Timing Analysis

Lecture 9

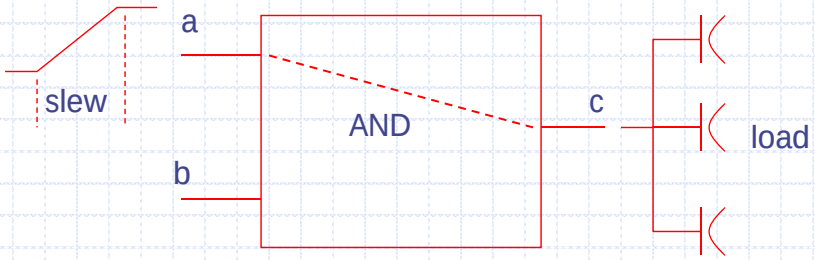
General

- Timing analysis can be done right after synthesis
  - But it can only be accurately done when layout is available
- Timing analysis at an early stage is not accurate because no detailed physical information is available

Tools

- Synopsys set of tools
- Design Compiler for synthesis and optimization
- Primetime is pretty much an industrial standard tool for timing analysis
  - Cell-based timing analysis
- Pathmill is a tool to calculate timing at transistor level

A cell delay model



The delay of “a rising => c rising” is modeled as a function of (input slew, output load)

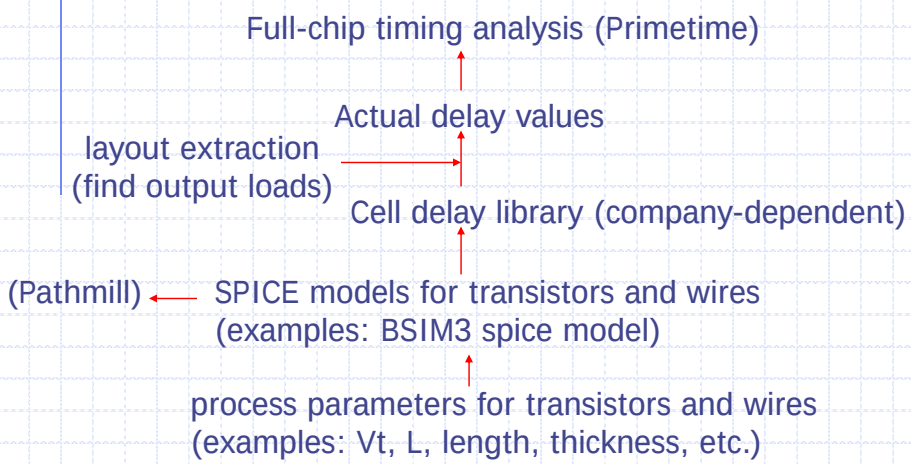
The delay of “a => c” is modeled assuming that “b” is at its steady value

wire delay is modeled differently or assumed to be included in the cell model

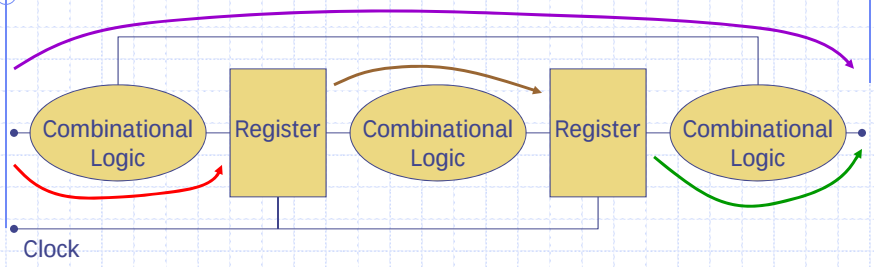
Typical delay model

- Cell-based
  - Interconnect is considered by worst-case model
- Pin-to-pin delay (a => c, b => c)
- Delay is a function of
  - Input transition
  - Output load
  - Vdd and temperature
- [min, typ, max] model
  - Also separate rising delay and falling delay

