CS250
VLSI Systems Design

Fall 2020

John Wawrzynek

with

Arya Reais-Parsi

Project Team Presentations

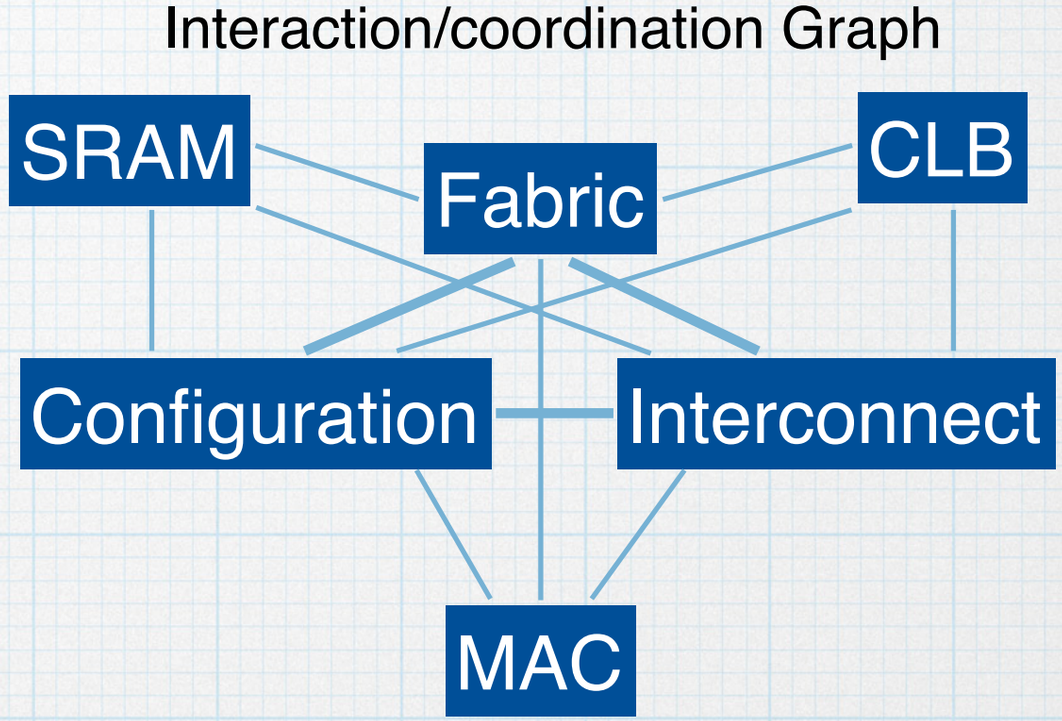
- In class, Oct 1 (this Thursday)
- Target 10 minutes (with discussion)
- Slides with illustrations (powerpoint, …)
- One presentation each from: config, CLB, SRAM, MAC teams
- Two interconnection presentations
- Three fabric team presentations:
  - tool support
  - high-level fabric architecture
  - simulation, testing, integration plan
- Following week, private group meetings with Arya & John to get feedback and brainstorm ideas

Project Team Presentations

- The point is to get the discussion going on the function and implementation of your piece.
- You are responsible for a "straw-man"/draft proposal
- Okay to leave some issues open for now
- Outline
  - Only one person needs to speak but, introduce team members
  - Describe your proposed function/features and structure (block diagram/circuit) of your piece
  - Describe how you plan to refine the definitions of function/ structure and to optimize the design
  - Say something about implementation strategy
  - Say something about what information you will need from other teams and what other teams will need from you

Project Teams

- If you have questions about how you ended up in which team, mail me or set up appointment
- If you have questions about your team's role and responsibility, ask now, or mail us later
- If you don't have email contacts for your other team members, ask now, or mail us later
- To prepare for the presentations next week, not necessary right now to reach out to other groups, but feel free to do so



Circuits Topics: Basic (review?)

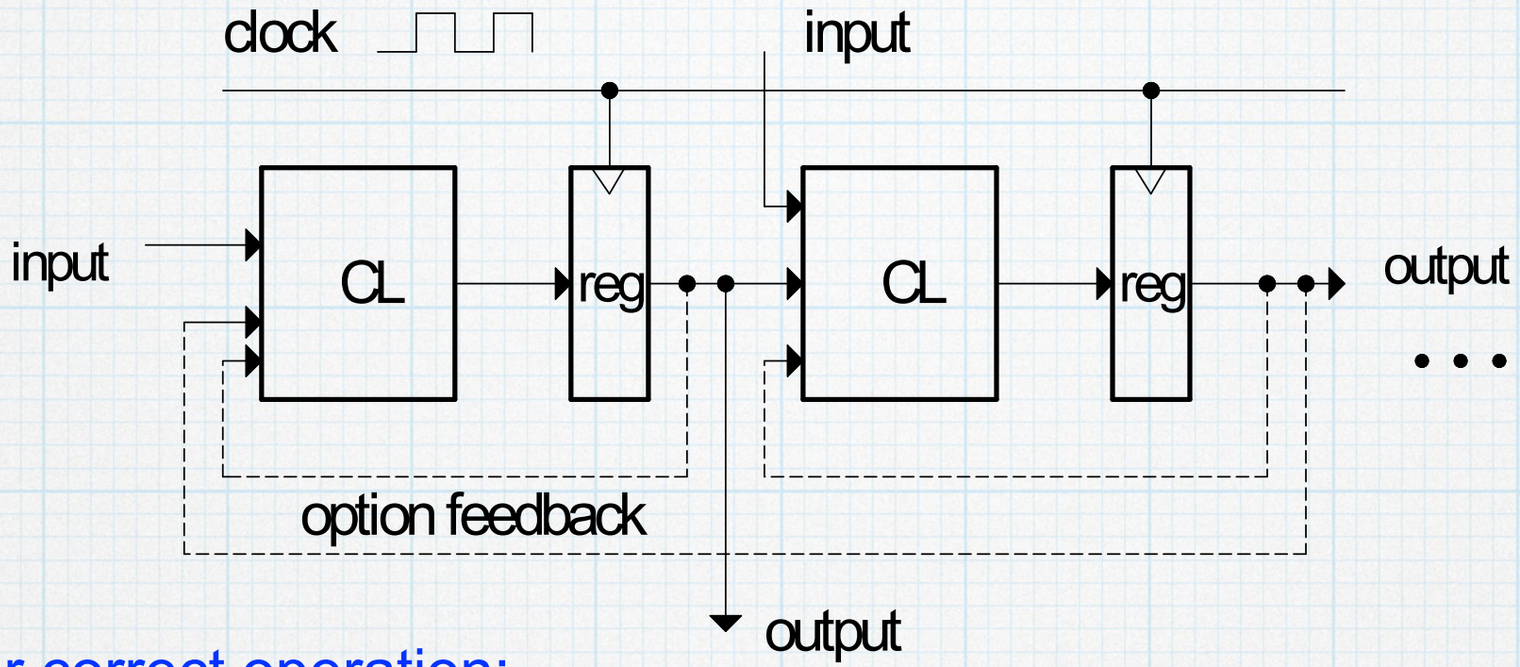
What you need to know as a VLSI Systems designer.

- Processing/devices: planar, finfets, GDR
- Device models: switch, RC, Vth
- Logic circuits: gates, muxes, transmission gates, FFs
- Circuit Delay: gate delay, wire delay, FET sizing
- Circuit Power: formulation/ factors
- System Delay: factors, optimization
- System Power: factors, optimization

System Delay

- Critical Path
- Optimization Techniques
- Clock distribution
-

In General …

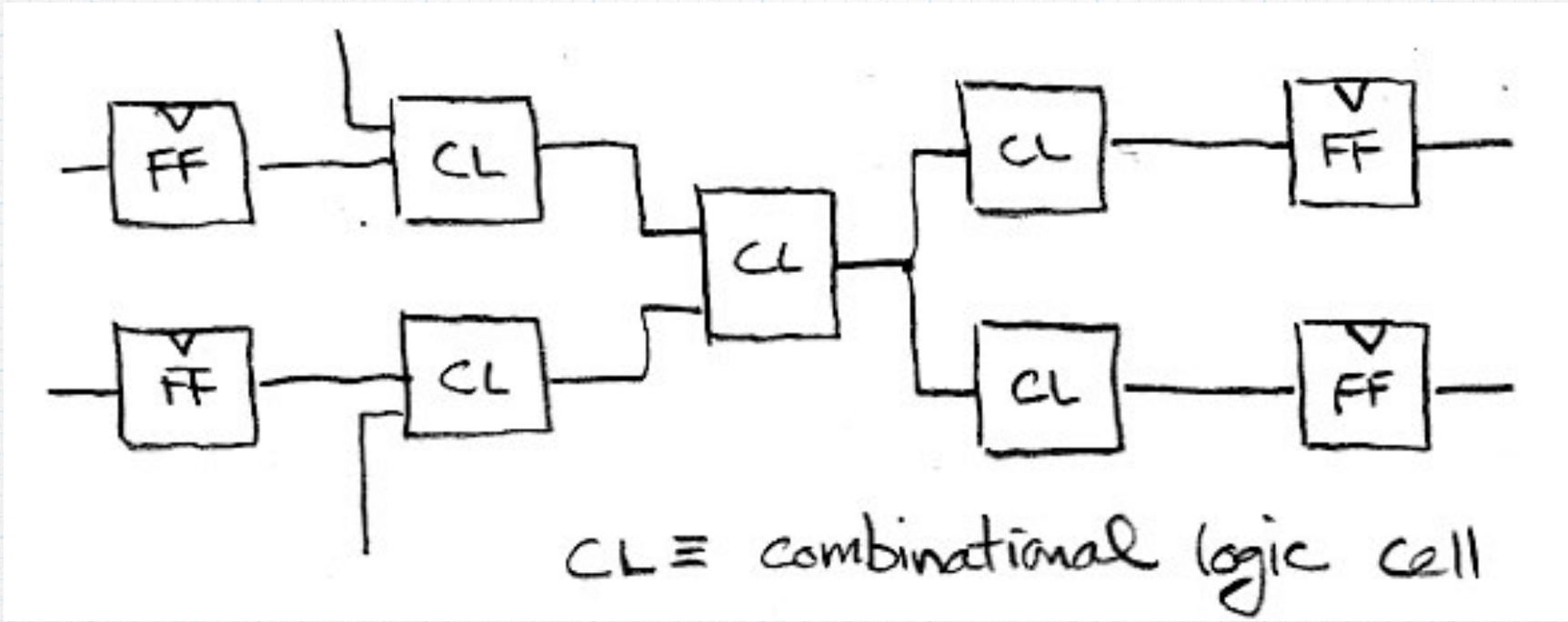


For correct operation:

$$T \geq \tau_{clk \rightarrow Q} + \tau_{CL} + \tau_{setup}$$ for all paths.

- How do we enumerate ***all*** paths?
  - Any circuit input or register output to any register input or circuit output?
- Note:
  - "setup time" for outputs is a function of what it connects to.
  - "clk-to-q" for circuit inputs depends on where it comes from.

Components of Path Delay



1. # of levels of logic
2. Internal cell delay
3. wire delay
4. cell input capacitance
5. cell fanout
6. cell output drive strength

How do we optimize?

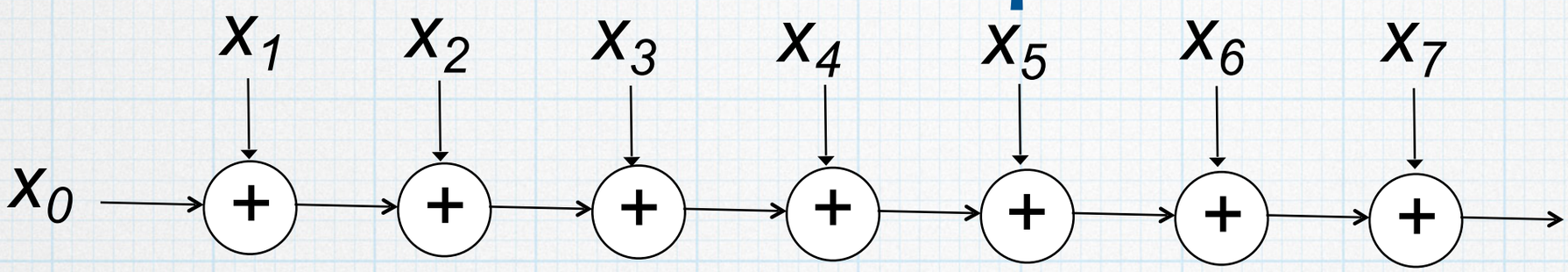
Tackle "critical path"

Synthesis tools approximate path delay and attempt to optimize by rearranging logic network and choosing appropriately sized cells.

Place and route tools attempt to minimize wire delay on critical paths.

"Logical Effort" method for hand sizing of transistors.

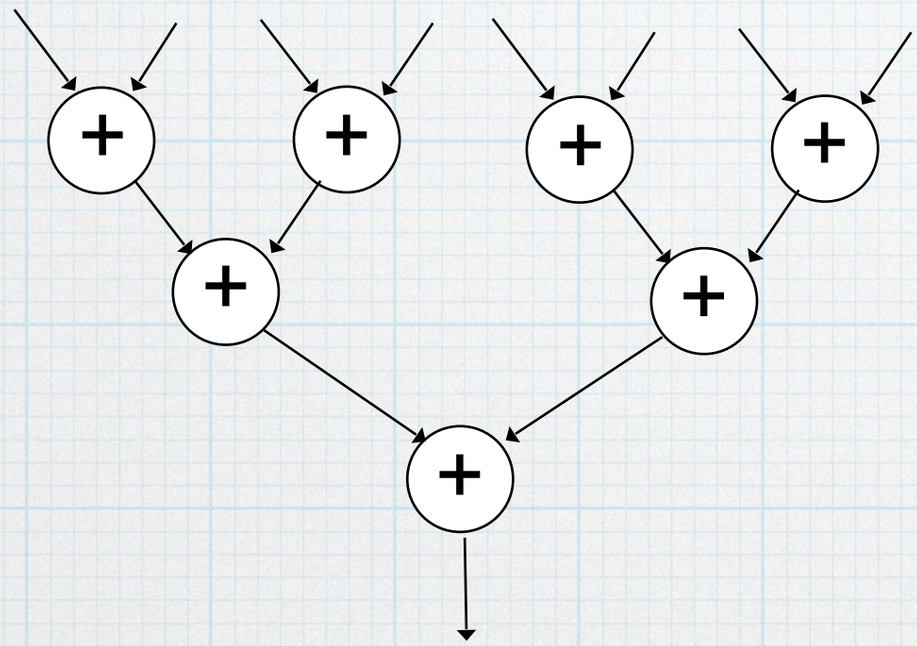
Trees for optimization



$T = O(N)$

$((((((x_0 + x_1) + x_2) + x_3) + x_4) + x_5) + x_6) + x_7$

Same number of operations (N-1)



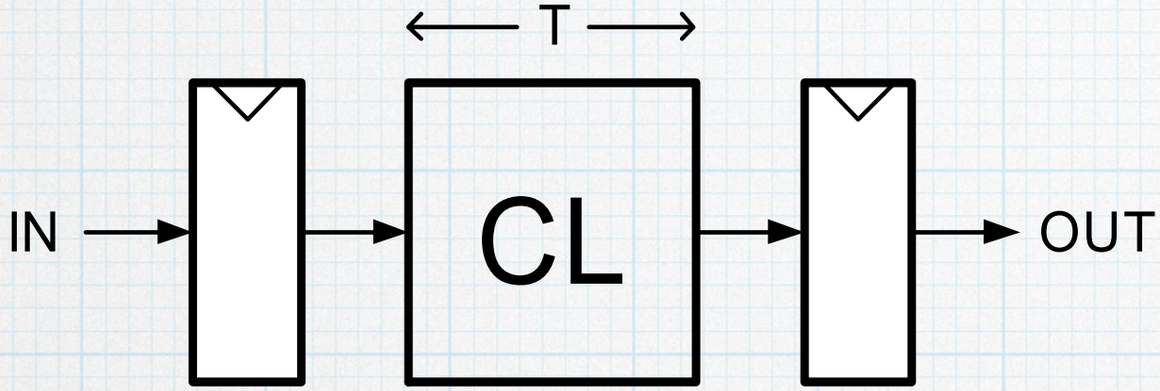
$T = O(\log N)$

$((x_0 + x_1) + (x_2 + x_3)) + ((x_4 + x_5) + (x_6 + x_7))$

- *What property of "+" are we exploiting?*
- *Other associate operators? Boolean operations? Division? Min/Max?*

