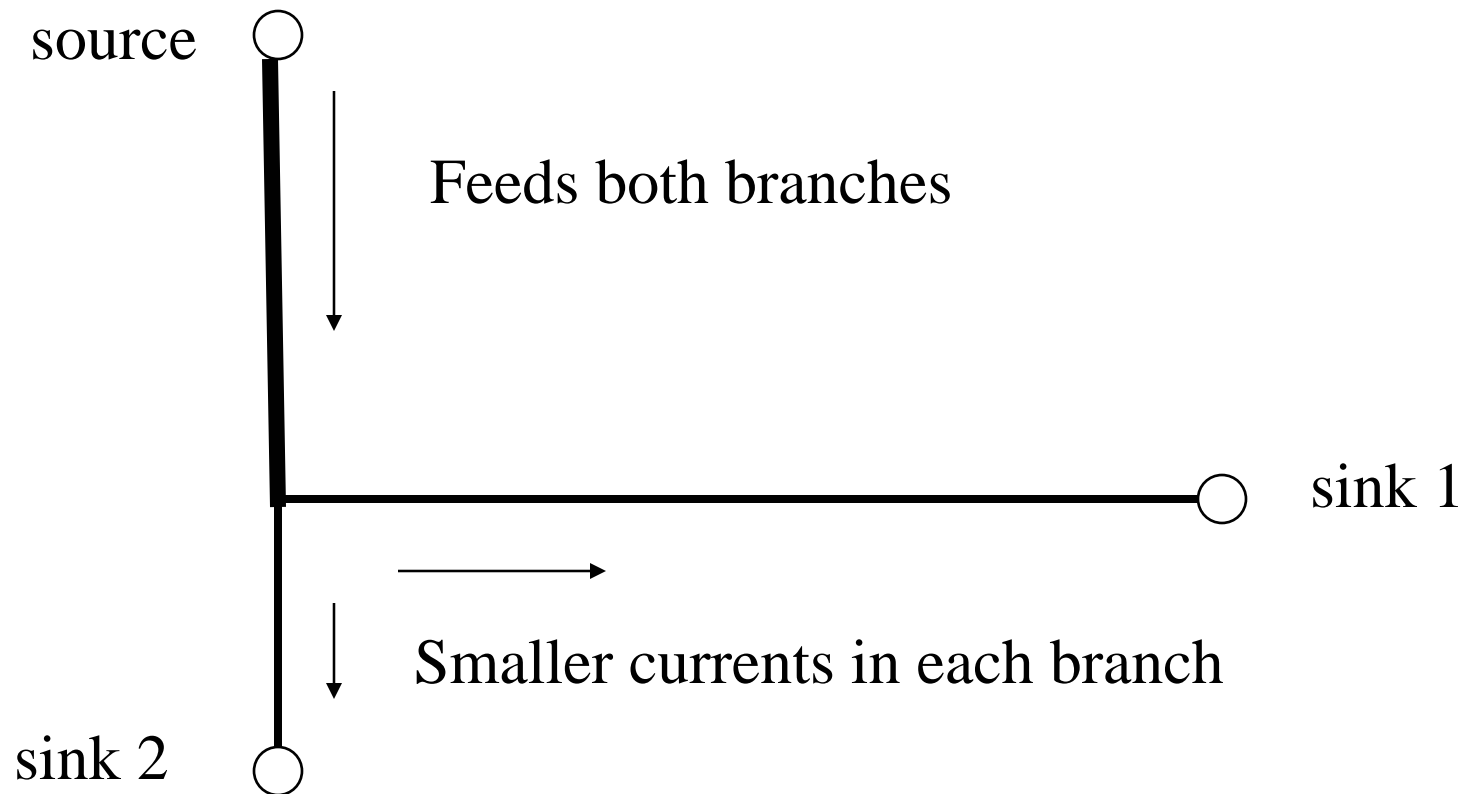
Multipoint nets

- Two-point nets are easy to design.
- Multipoint nets are harder:
 - How do we connect all the pins using two-point connections?