Cell-based timing analysis

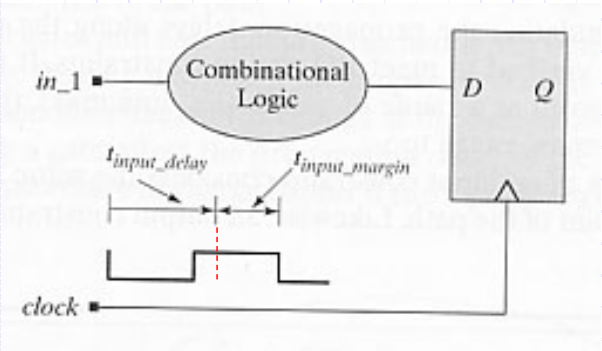


Type of Timing Paths to check

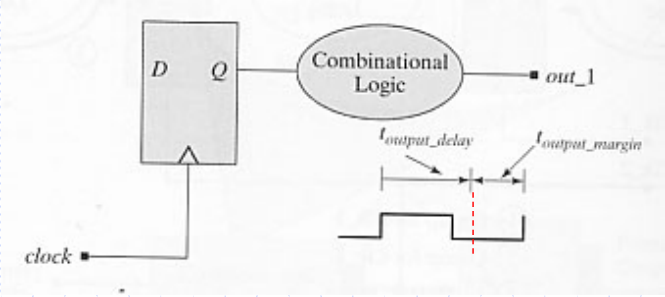


1. Input -> register
2. Register -> register
3. Register -> output
4. Input -> output

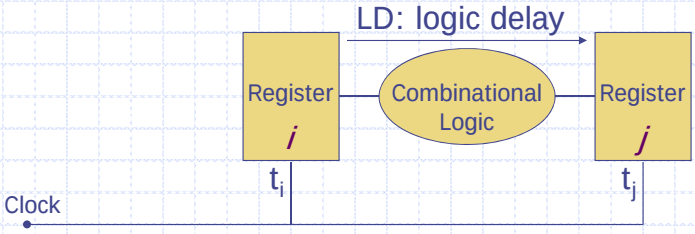
Input delay constraint



Output delay constraint

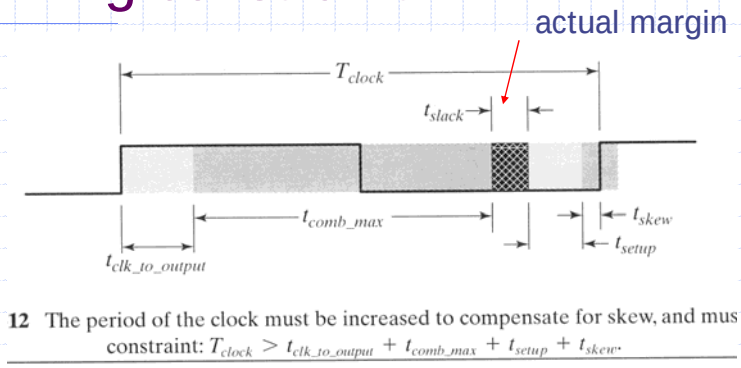


Clock Scheduling



clock skew = |ti – tj|

Timing constraint



12 The period of the clock must be increased to compensate for skew, and mus
constraint: $T_{clock} > t_{clk\_to\_output} + t_{comb\_max} + t_{setup} + t_{skew}$.

- $T_{clock} > t_{clk\_to\_output} + t_{comb\_max} + t_{setup} + t_{skew}$

Timing check: setup time

- The setup time constraint of a flip-flop specifies a time interval before the active edge of clock.
- Data must arrive before the interval.
- $setup(data, posedge clk, 5);

Setup time example

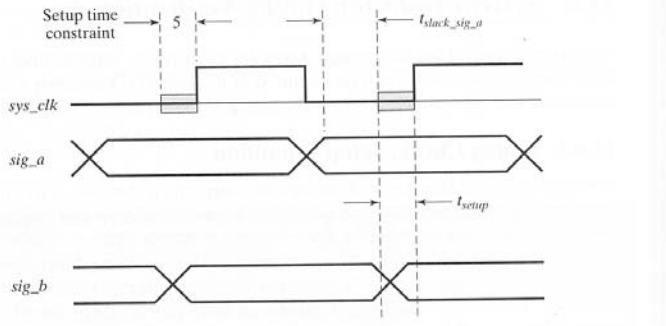


FIGURE 11-23 The setup time constraint on *sys_clk* requires *sig_a* and *sig_b* to be stable during the setup interval located ahead of the active edge of the clock. *sig_b* violates the setup constraint.