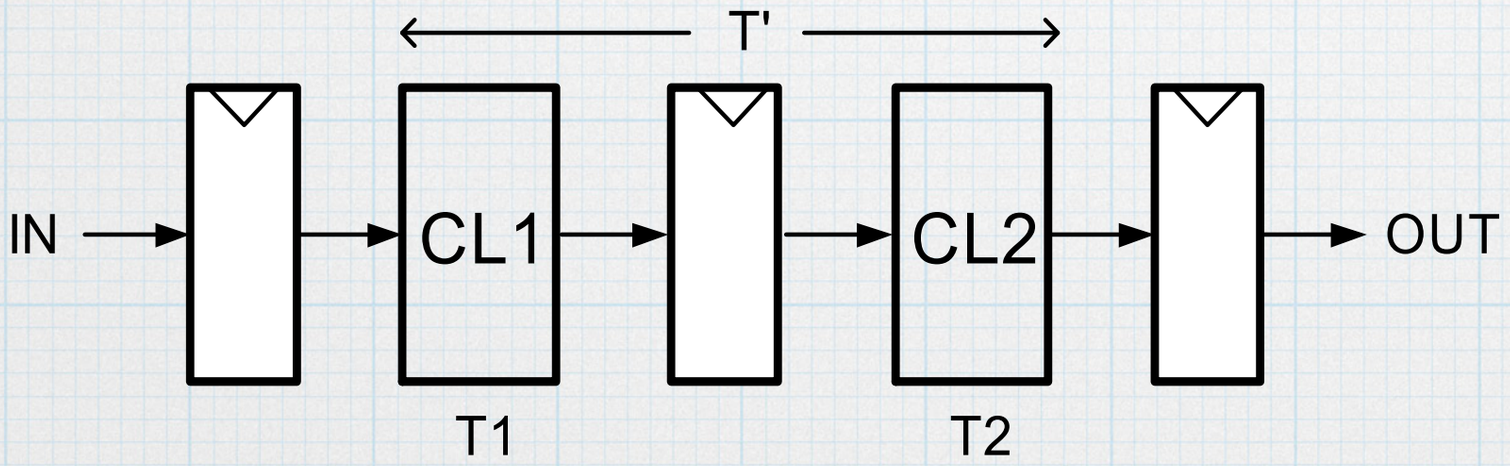
Pipelining

- General principle:



Assume T=8ns
T_{FF}(setup +clk→q)=1ns
F = 1/9ns = 111MHz

- Cut the CL block into pieces (stages) and separate with registers:



Assume T1 = T2 = 4ns

Simple, except in classes of feedback (loop carry dependance)

System Power

- Chip/block level Power
- Optimization for power and energy efficiency
- Power distribution

Energy and Power

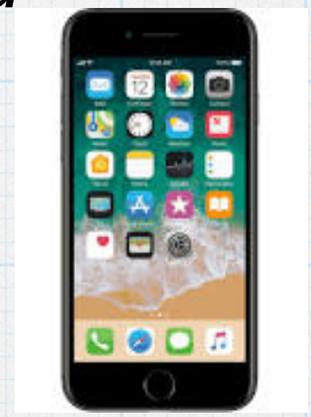
Energy is the ability to do work (W).

Power is rate of expending energy.

$$P = \frac{dW}{dt}$$

Energy Efficiency: energy per operation

Heat is a byproduct of computation. Heat dissipated is proportional to the energy used per unit time, P.



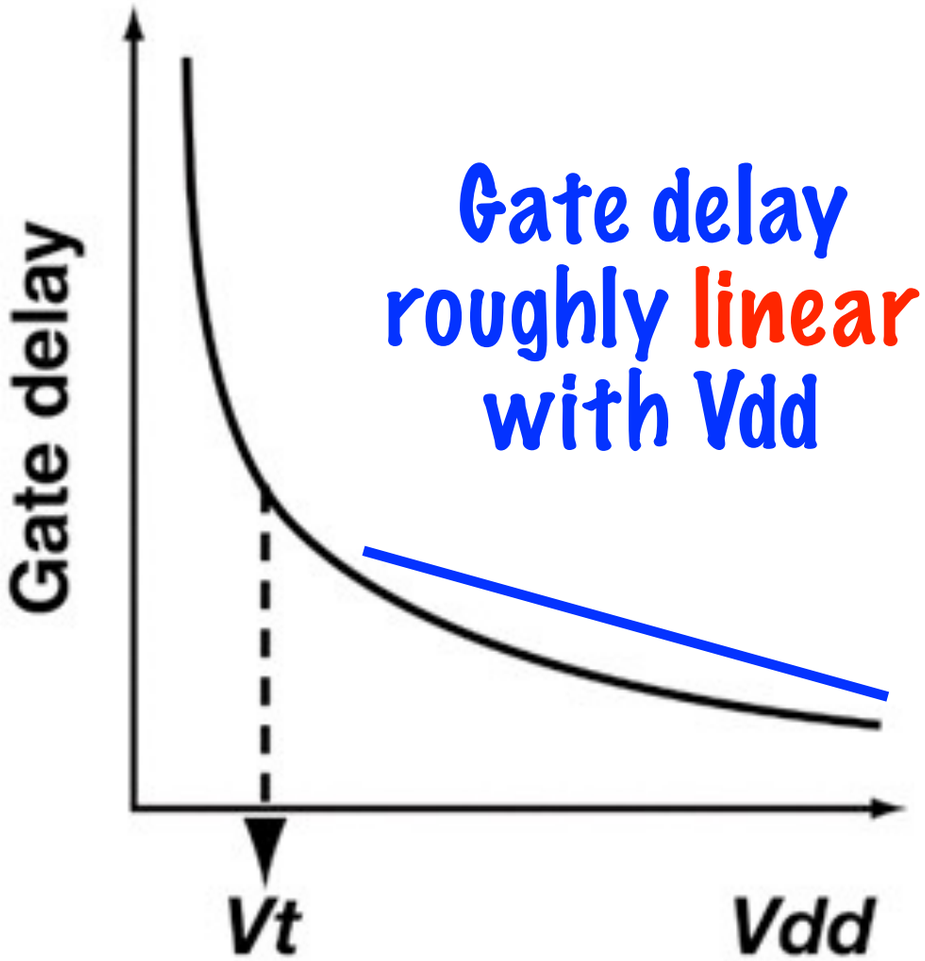
- ***Handheld and portable*** (battery operated):
  - Energy Efficiency - limits battery life
  - Power - limited by heat

- ***Infrastructure and servers*** (connected to power grid):
  - Energy Efficiency - dictates operation cost
  - Power - heat removal contributes to TCO

Remember: reducing power is easy - just slow down.
Improving energy efficiency is difficult.

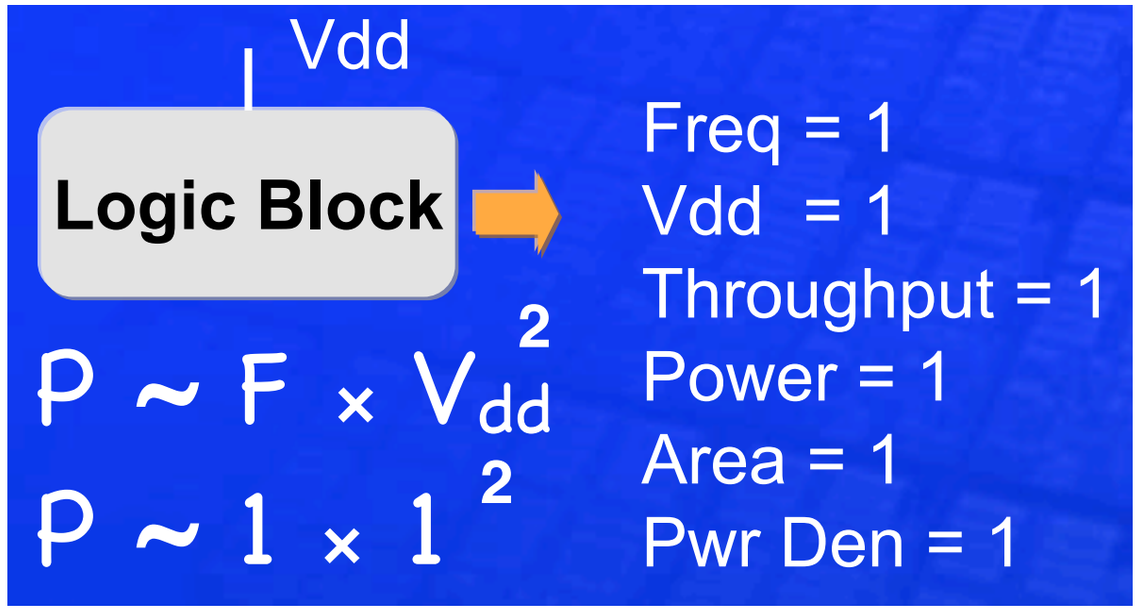
Five low-power design techniques

- **Parallelism and pipelining**
- **Power-down idle transistors**
- **Slow down non-critical paths**
- **Clock gating**
- **Thermal management**



Gate delay roughly linear with Vdd

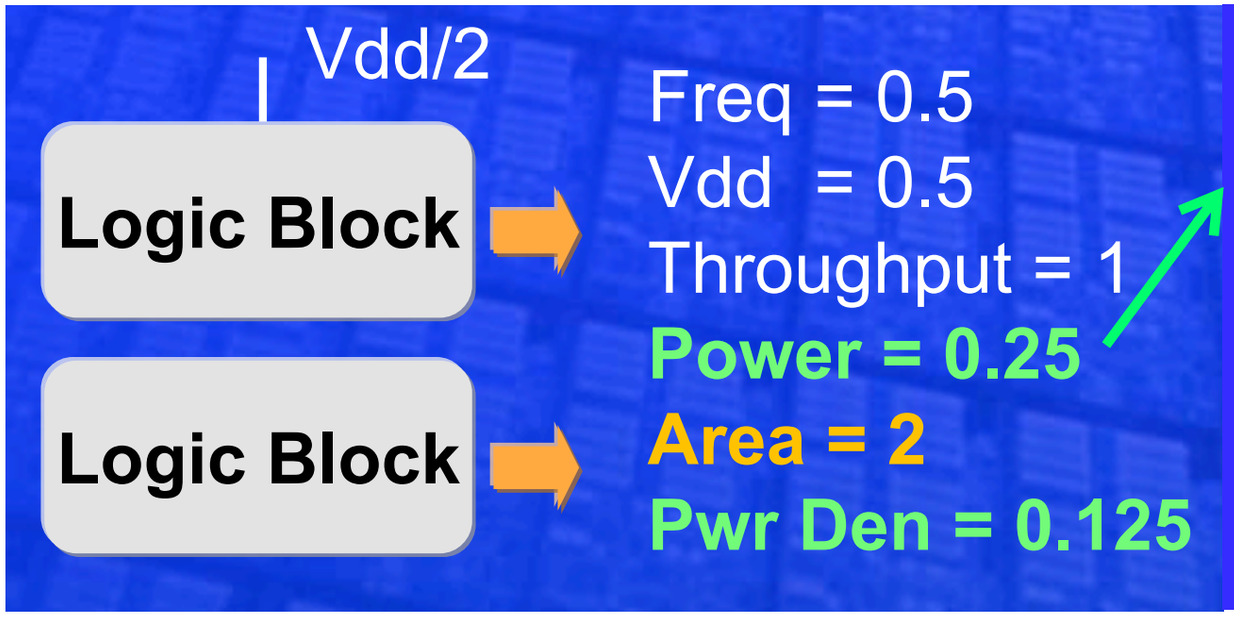
And so, we can transform this:



Block processes stereo audio. 1/2 of clocks for "left", 1/2 for "right".

Into this:

Top block processes "left", bottom "right".



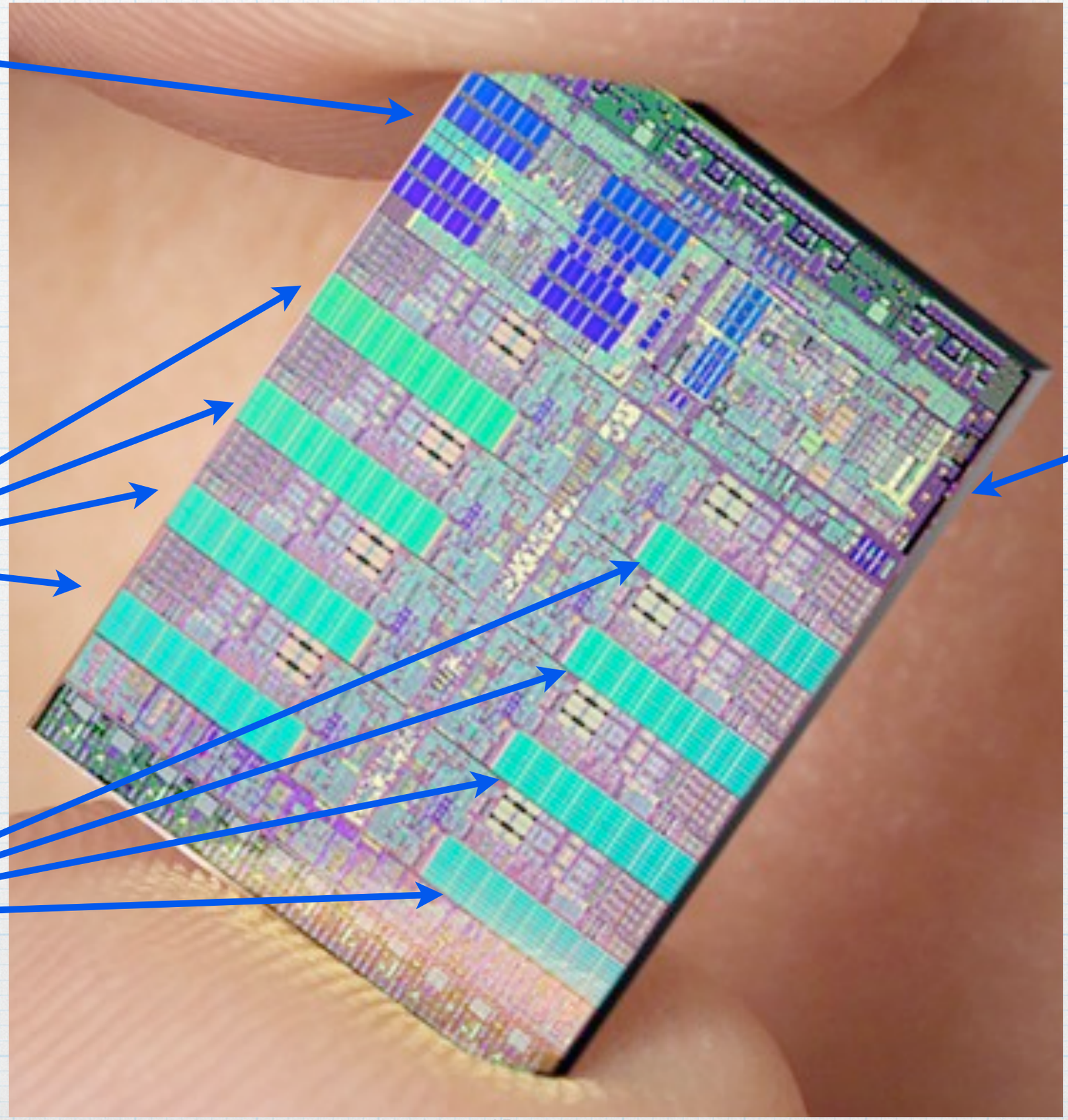
$P \sim \#blks \times F \times V_{dd}^2$