Styles of wiring trees



Sized Steiner tree



Buffer insertion in wiring trees

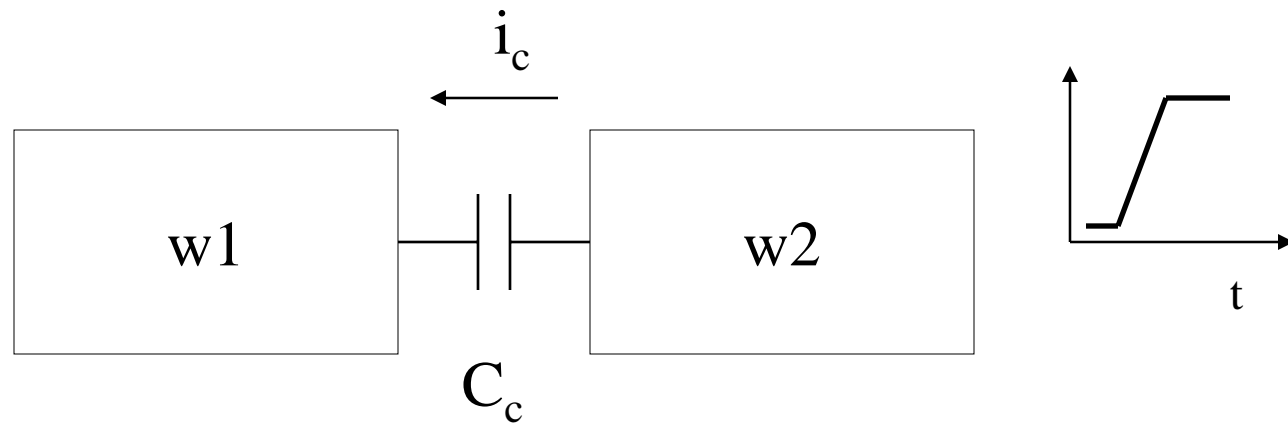
- More complex than placing buffers along a transmission line:
 - complex topology;
 - unbalanced trees;
 - differing timing requirements at the leaves.

Crosstalk

- Capacitive coupling introduces crosstalk.
- Crosstalk slows down signals to static gates, can cause hard errors in storage nodes.
- Crosstalk can be controlled by methodological and optimization techniques.

Coupling and crosstalk

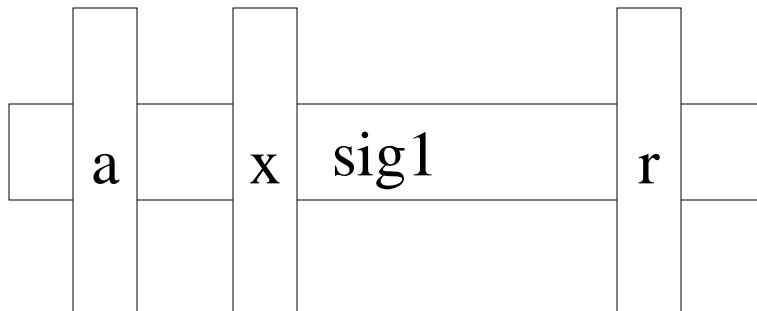
- Crosstalk current depends on capacitance, voltage ramp.



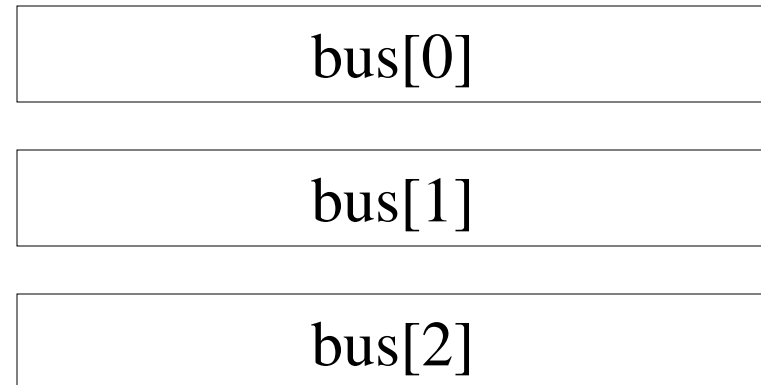
Crosstalk analysis

- Assume worst-case voltage swings, signal slopes.
- Measure coupling capacitance based on geometrical alignment/overlap.
- Some nodes are particularly sensitive to crosstalk:
 - dynamic;
 - asynchronous.