Timing check: hold time

- The hold time constraint specifies an interval after the active edge of clock.
- Data must be stable in the interval.
- $hold(data, posedge clk, 2);

Hold time example

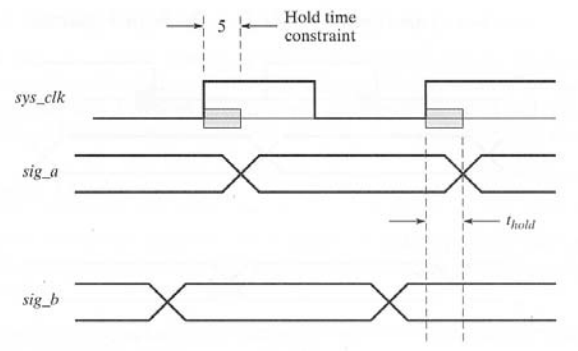


FIGURE 11-25 The hold time constraint on *sys_clk* is violated by *sig_a*.

Pulse Width

- The width of the clock pulse must not be too small.
- $width(posedge clock_a, t_mpw);

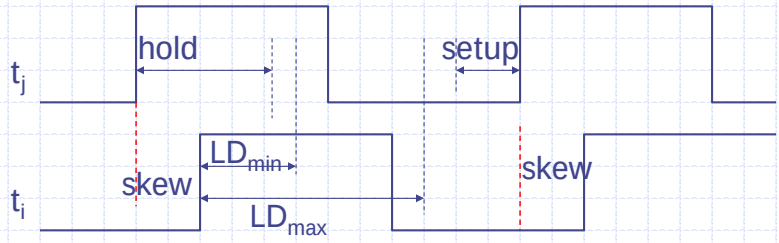
Clock Skew

- Signal skew is the arriving time difference of two signals.
- Clock skew should be low.
- $skew(negedge clk1, negedge clk2, t_skew);

Verilog timing check

- Verilog timing check provides only rough check
- Usually it needs a separate tool (timing analysis tool) to do the check
- It is not part of the verilog simulator nor part of the synthesis process

Timing Constraint Examples



- $skew_{ij}$ constraint: $hold_{max} - (t_i - t_j) < LD_{min}$
  - i.e. (hold – skew) < LD
- $skew_{ij} = t_i - t_j < CP - LD_{max} - setup_{max}$
  - i.e. LD < (clock – skew) – setup

LD: Logic Delay

Factors that affect timing

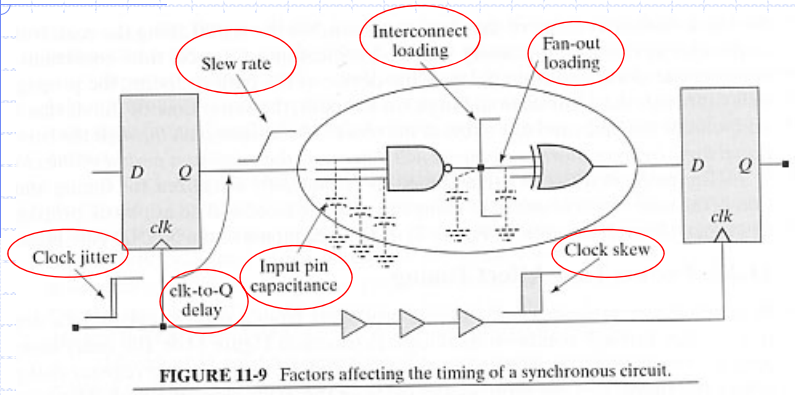


FIGURE 11-9 Factors affecting the timing of a synchronous circuit.

Multiple clock domains

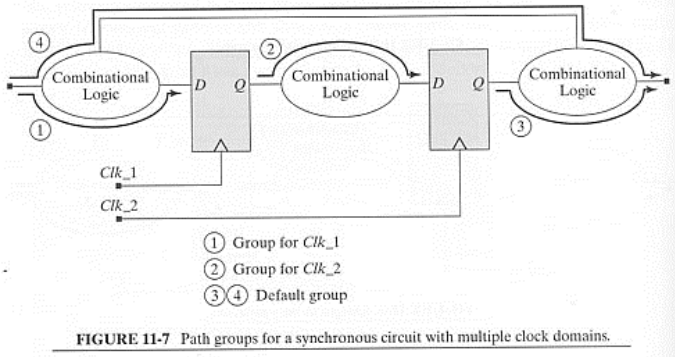


FIGURE 11-7 Path groups for a synchronous circuit with multiple clock domains.