$P \sim 2 \times 1/2 \times 1/4 = 1/4$

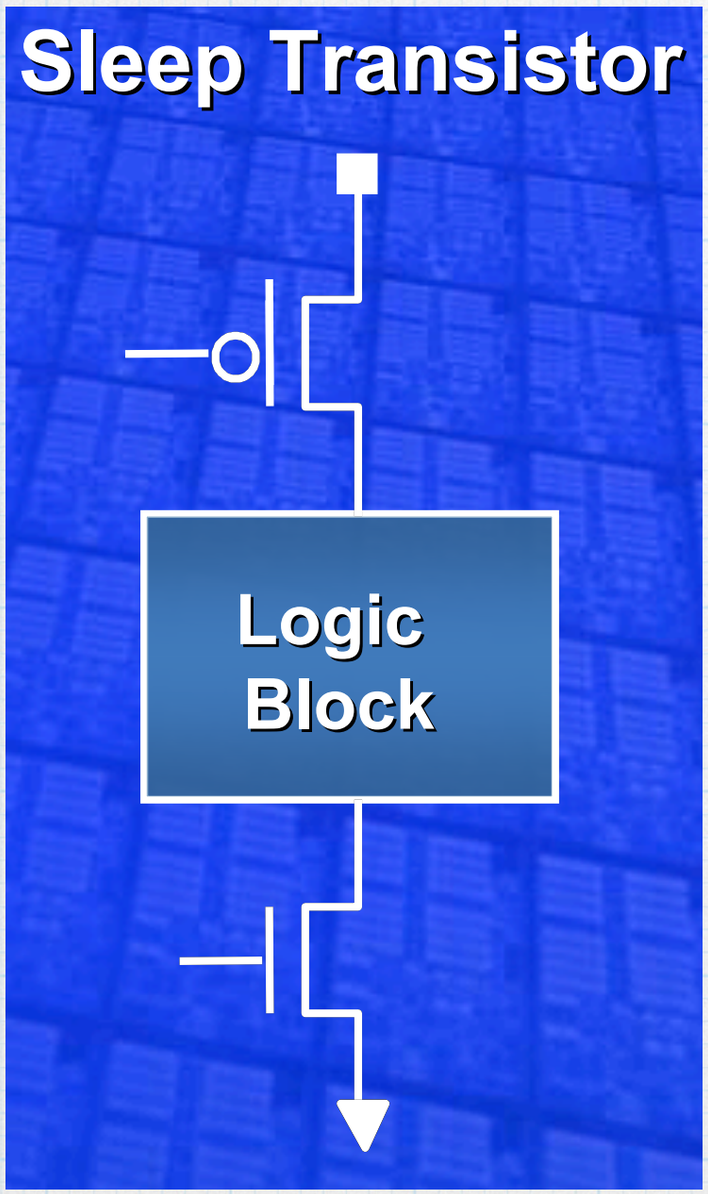
CV^2 power only

THIS MAGIC TRICK BROUGHT TO YOU BY CORY HALL …

Cell (PS3 Chip): 1 CPU + 8 "SPUs"



Add “sleep” transistors to logic …



Example: Floating point unit logic.

When running fixed-point instructions, put logic “to sleep”.

+++ When “asleep”, leakage power is dramatically reduced.

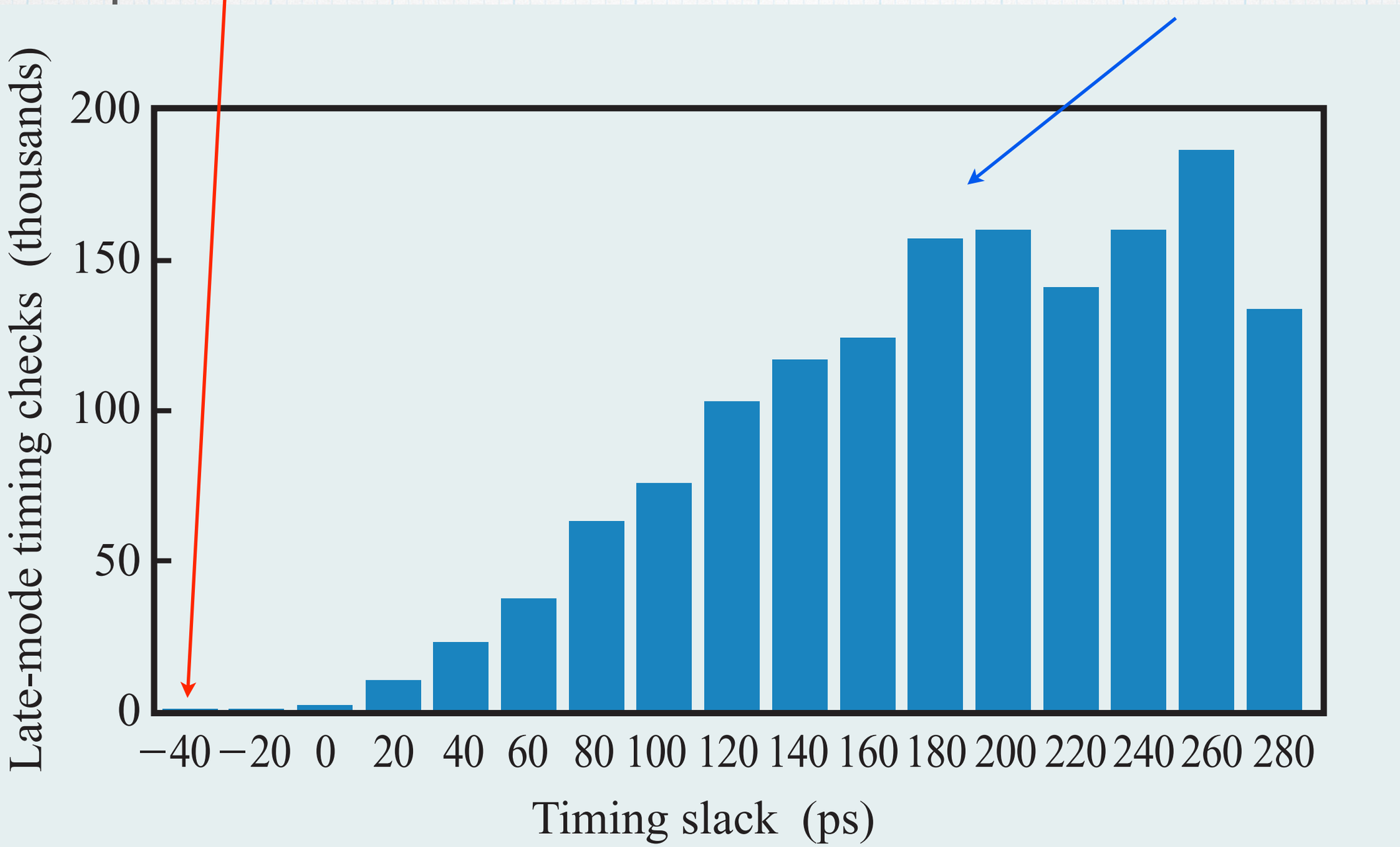
--- Presence of sleep transistors slows down the clock rate when the logic block is in use.

Fact: Most logic on a chip is "too fast"

- A product that

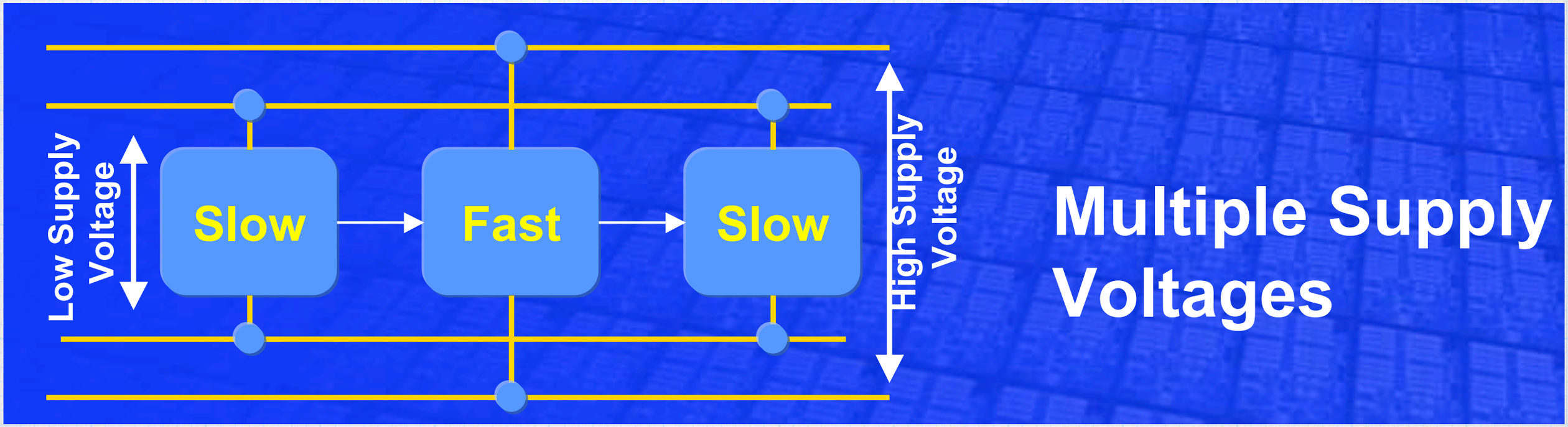
The critical path

Most wires have hundreds of picoseconds to spare.



From "The circuit and physical design of the POWER4 microprocessor", IBM J Res and Dev, 46:1, Jan 2002, J.D. Warnock et al.

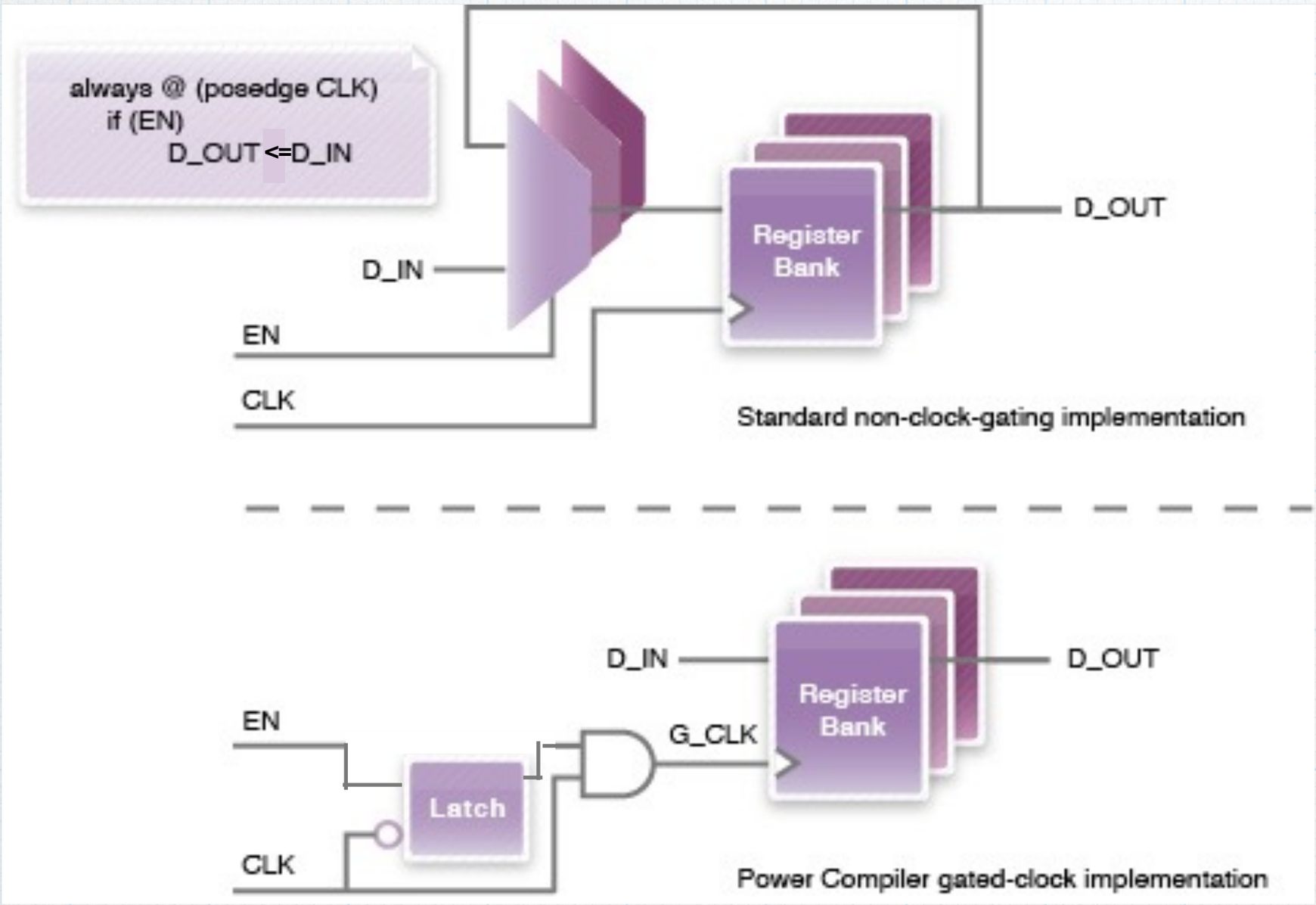
Use several supply voltages on a chip …



Why use multi-Vdd? We can reduce dynamic power by using low-power Vdd for logic off the critical path.

What if we can't do a multi-Vdd design?
In a multi-Vt process, we can reduce leakage power on the off critical path logic by using high-Vth transistors.