Coupling situations



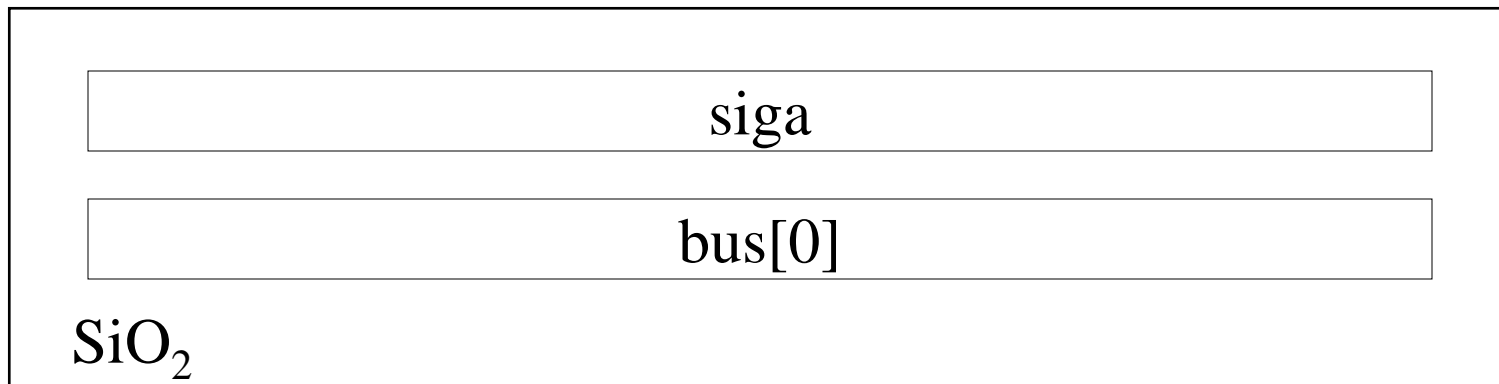
better



worse

Layer-to-layer coupling

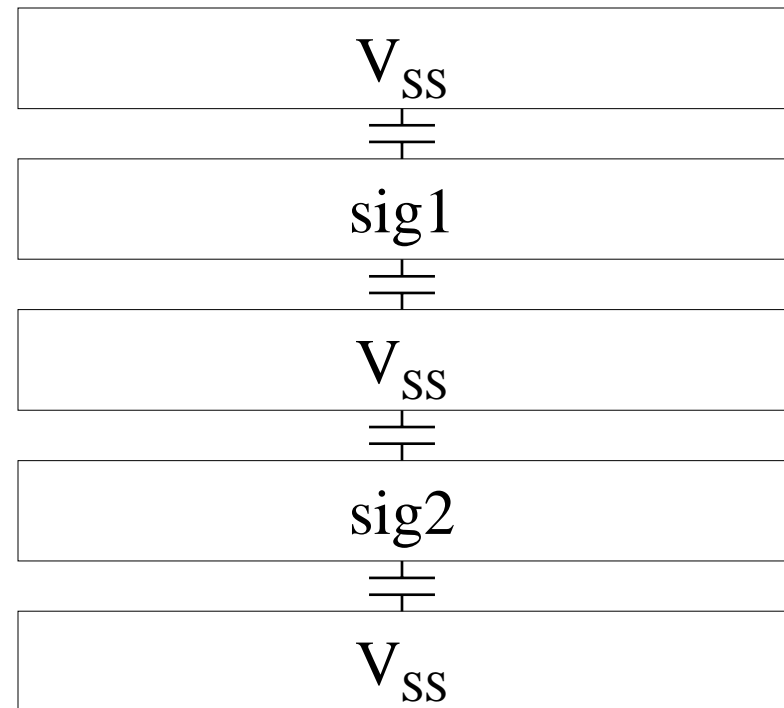
- Long parallel runs on adjacent layers are also bad.