- clock skew = arrival time of clk_1 – arrive time of clk_2

Model to check via timing DAG (find worst LDs between FFs)

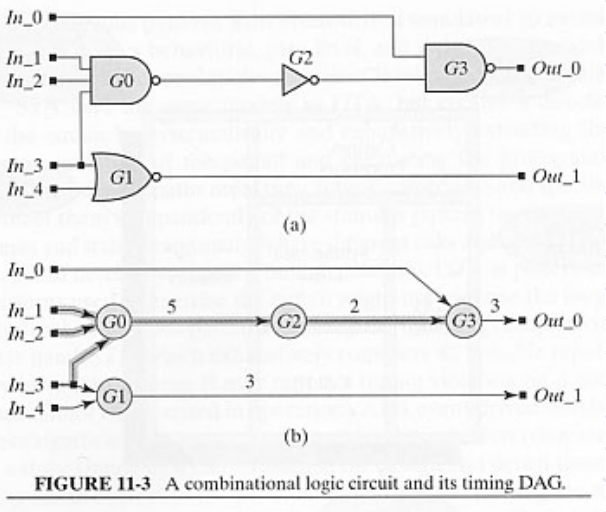
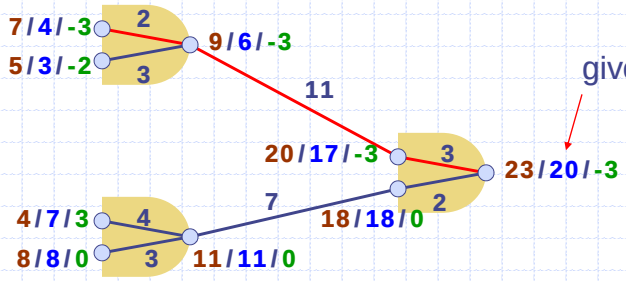


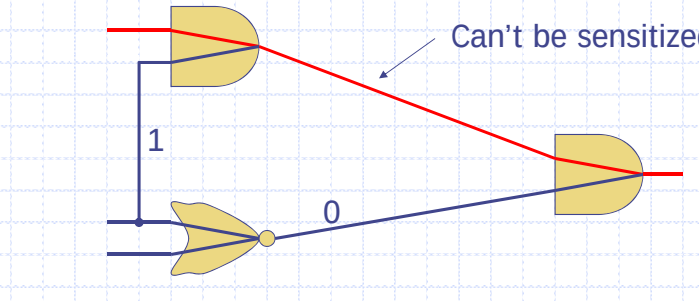
FIGURE 11-3 A combinational logic circuit and its timing DAG.

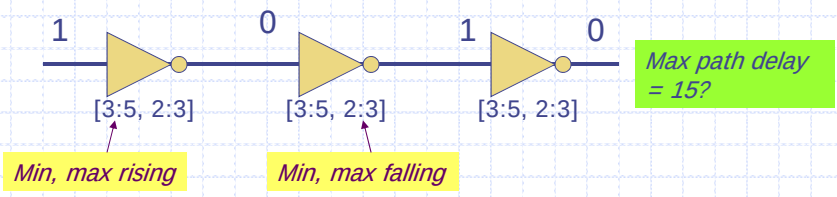
Static Timing Analysis



- Arrival time: input -> output, (take max)
- Required arrival time: output -> input, (take min)
- Slack = required arrival time – arrival time

False Paths (don't affect timing)





Logical false path

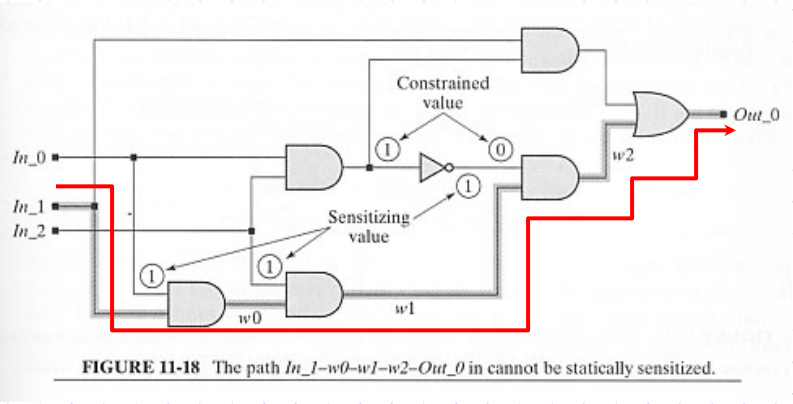
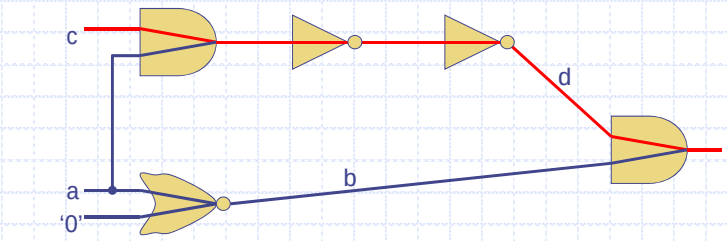
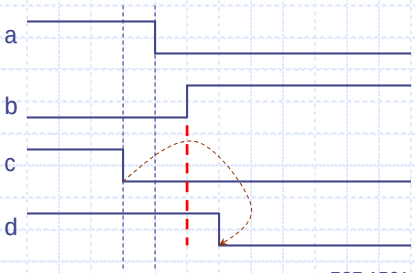