Clock Gating Reduces Clock Load



"Up to 70% power savings at the block level, for applicable circuits"
Synopsis Data Sheet

Keep chip cool to minimize leakage

A recipe for thermal runaway

Junction Temperature (T_J °C)	Normalized Static Power or I_{CCINTQ} Typical
25	1.00
50	1.46
85	2.50
100	3.14

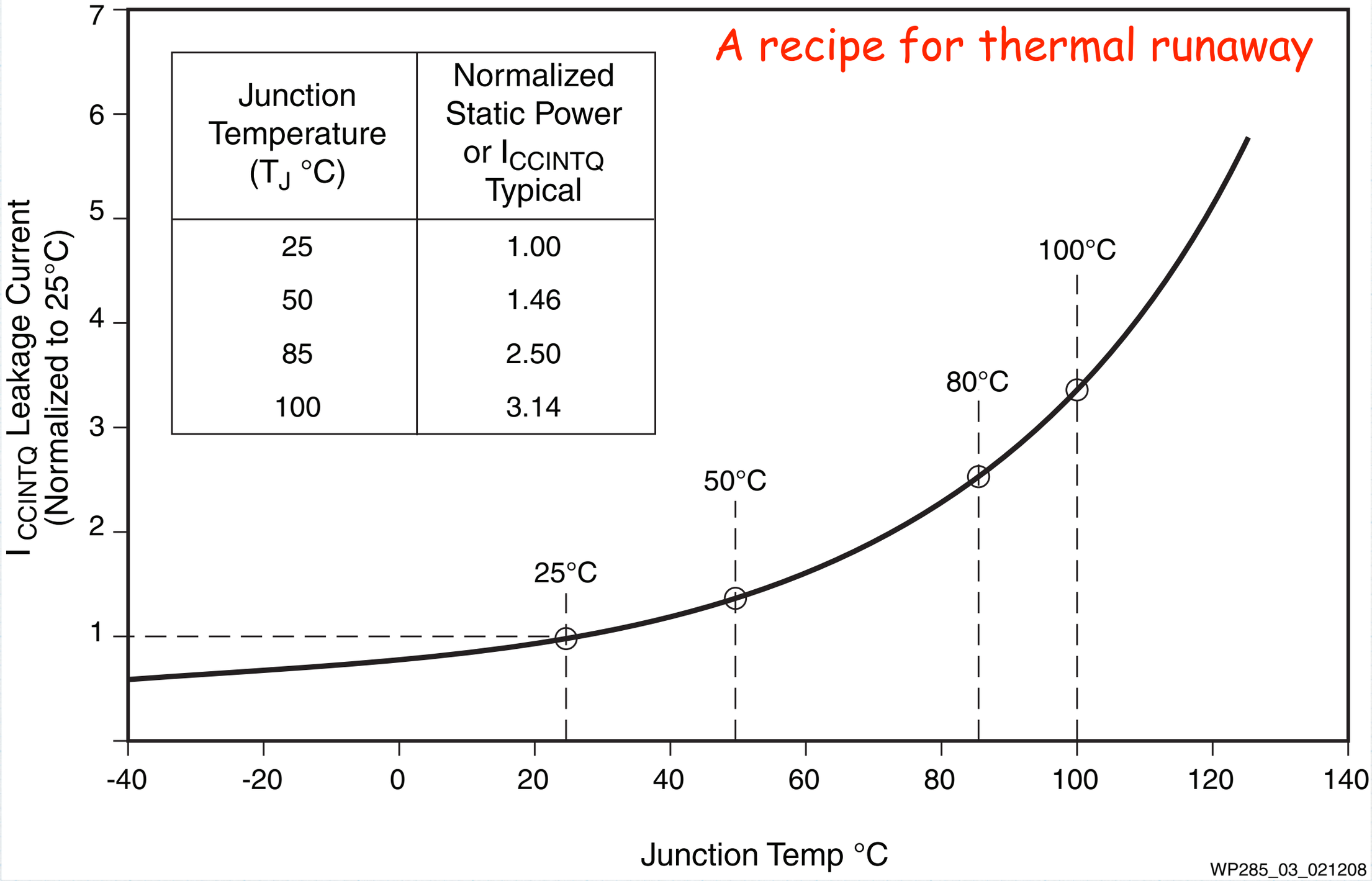


Figure 3: **I_{CCINTQ} vs. Junction Temperature with Increase Relative to 25°C**

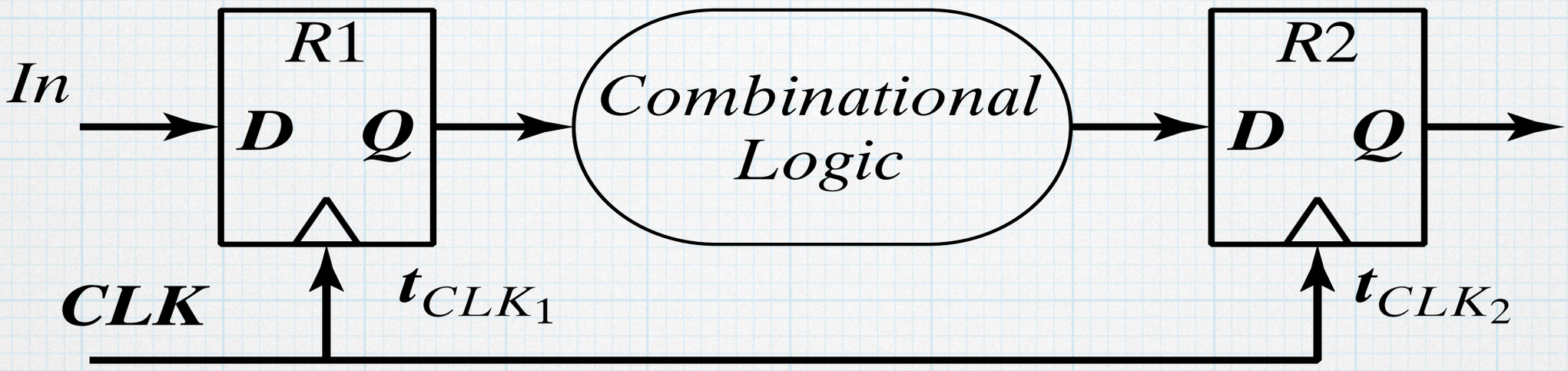
Circuits Topics: Advanced

- **Clocks and clocking:**
  - clock drivers and distribution
  - skew effects
  - hold-time
  - clock domains and synchronization
  - Phase-locked Loops (PLL) / Delay-locked Loops (DLL)
  - Globally Asynchronous locally Synchronous (GALS) clocking
- **Power supply and use**
  - Power distribution and decoupling capacitors
  - Dynamic Voltage and Frequency Scaling (DVFS)
  - voltage regulators
  - device stacking, power gating, clock gating, multi-threshold
  - Multi-voltage systems
  - charge pumps
  - latch-up / well plugs
- **Input/Output**
  - Electrostatic Discharge (ESD) suppression / pad-drivers
  - High-speed I/O, Serializer/Deserializer (SerDes)
  - packaging

Clocks and Clocking

- In my experience building chips and bringing up systems, the great majority of problems relate to clocks and power.

Revised Timing Constraints



$t_{clk\text{-}q,max}$ $t_{logic,max}$

$t_{clk\text{-}q,min}$ $t_{logic,min}$

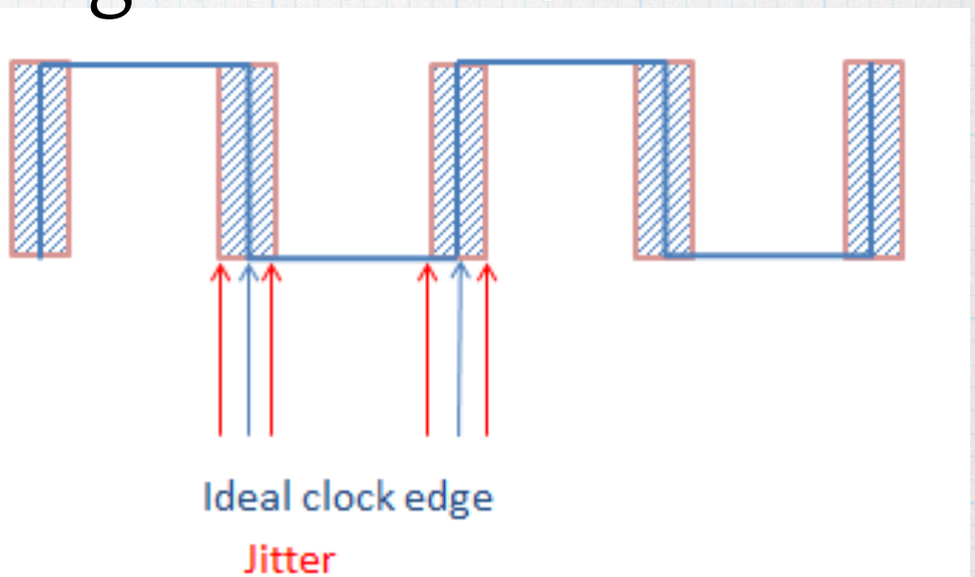
t_{setup}, t_{hold}

Cycle time (max): $T_{Clk} > t_{clk\text{-}q,max} + t_{logic,max} + t_{setup}$

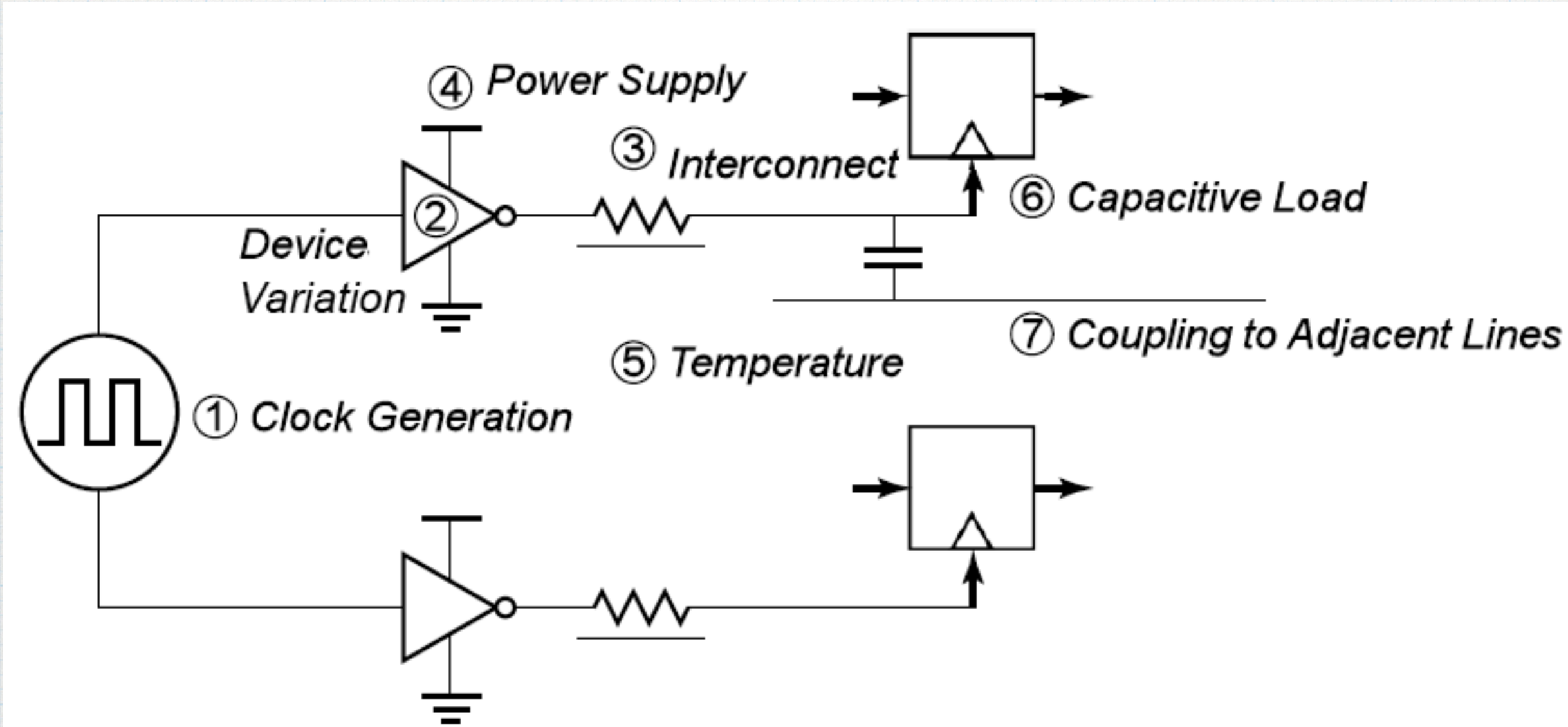
Race margin (min): $t_{hold} < t_{clk\text{-}q,min} + t_{logic,min}$

Clock Nonidealities

- **Clock skew: t_{SK}**
  - Time difference between the sink (receiving) and source (launching) clock edge; deterministic + random
- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) t_{JS}
  - Long term t_{JL}
- **Variation of the pulse width**
  - Important for level sensitive clocking