Methodological solutions

- Add ground wires between signal wires:
 - coupling to V_{SS} , a stable signal, dominates;
 - can use V_{SS} to distribute power, so long as power line is relatively stable.
- Extreme case: add ground plane. Costs an entire layer.

Ground wires

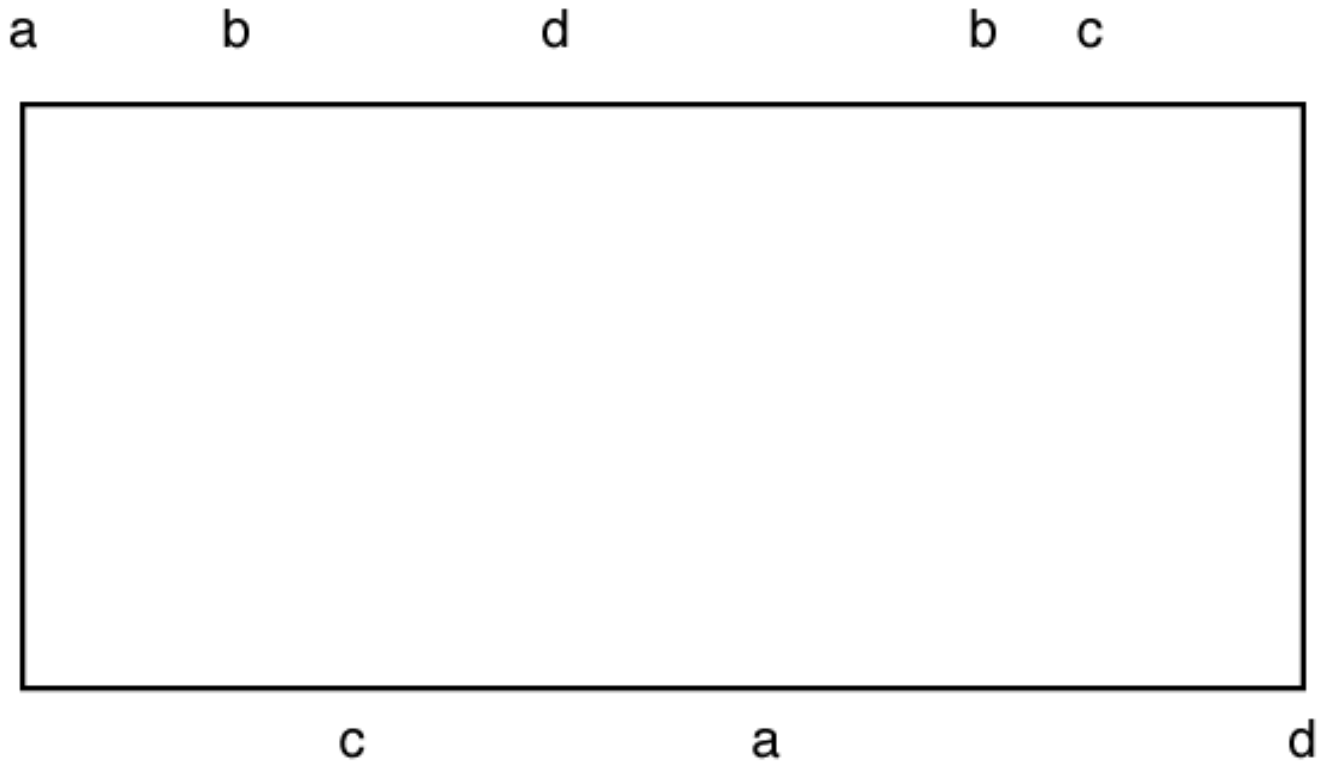


Crosstalk and signal routing

- Can route wires to minimize required adjacency regions.
- Take advantage of natural holes in routing areas to decouple signals.
- Minimizes need for ground signals.