FIGURE 11-18 The path *In_1–w0–w1–w2–Out_0* in cannot be statically sensitized.

Dynamically Sensitized Paths





The path is not logically sensitizable but time-wise it can be

A circuit with false path

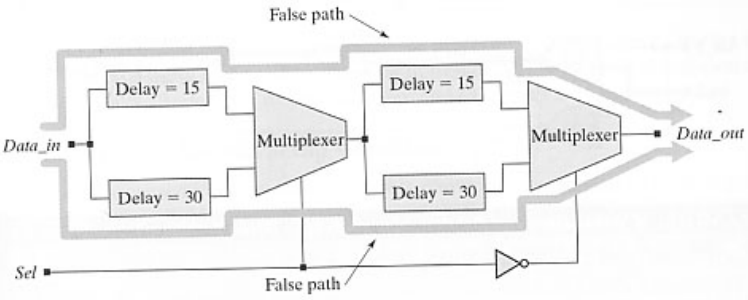
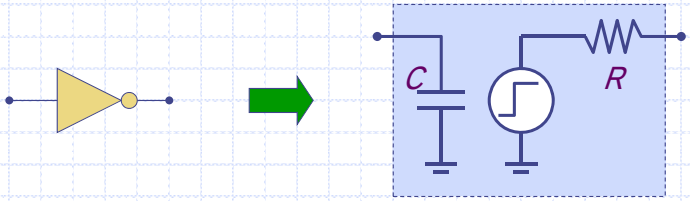


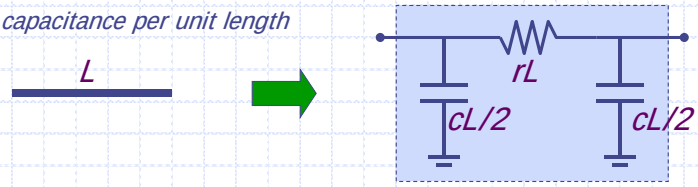
FIGURE 11-20 A circuit with false paths.