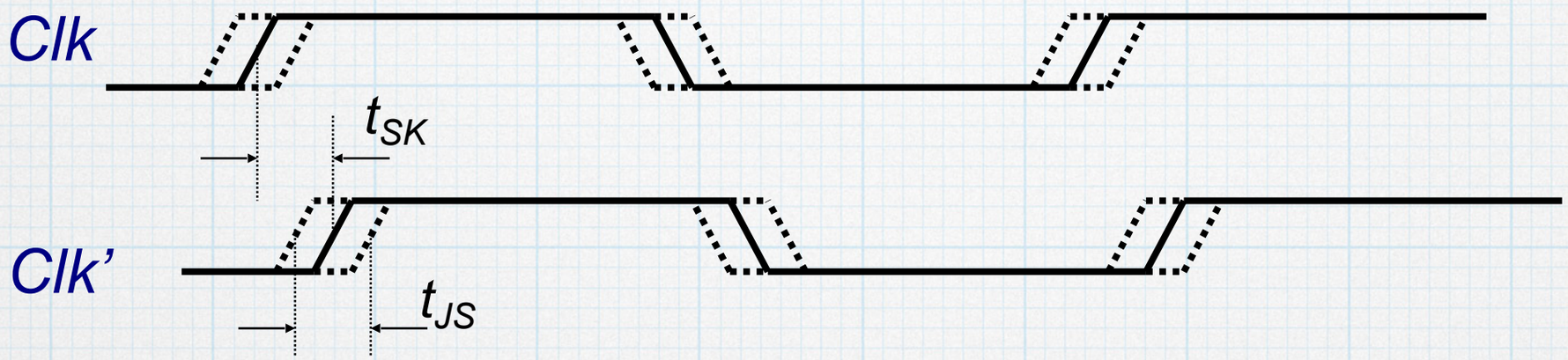


Sources of Clock Uncertainties

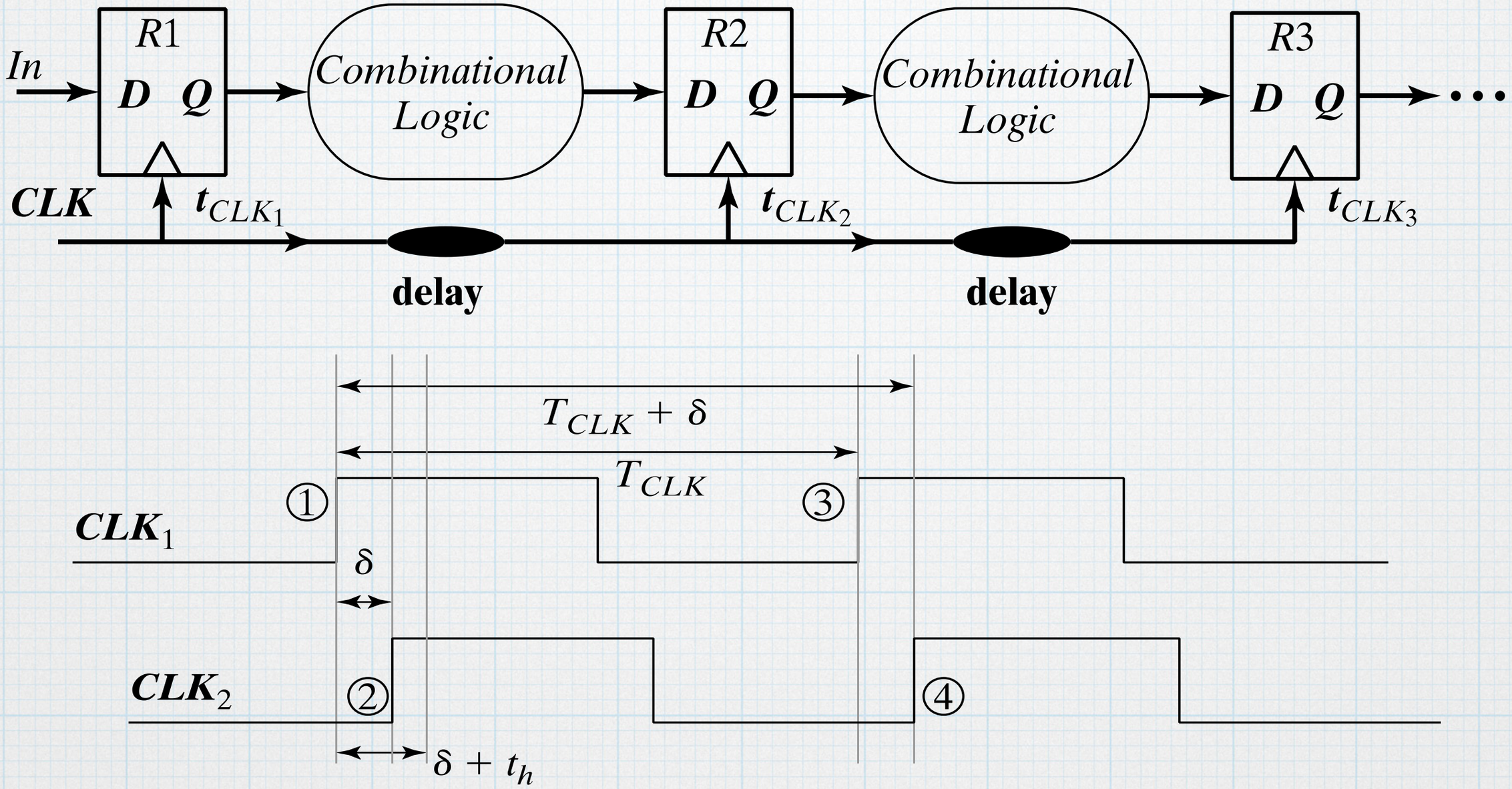


Clock Skew and Jitter

- Both skew and jitter affect the effective cycle time and the race margin

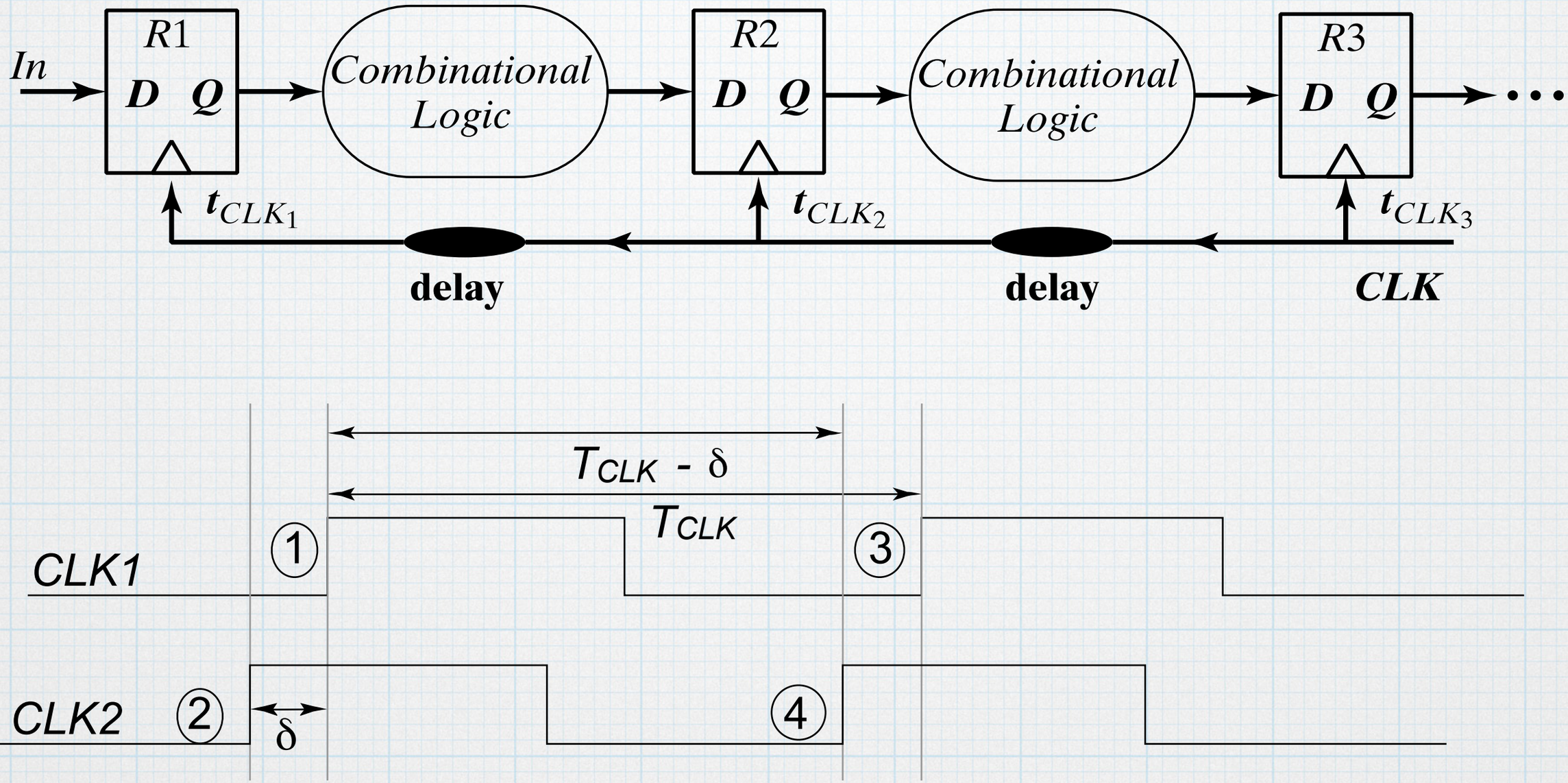


Positive Skew



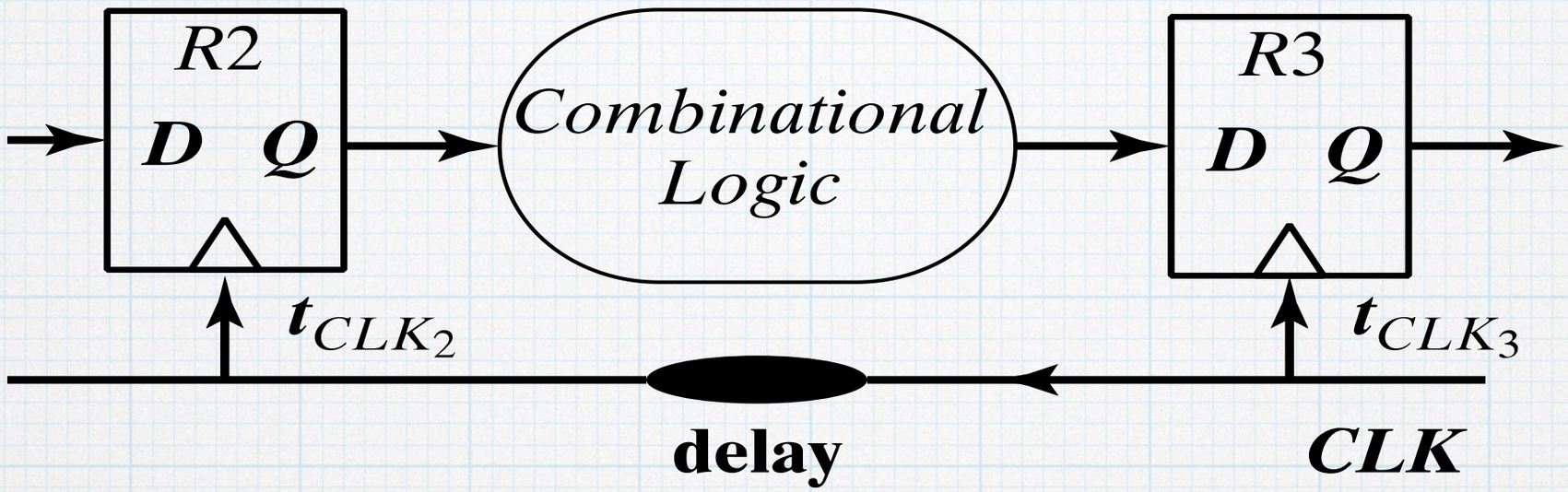
Launching edge arrives before the receiving edge

Negative Skew



Receiving edge arrives before the launching edge

Timing Constraints



$t_{clk\text{-}q,max}$ $t_{logic,max}$

$t_{clk\text{-}q,min}$

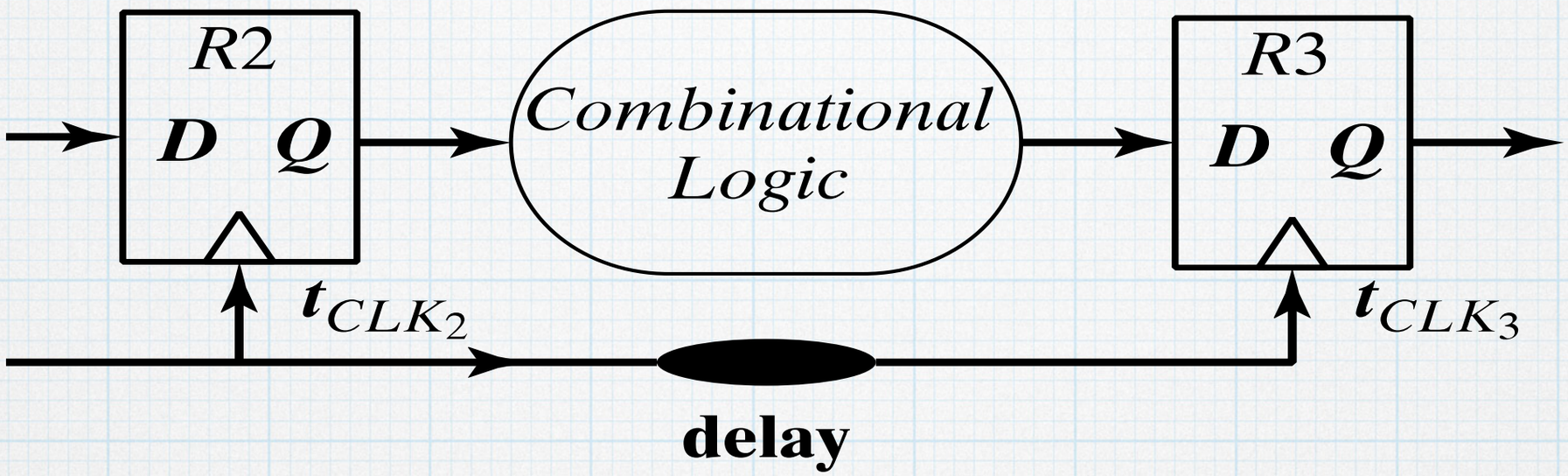
t_{setup}, t_{hold} $t_{logic,min}$

Minimum cycle time:

$$T_{clk} + \delta = t_{clk\text{-}q,max} + t_{setup} + t_{logic,max}$$

Skew may be negative or positive

Timing Constraints



$t_{clk\text{-}q,max}$ $t_{logic,max}$

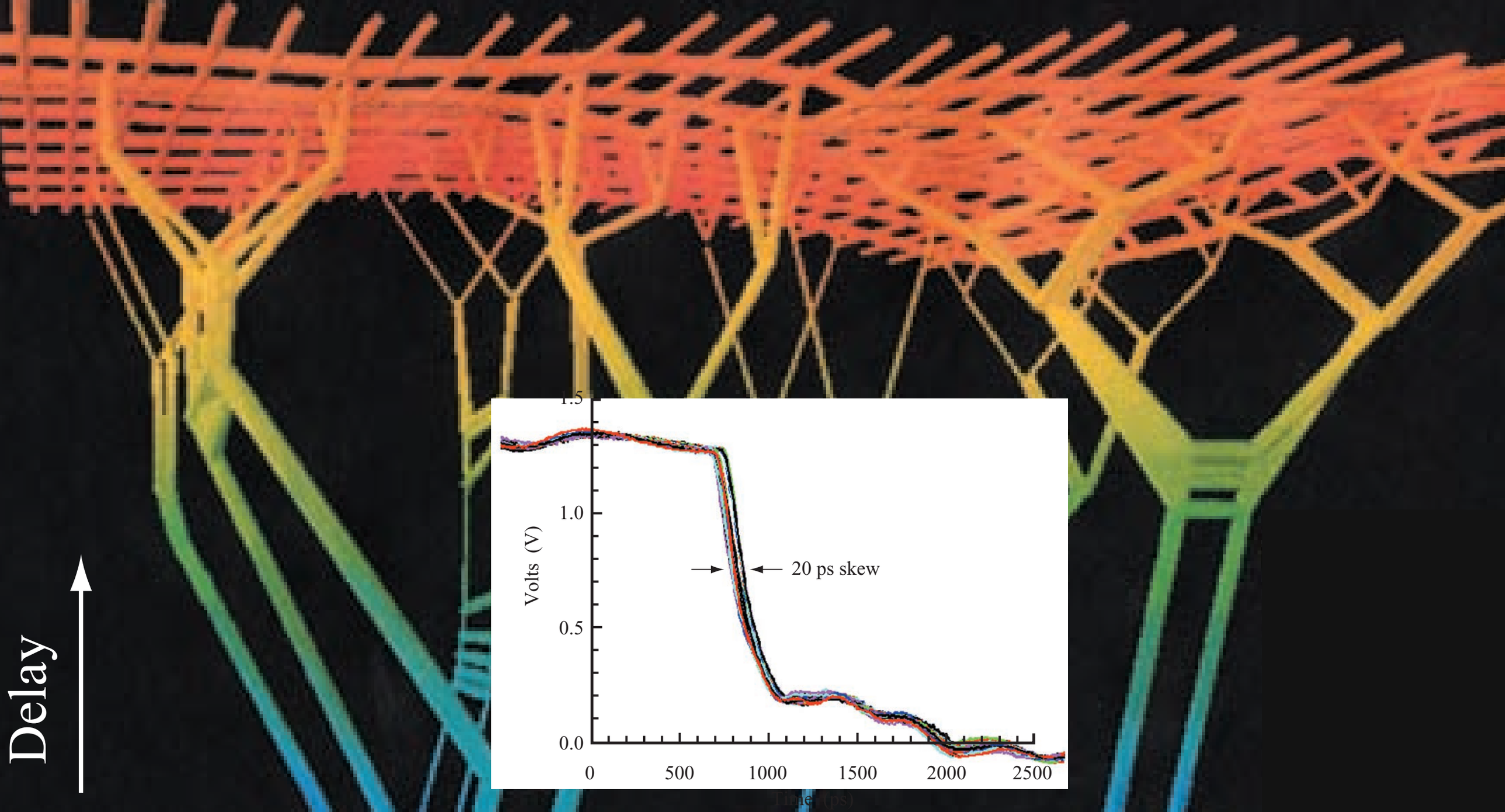
$t_{clk\text{-}q,min}$ $t_{logic,min}$

t_{setup}, t_{hold}

Hold time constraint:

$t_{(clk\text{-}q,min)} + t_{(logic,min)} > t_{hold} + \delta$

Skew may be negative or positive



Clock Tree Delays, IBM Power

Clock Constraints in Edge-Triggered Systems

If launching edge is late and receiving edge is early, the data will not be too late if:

$$t_{clk\text{-}q,max} + t_{logic,max} + t_{setup} < T_{CLK} - t_{JS,1} - t_{JS,2} + \delta$$