Crosstalk routing example

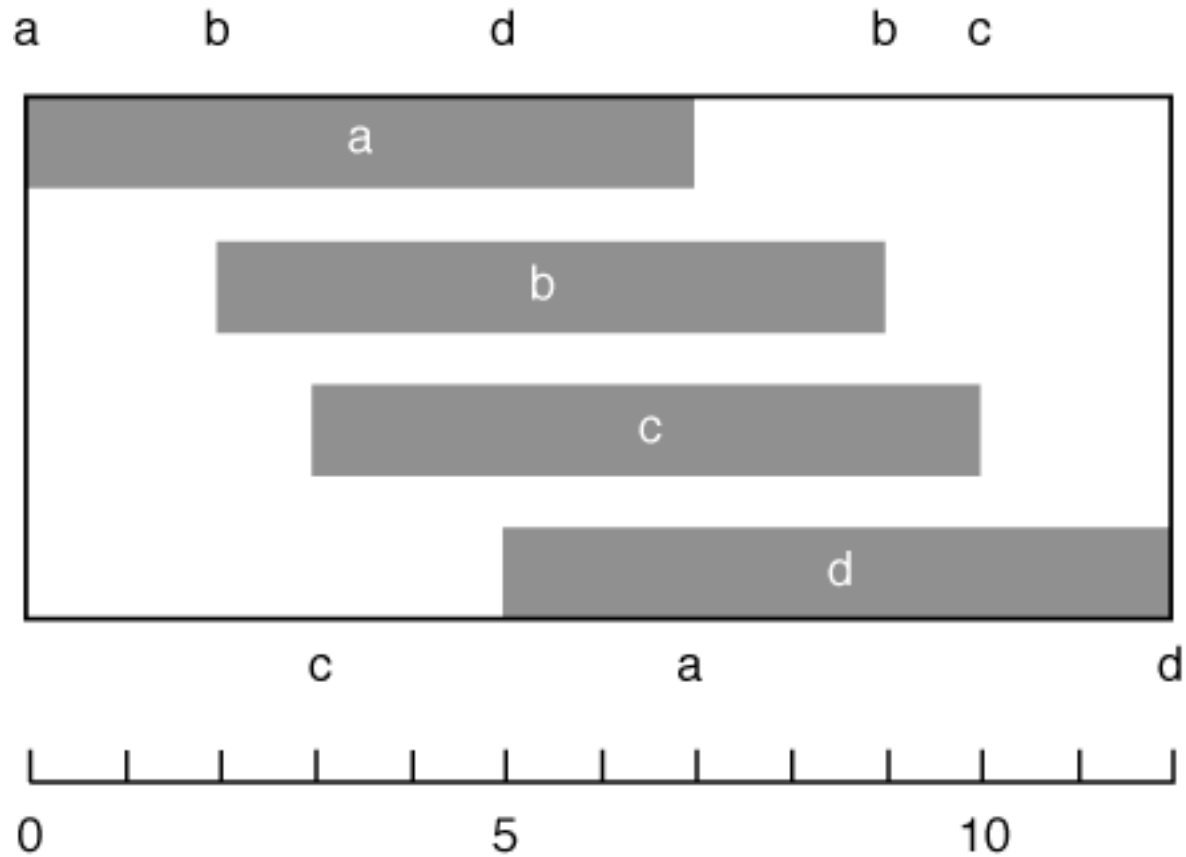
■ Channel:



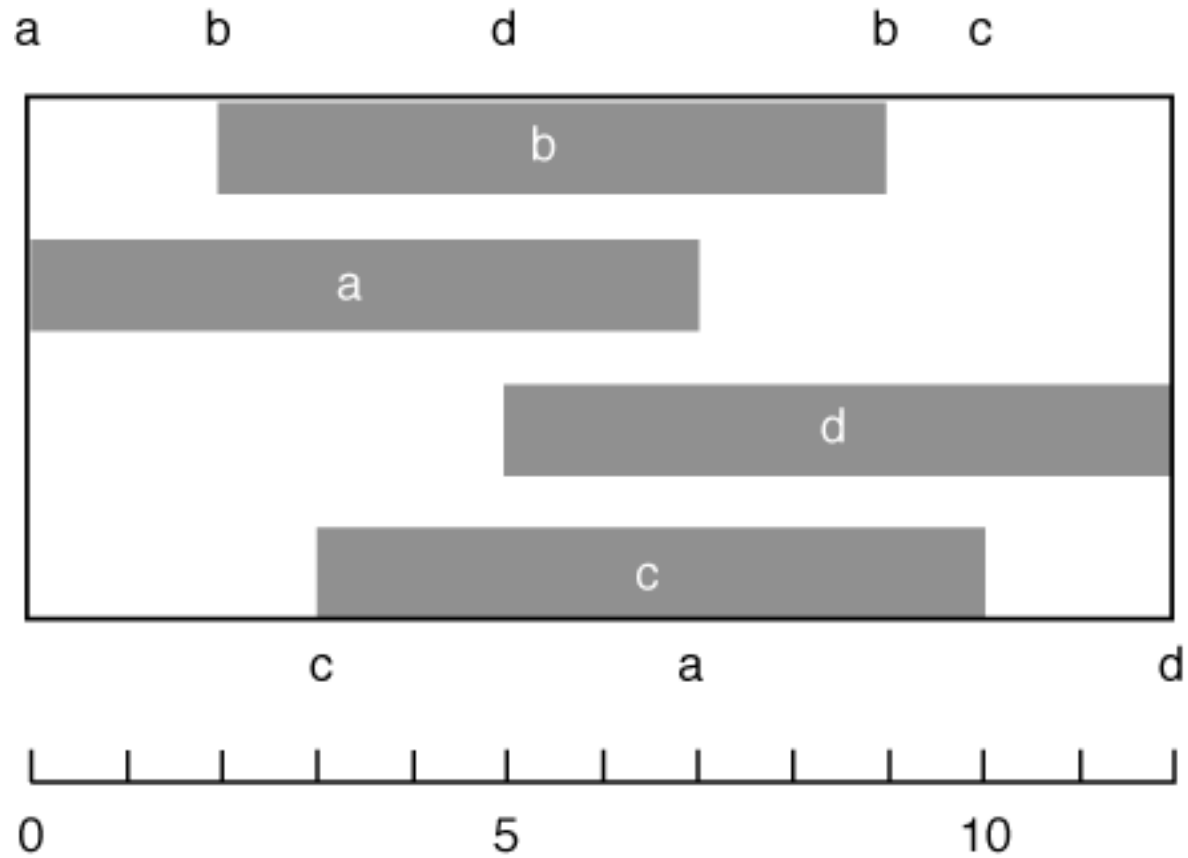
Assumptions

- Take into account coupling only to wires in adjacent tracks.
- Ignore coupling of vertical wires.
- Assume that coupling/crosstalk is proportional to adjacency length.

Bad routing



Good routing



Crosstalk analysis

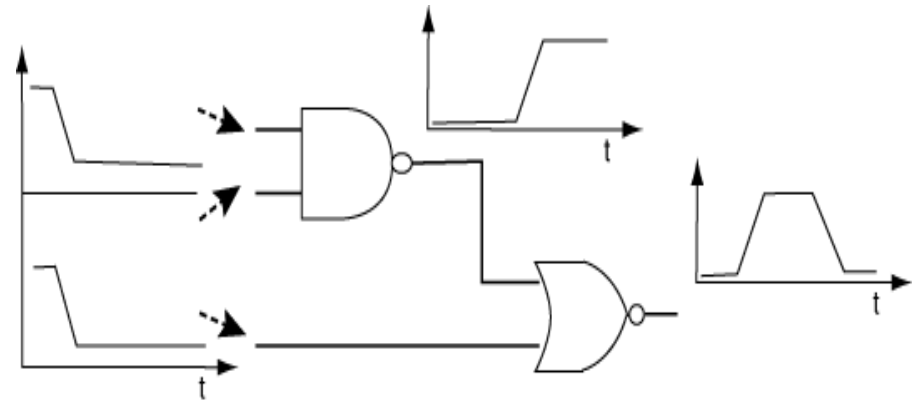
- Want to estimate delays induced by crosstalk.
- Effect of coupling capacitance C_c depends on relative transitions.
 - Aggressor changes, victim does not: C_c .
 - Aggressor, victim move in opposite directions: $2C_c$.
 - Aggressor, victim move in same direction: 0.
- Coupling effects depend on relative switching time of nets.
- Must use iterative algorithm to solve for coupling capacitances and delays.