Gate and Wire Model

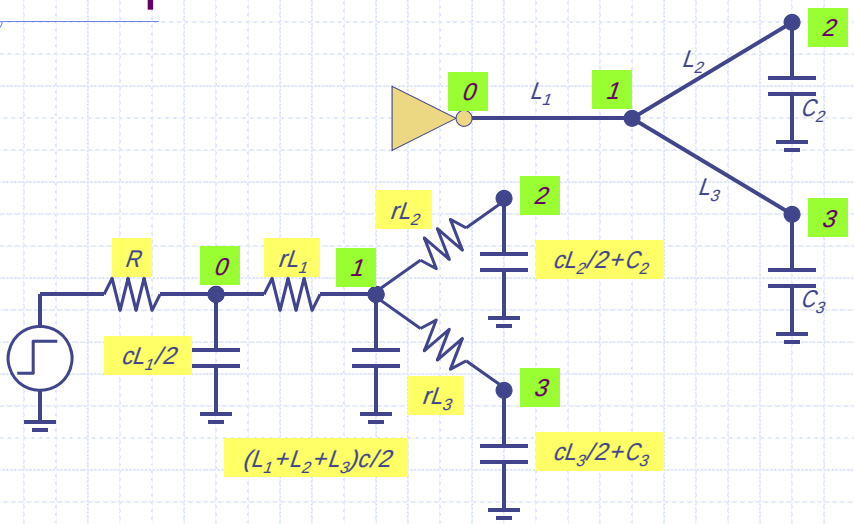


r: resistance per unit length

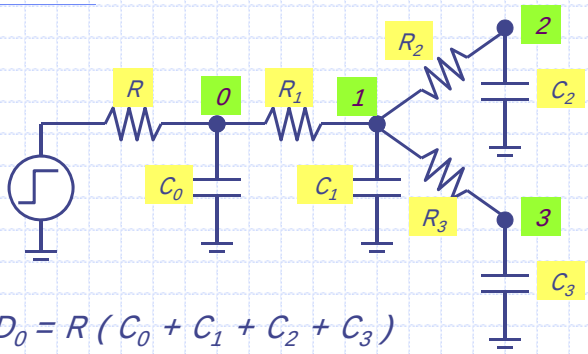
c: capacitance per unit length



Example of Model



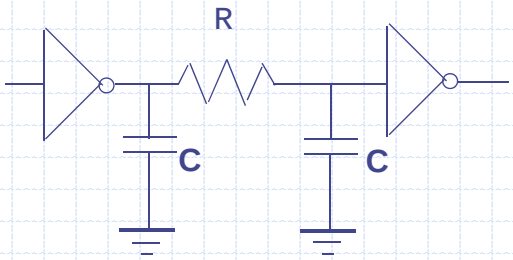
Delay Estimation



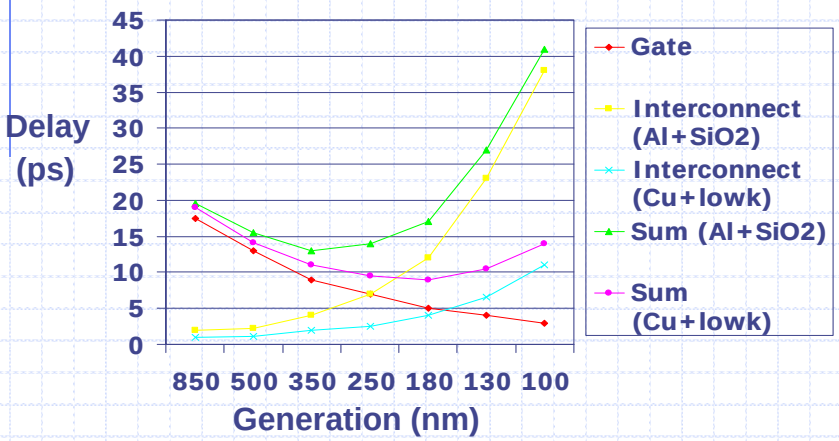
- $D_0 = R\,(C_0 + C_1 + C_2 + C_3)$
- $D_1 = D_0 + R_1\,(C_1 + C_2 + C_3)$
- $D_2 = D_1 + R_2\,C_2$
- $D_3 = D_1 + R_3\,C_3$

Interconnect Delay

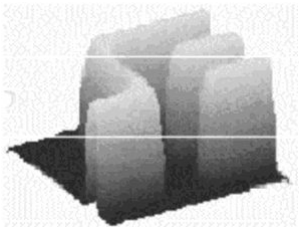
- Interconnect delay is caused by parasitic capacitance and resistance



Myth: Interconnect Dominates?

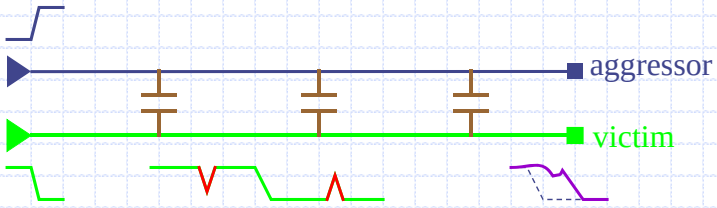


Interconnect Size Scaling



- Wire width scales faster than wire height ⇒ wires are thinner and taller
- Wires are placed closer
- Coupling capacitance start to dominate substrate capacitance

Other: Crosstalk Noise



- Crosstalk noise may cause
  - Glitch and logical error
  - Extra propagation delay

Elimination of Timing Violation

Action	Effect
Increase clock period	Eliminates the violation, constrained by specifications
Reroute critical path	Reduce interconnect delays
Resize and substitute devices	Reduce device delays and improve setup and hold margins
Redesign clock tree	Reduce clock skew
Substitute a different algorithm	Reduce path delays
Substitute architecture	Reduce path delays
Pipeline/retiming	Reduce path delays
Change technologies	Reduce path and device delays