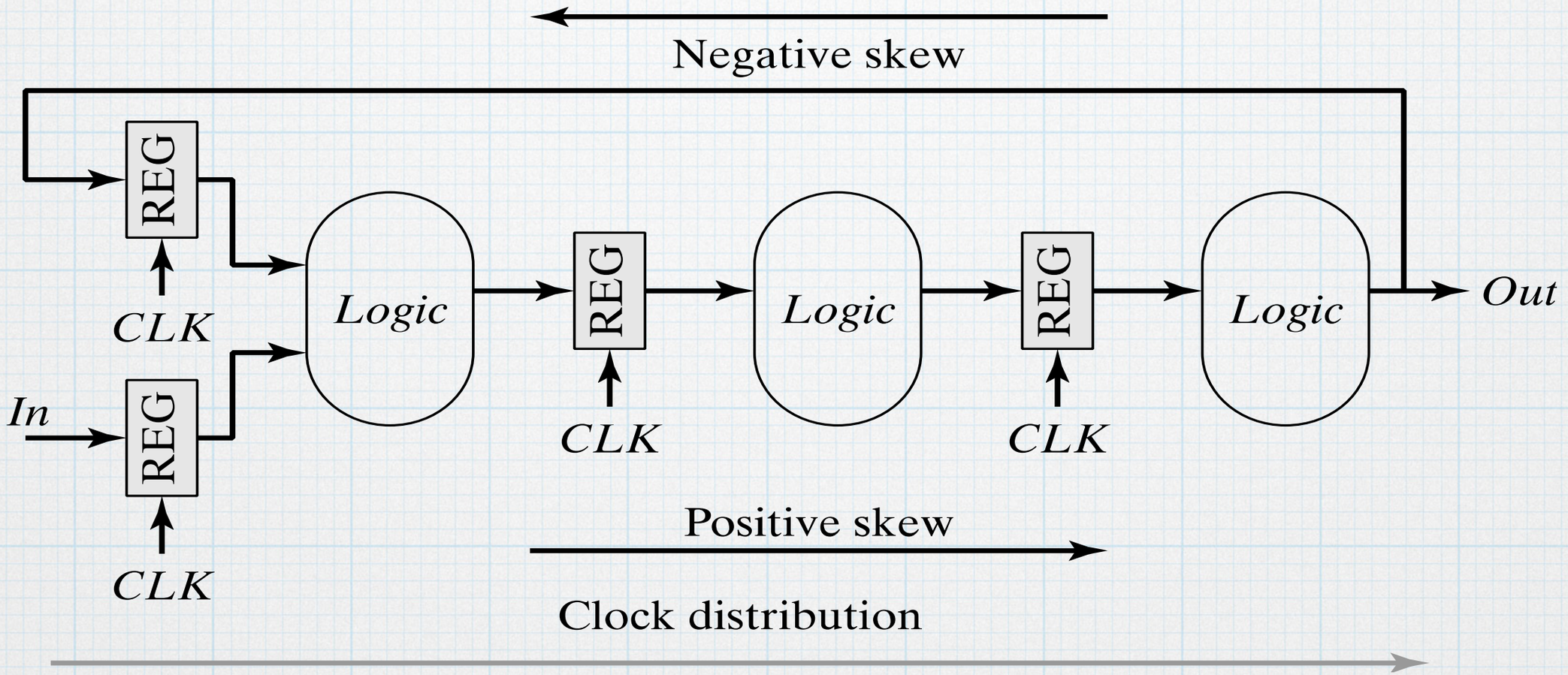
Minimum cycle time is determined by the maximum delays through the logic

$$t_{clk\text{-}q,max} + t_{logic,max} + t_{setup} - \delta + 2t_{JS} < T_{CLK}$$

Skew can be either positive or negative

Jitter t_{JS} usually expressed as peak-to-peak or n x RMS value

Datapath with Feedback

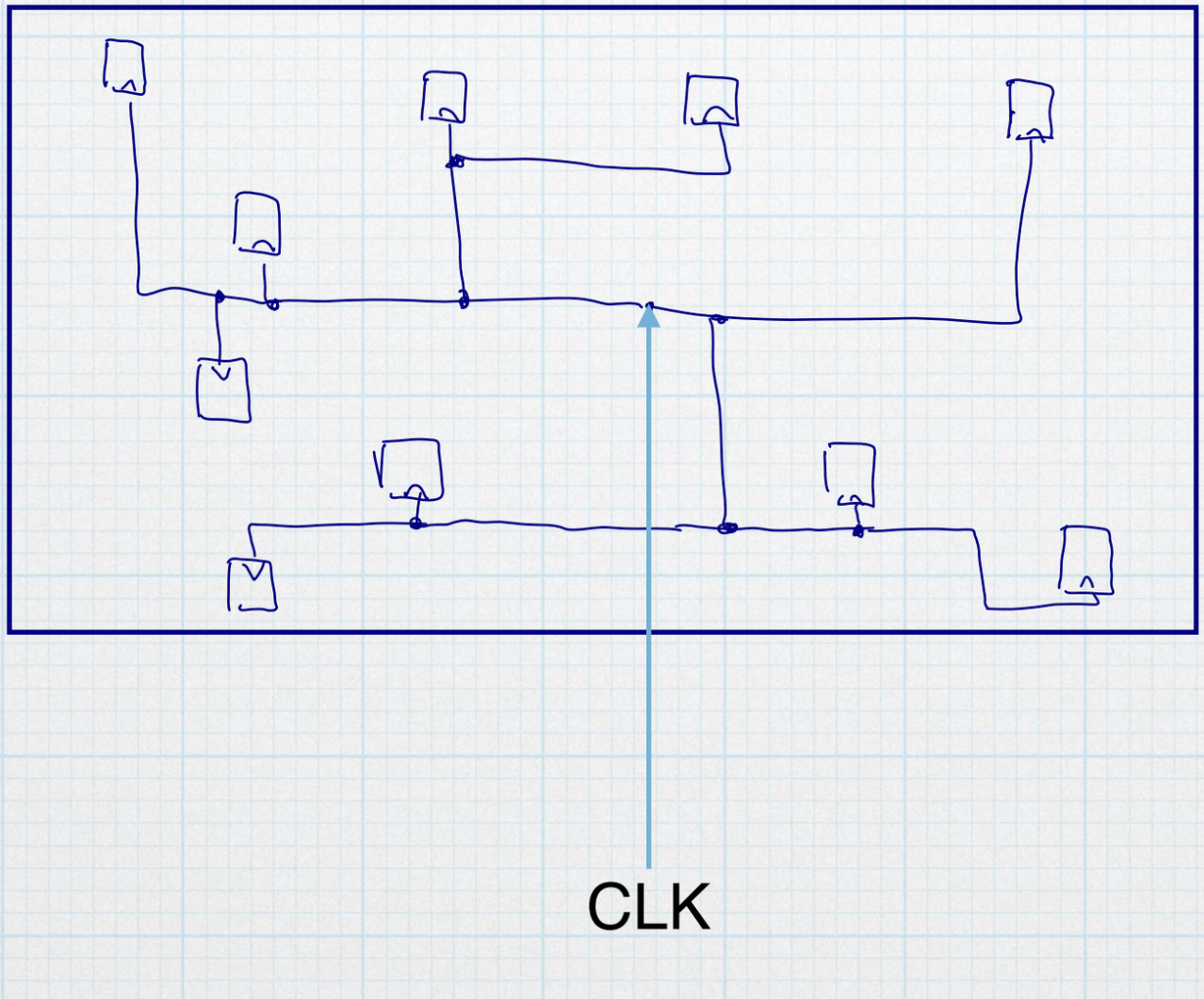


Clock Distribution

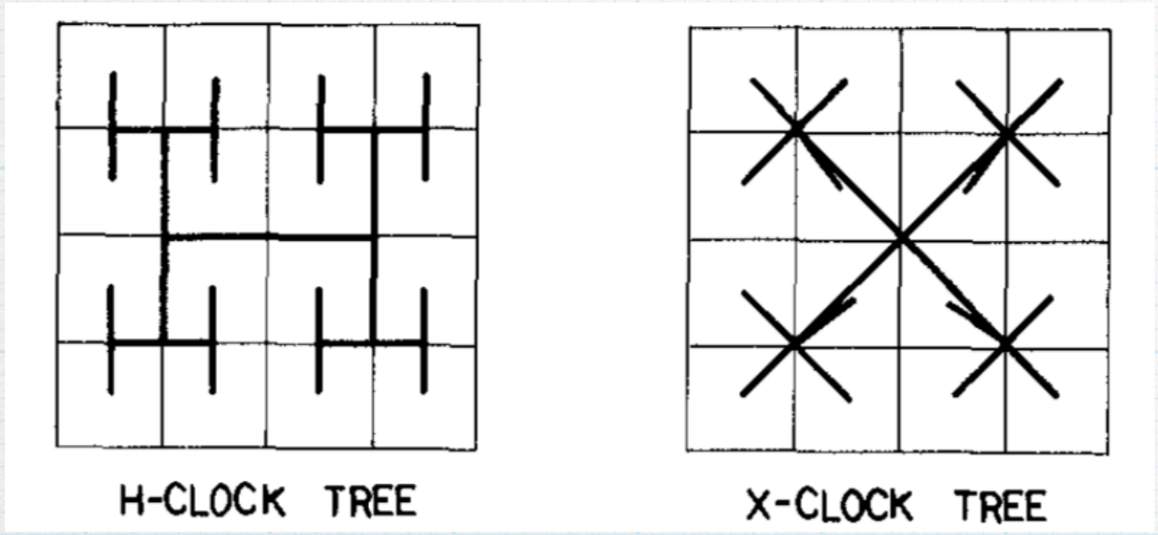
- Single clock generally used to synchronize all logic on the same chip (or region of chip)
  - Need to distribute clock over the entire region
  - While maintaining low skew/jitter
  - And without burning too much power

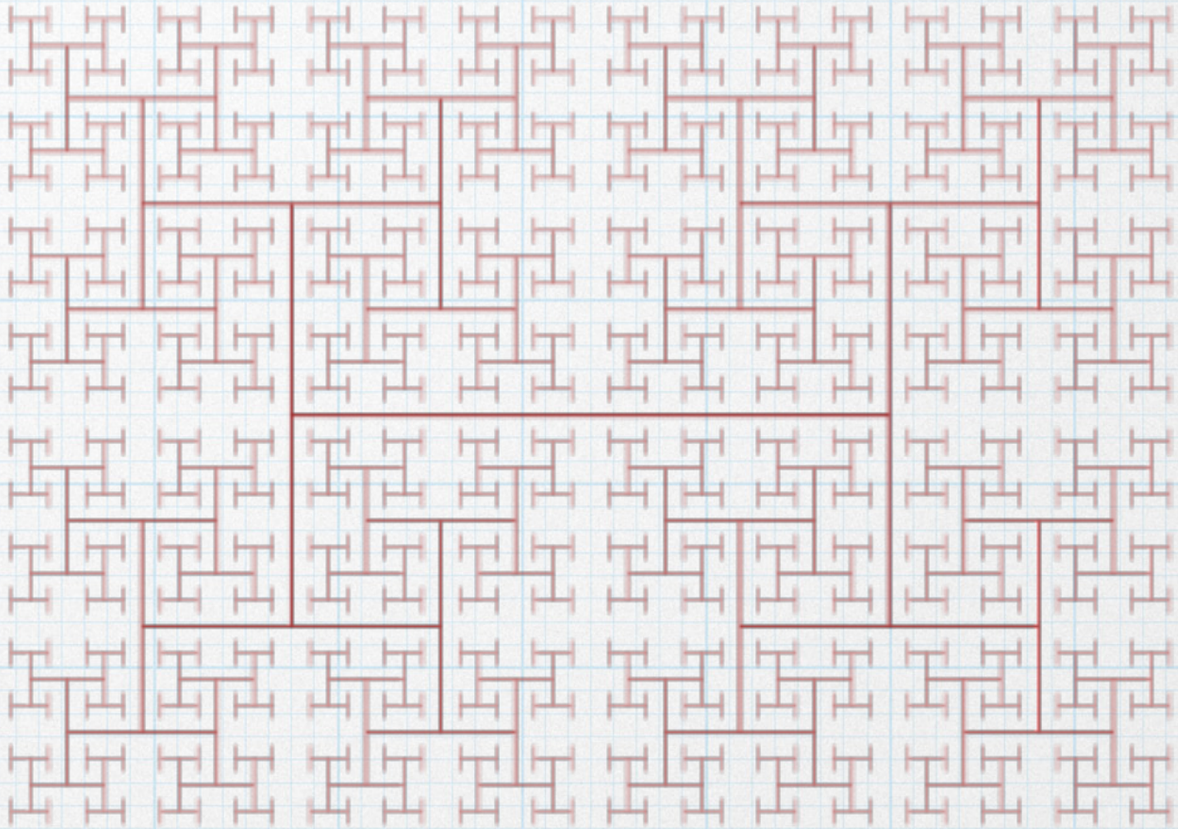
Clock Distribution

- What's wrong with just routing wires to every point that needs a clock?



Symmetric Clock Trees

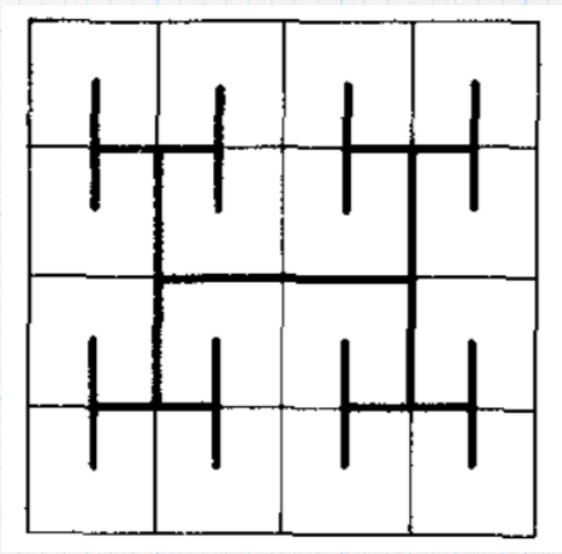




- Minimizes skew by matching RC delays from drive point to all terminal points
- Is that sufficient?

H-Tree Clock Distribution

- H-tree ensures that the relative RC delay to each terminal is *matched*
- *What about absolute RC delay?*



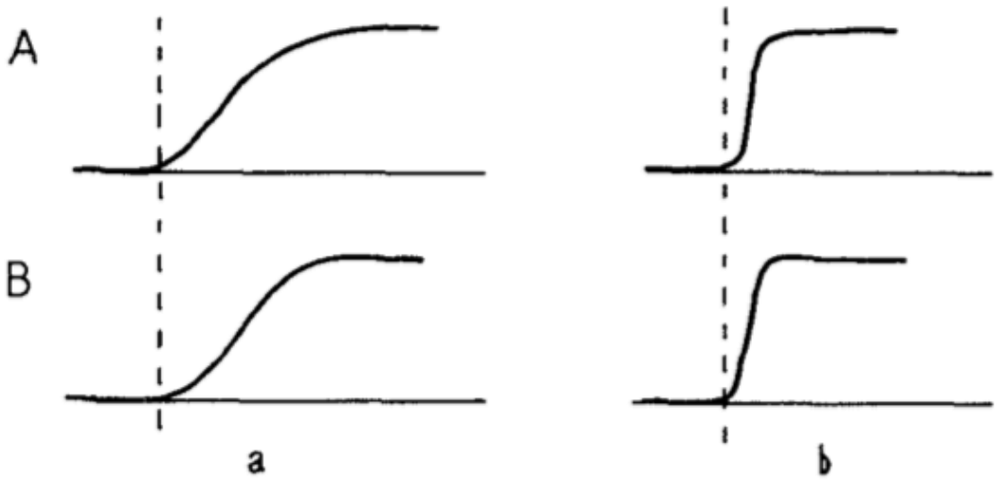
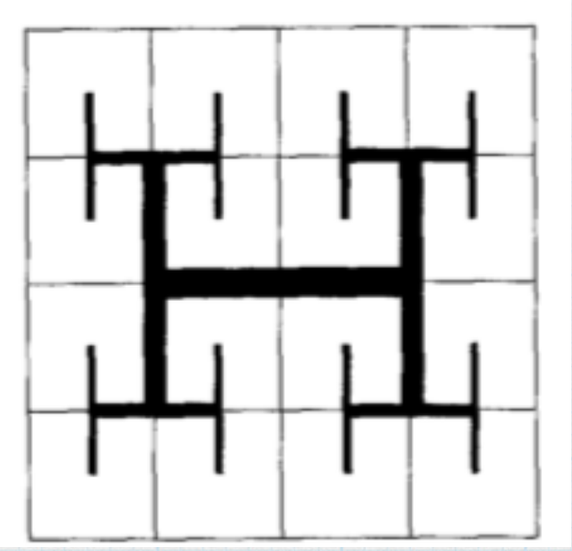


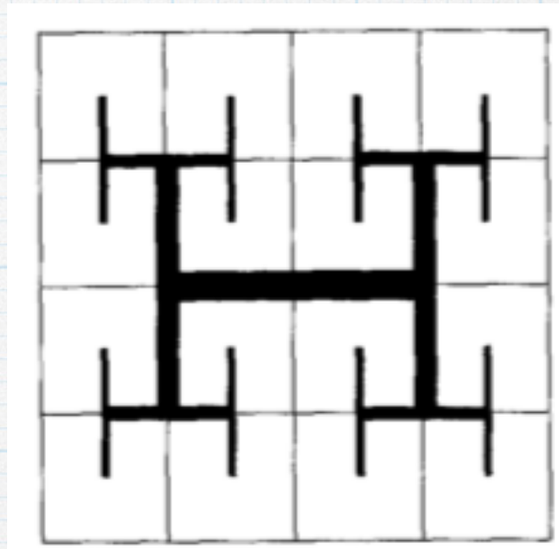
FIGURE 8.30 Comparison of slow- and fast-rising clock signals. Even if the subblocks receive identical signals, the RC constant of the interconnections must be sufficiently small to achieve high-frequency clock signals: (a) no clock skew and long rise time, (b) no clock skew and short rise time.