Power optimization

- Glitches cause unnecessary power consumption.
- Logic network design helps control power consumption:
 - minimizing capacitance;
 - eliminating unnecessary glitches.

Glitching example

- NOR gate produces 0 output at beginning and end.

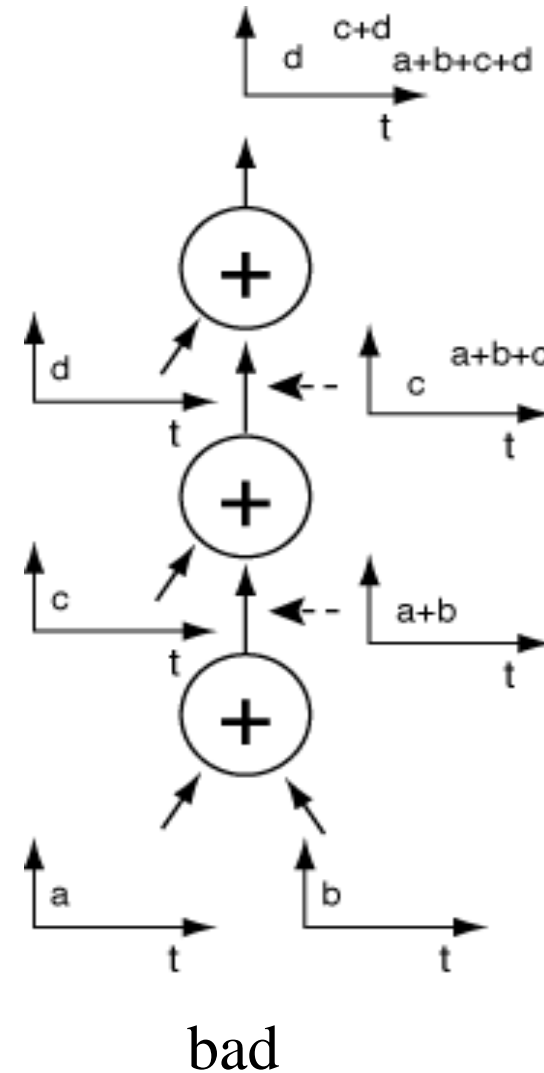


- beginning: bottom input is 1;
- end: NAND output is 1.

- Difference in delay between application of primary inputs and generation of new NAND output causes glitch.