Wide wires reduce RC delay

H-Tree Clock Distribution

- Tapered H-tree can ensure high frequency (low RC value on paths)
- Furthermore, wide wires reduces the RC variation (less skew)

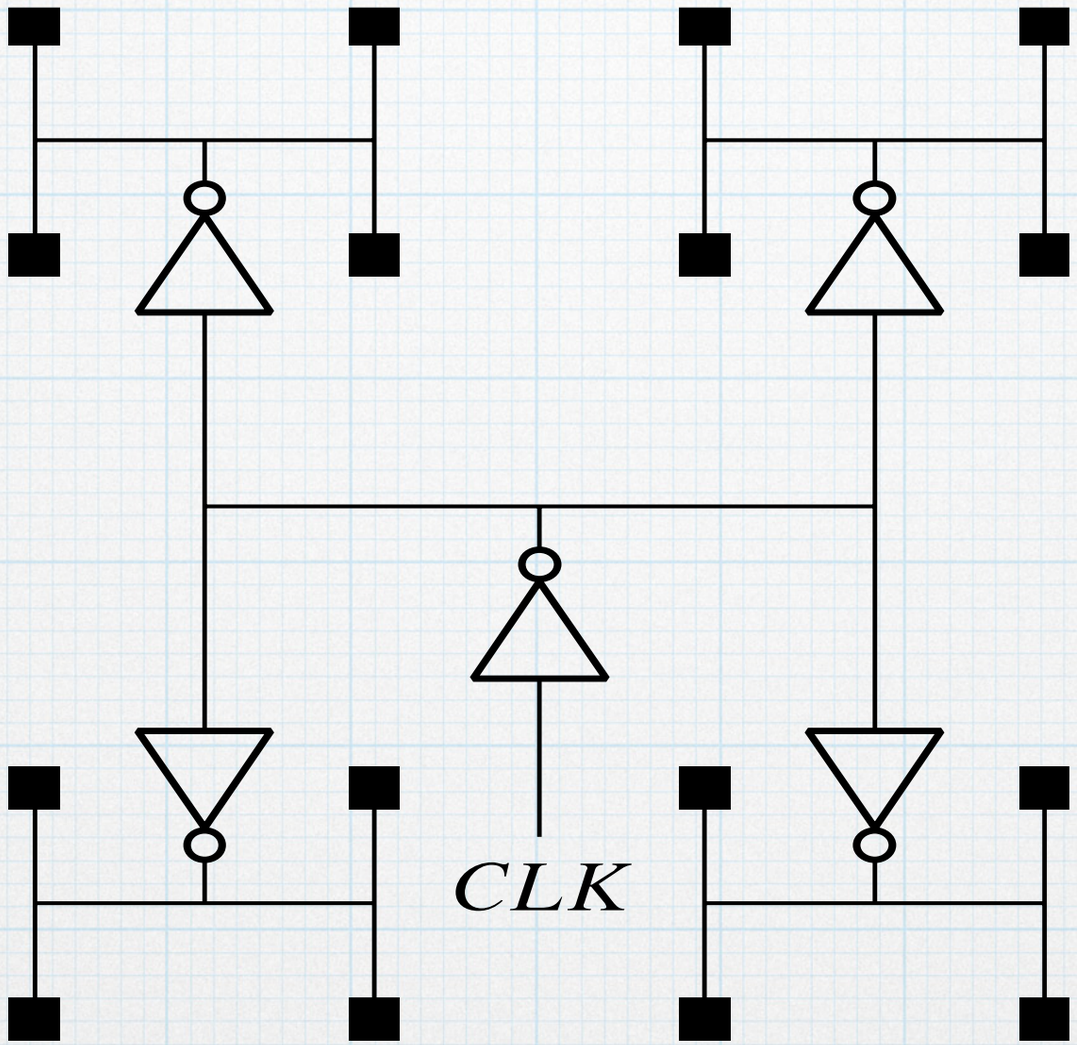


Wide wires increases C

- *What about power?*

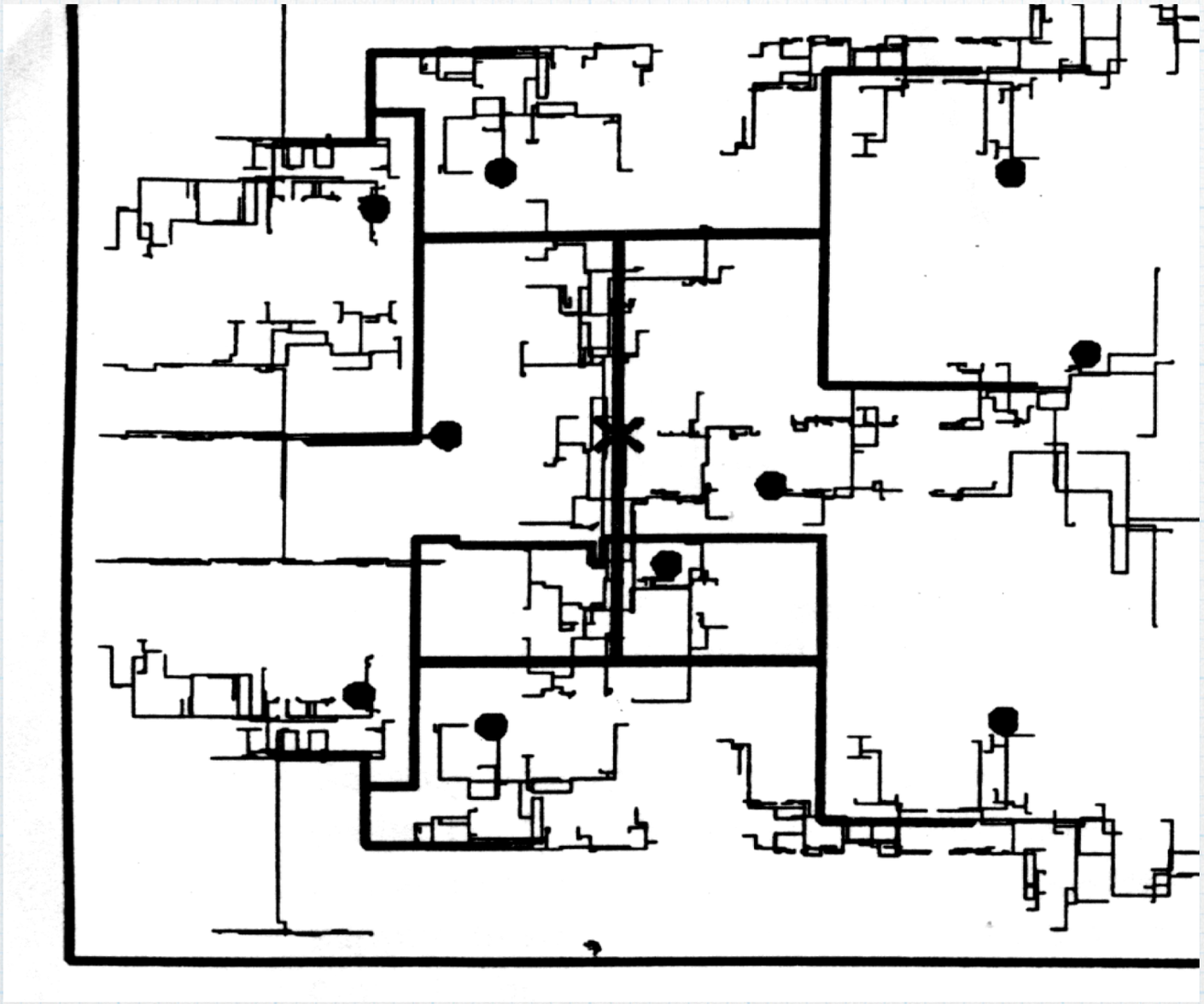
- *Adding buffers along the way reduces power and maintains high-frequency, but introduces skew!*

Buffered H-Tree



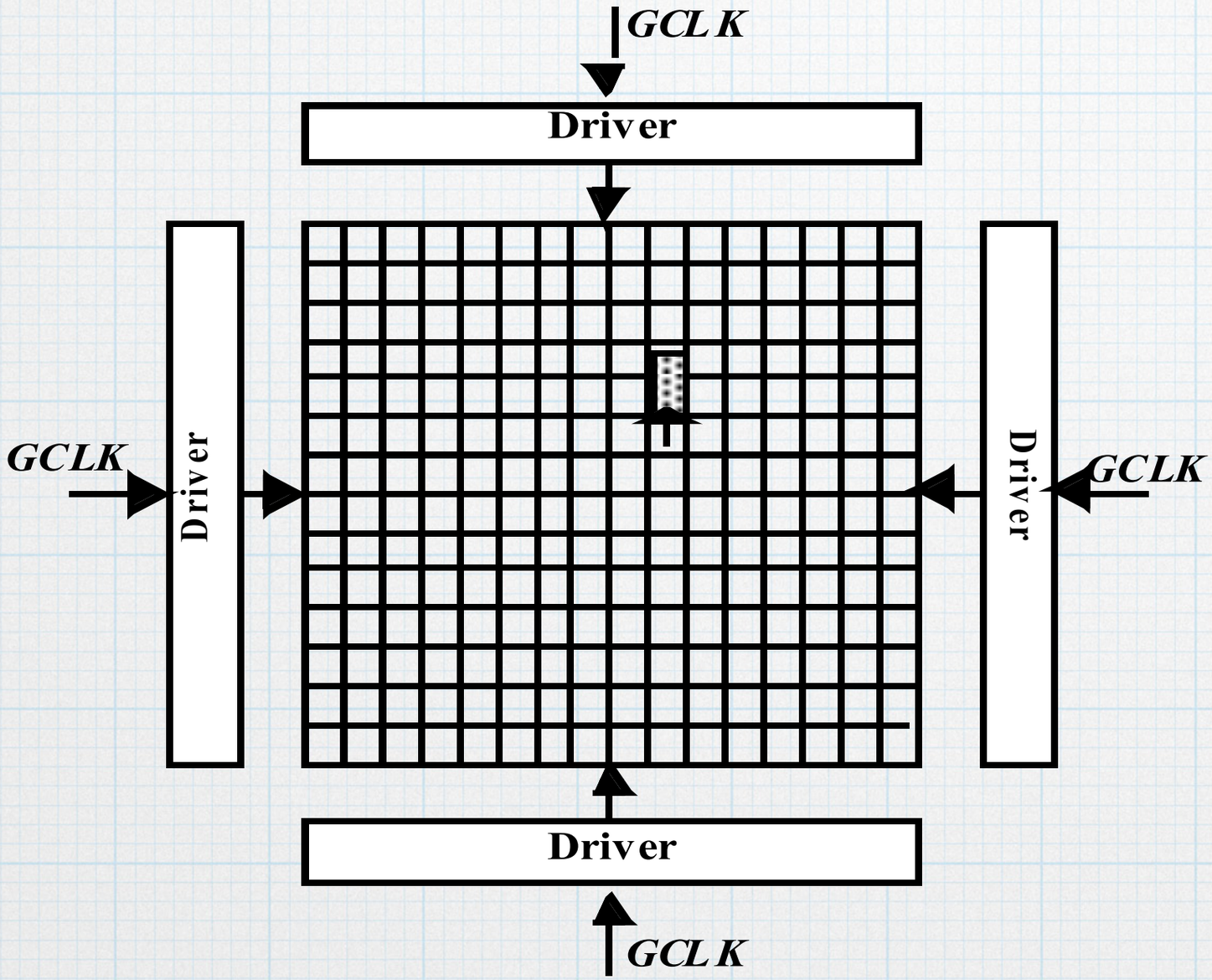
To minimize skew, all buffers must be matched (same layout, orientation)

More realistic ASIC H-tree



[Restle98]

Clock Grid Alternative



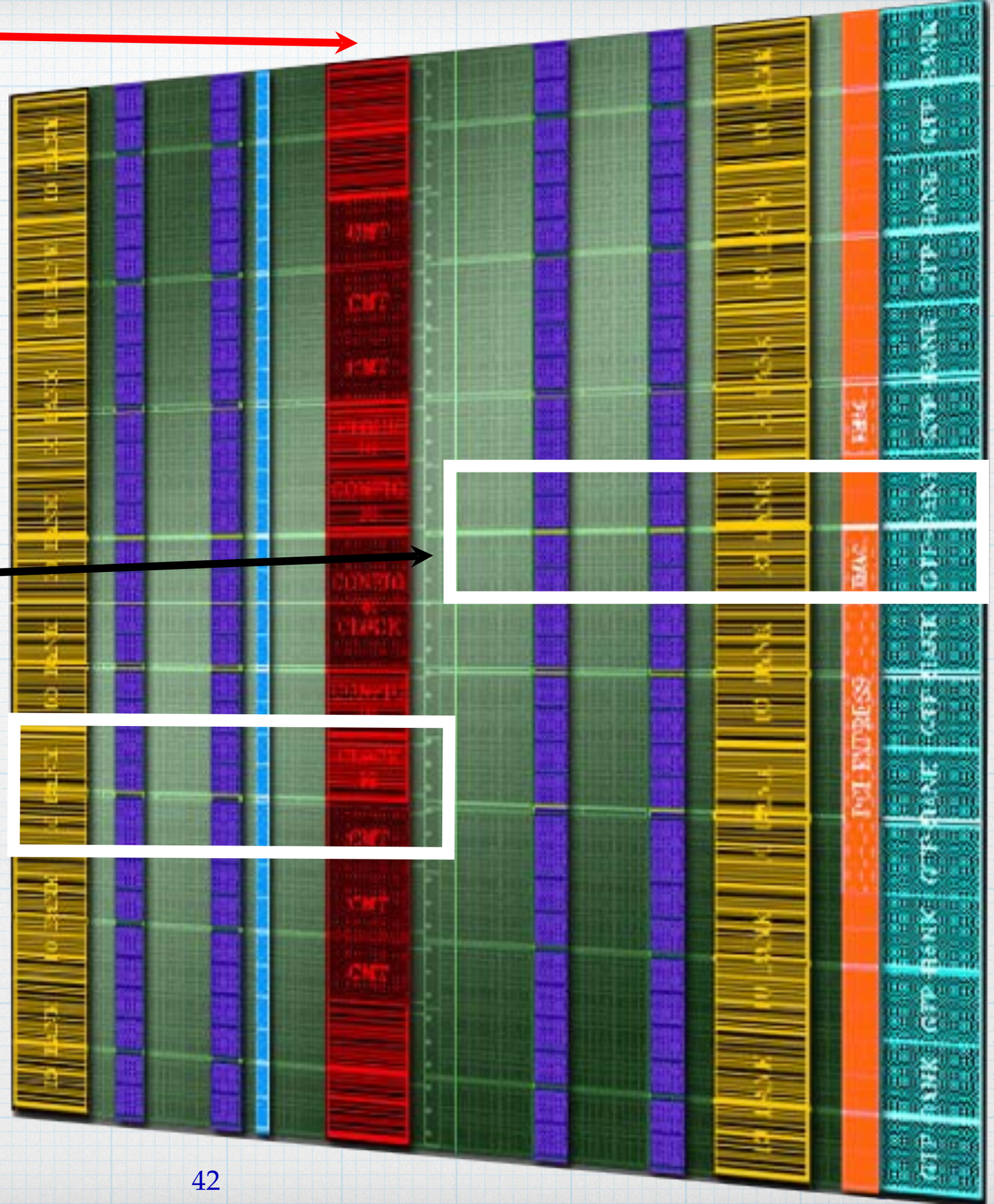
- No RC matching
- But huge power

Clock circuits live in center column.