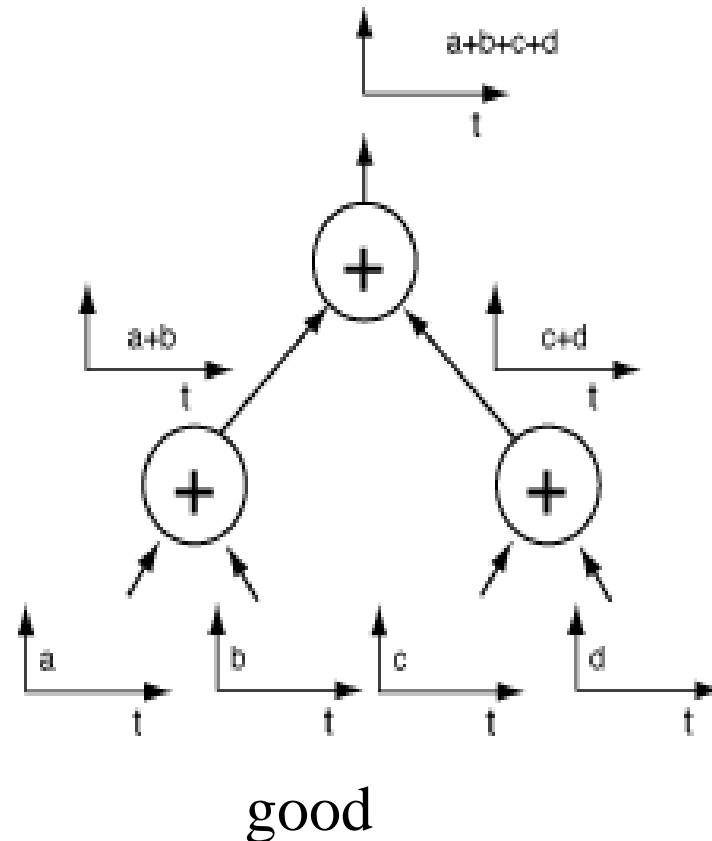
Adder chain glitching

- Unbalanced chain has signals arriving at different times at each adder.
- A glitch downstream propagates all the way upstream.



Adder chain glitching (cont'd)

- **Balanced tree** introduces multiple glitches simultaneously, reducing total glitch activity.



Signal probabilities

- Glitching behavior can be characterized by signal probabilities.
- Transition probabilities can be computed from signal probabilities if clock cycles are assumed to be independent.
- Some primary inputs may have non-standard signal probabilities— control signal may be activated only occasionally.

Delay-independent probabilities

- Compute output probabilities of primitive functions:
 - $P_{\text{NOT}} = 1 - P_{\text{in}}$
 - $P_{\text{OR}} = 1 - \prod(1 - P_i)$
 - $P_{\text{AND}} = \prod P_i$
- Can compute output probabilities of reconvergent fanout-free networks by traversing tree.

Delay-dependent probabilities

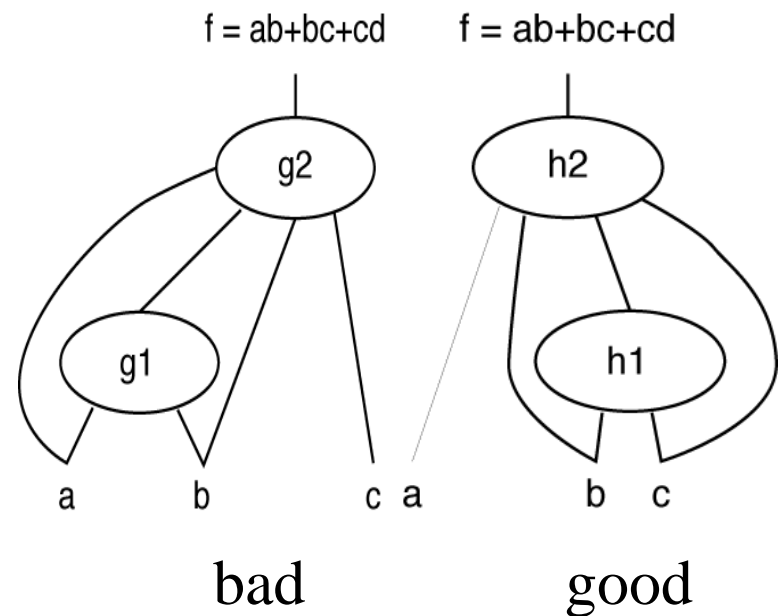
- More accurate estimation of glitching. Glitch accuracy depends on accuracy of delay model.
- Can use simulation-style algorithms to propagate glitches.
- Can use statistical models coupled with delay models.

Power estimation tools

- Power estimator approximates power consumption from:
 - gate network;
 - primary input transition probabilities;
 - capacitive loading.
- May be switch/logic simulation based or use statistical models.

Factorization for low power

- Proper factorization reduces glitching.
- In function $f = ab+bc+cd$, suppose that a has high transition probability, b and c low probabilities.
- Reduce number of logic levels through which high-probability signals must travel in order to reduce propagation of glitches.



Layout for low power

- Place and route to minimize capacitance of nodes with high glitching activity.
- Feed back wiring capacitance values to power analysis for better estimates.