32 global clock wires go down the red column.

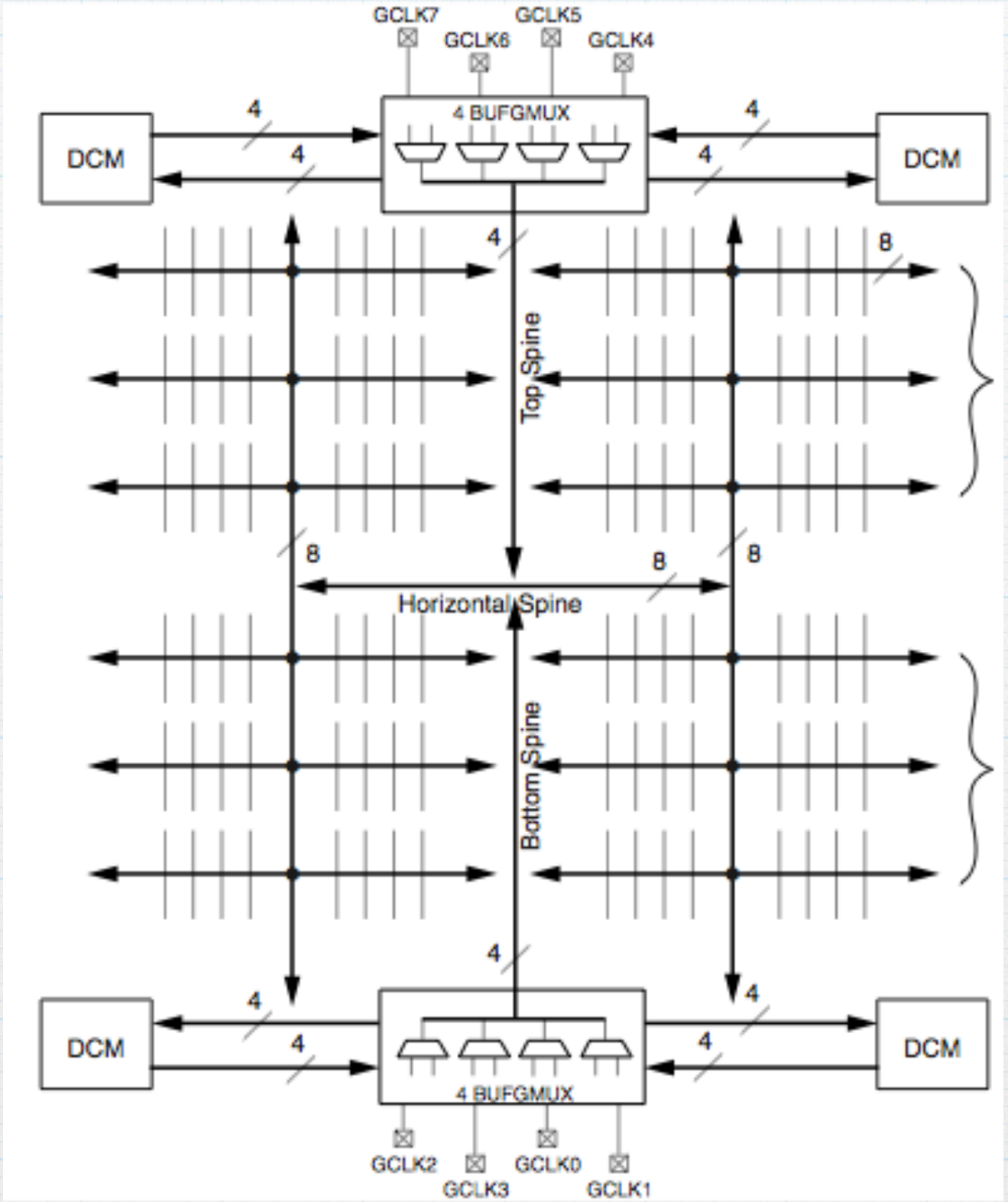
Any 10 may be sent to a clock region.

Also, 4 regional clocks (restricted functionality).



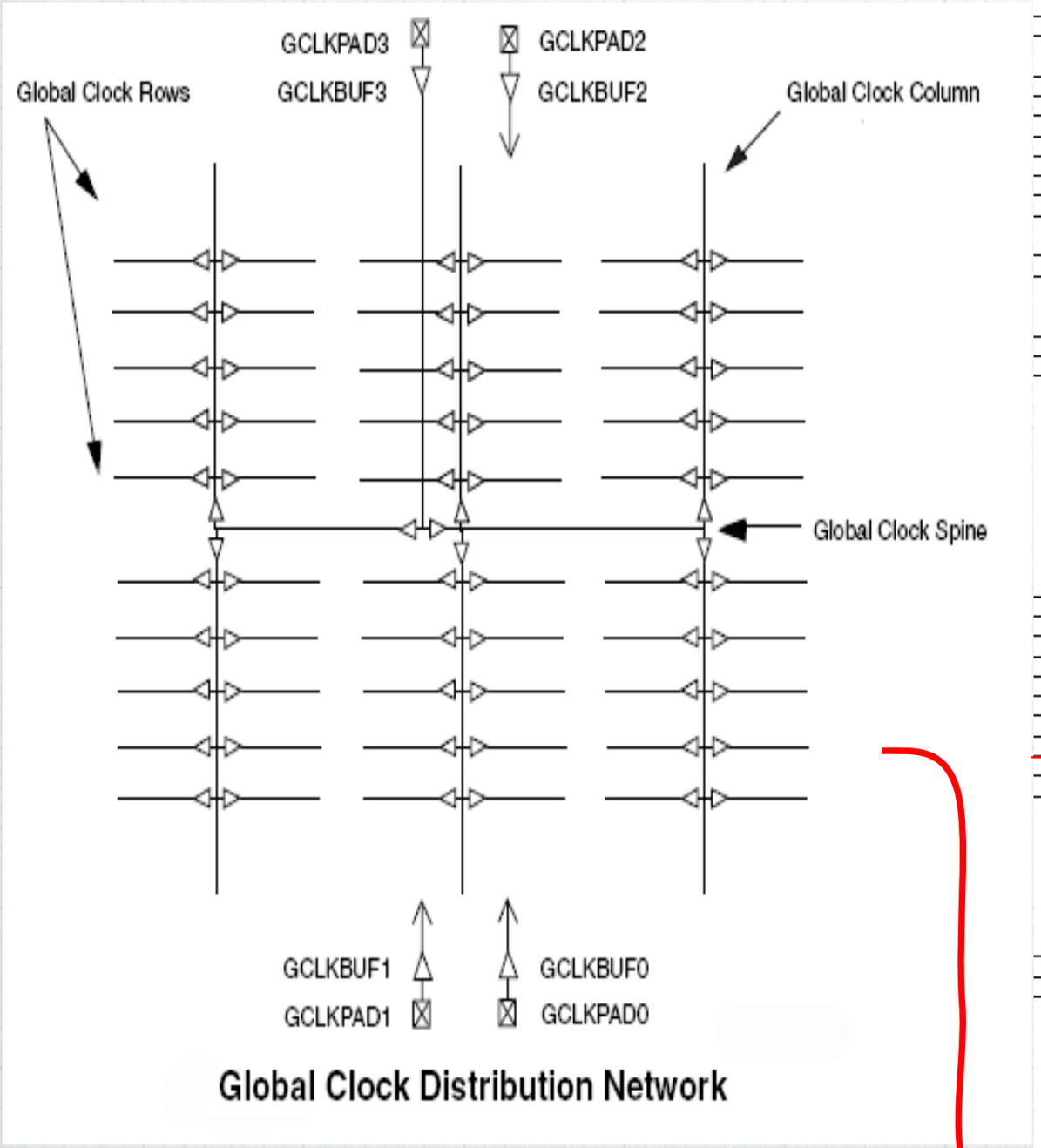
FPGA

Clocks have dedicated wires (low skew)

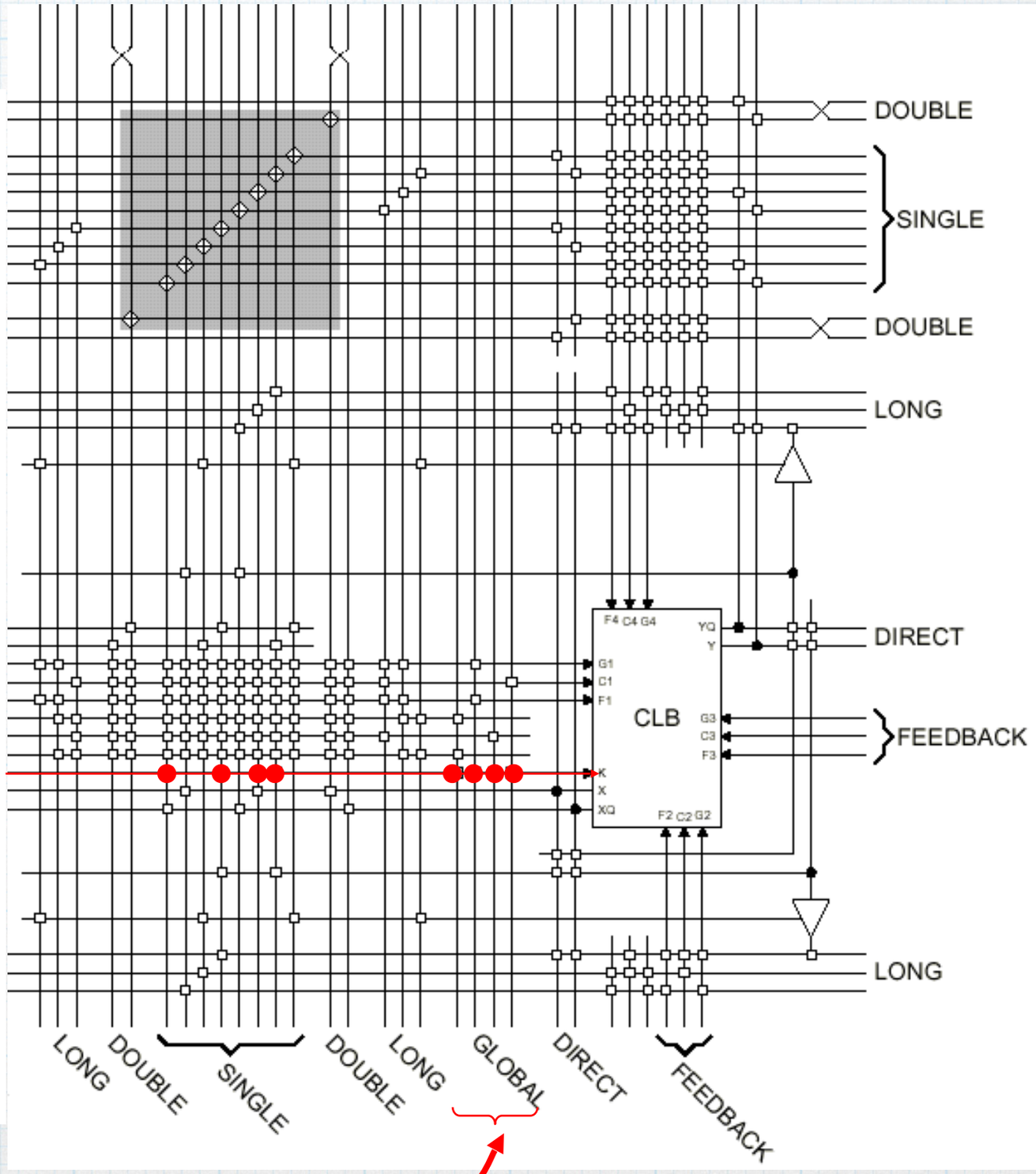


From: Xilinx Spartan 3 data sheet. Virtex is similar.

Low-skew Clocking in FPGAs



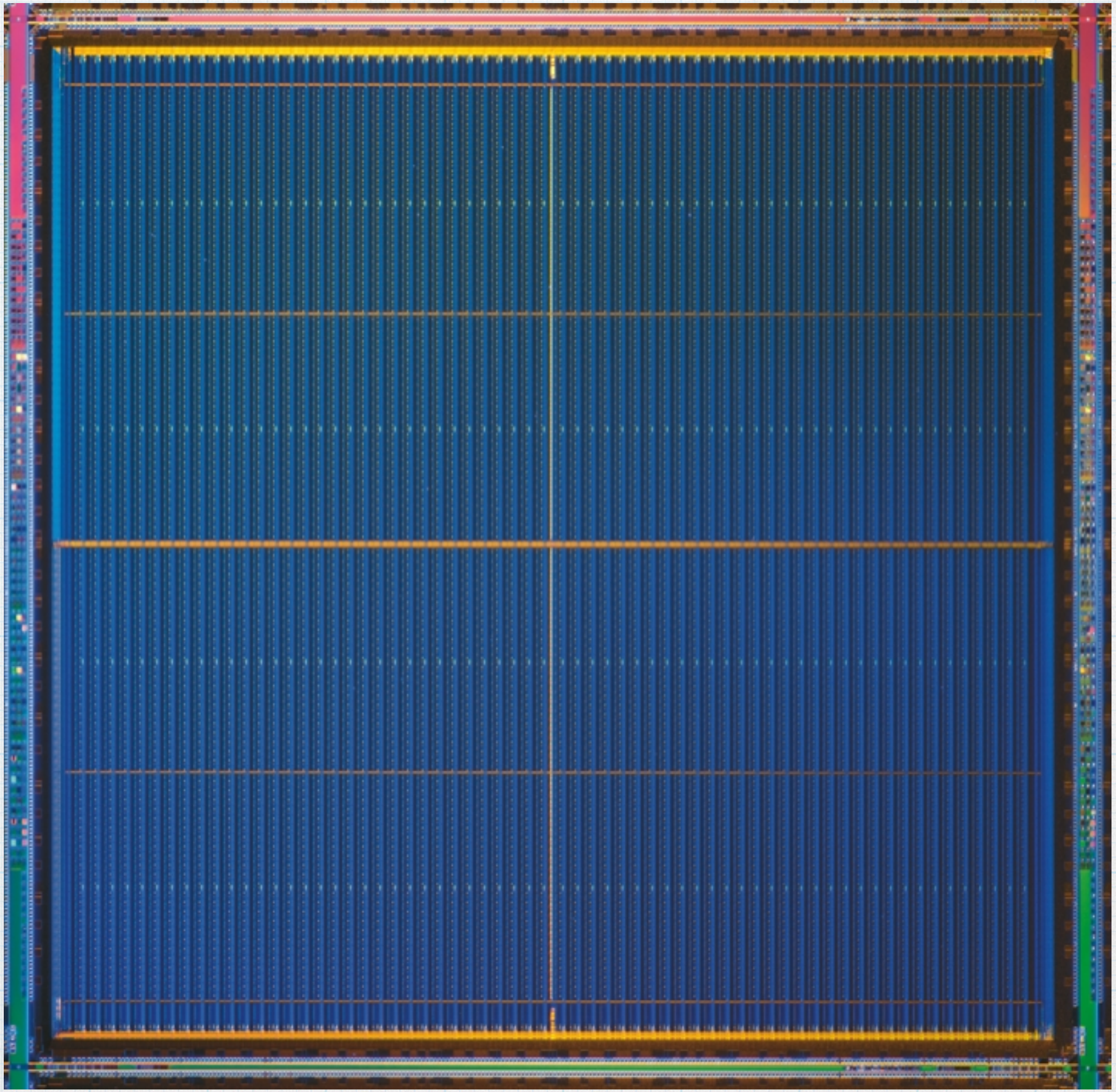
Global Clock Distribution Network



Figures from Xilinx App Notes

Die photo: Xilinx Virtex

Gold wires are the clock tree.



End of Lecture 10